# Oracle® Fusion Middleware

Administrator's Guide for Oracle Unified Directory

11*g* Release 2 (11.1.2)

**E22648-05**

June 2013

ORACLE

Oracle Fusion Middleware Administrator's Guide for Oracle Unified Directory, 11*g* Release 2 (11.1.2)

E22648-05



Primary Author: Devanshi Mohan, Showvik Chowdhuri

# Contents

## 7 Understanding the Oracle Unified Directory Indexing Model

## 8 Understanding the Oracle Unified Directory Access Control Model

## 9 Understanding the Oracle Unified Directory Schema Model

## 12 Understanding Oracle Unified Directory Mapping



## Part III Basic Administration

## 13 Starting and Stopping the Server

## 14 Configuring the Server Instance

## 15 Configuring the Proxy Components

## 16 Example Proxy Configurations



## 17 Managing Directory Data

## Part IV Advanced Administration: Security, Access Control, and Password Policies

## 20 Configuring Security Between Clients and Servers

## 21 Configuring Security Between the Proxy and the Data Source



## 22 Controlling Access To Data

## 23 Managing Administrative Users

## 24 Managing Password Policies

## 25 Integrating With Oracle's Enterprise User Security



# Part V Advanced Administration: Data Replication, Schema Management, and Moving Across Environments

## 26 Replicating Directory Data

## 27 Managing Directory Schema

## 28 Moving From a Test to a Production Environment

## Part VI Advanced Administration: Monitoring and Tuning Performance

## 29 Monitoring Oracle Unified Directory

## 30 Tuning Performance



## A Oracle Unified Directory Command Line Interface

## List of Examples

## List of Figures

## List of Tables

# Preface

The *Oracle Unified Directory 11g Release 2 (11.1.2) Administration Guide* is intended to provide typical configuration and administration tasks that are required on a deployed Oracle Unified Directory server.

## Audience

This document is intended for administrators of deployed Oracle Unified Directory servers.

## Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc`.

### Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

## Related Documents

The following documents provide related information for the daily administration of Oracle Unified Directory:

- *Oracle Fusion Middleware Installation Guide for Oracle Unified Directory*
- *Oracle Fusion Middleware Release Notes for Oracle Unified Directory*

## Conventions

The following text conventions are used in this document:

| Convention | Meaning |
|---|---|
| **boldface** | Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary. |

| Convention | Meaning |
| --- | --- |
| *italic* | Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values. |
| `monospace` | Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter. |

# What's New in This Guide?

This preface introduces the new and changed features of Oracle Unified Directory and Oracle Directory Services Manager (ODSM) since the previous release, and provides pointers to additional information. The information includes the following sections:

- What's New in Oracle Unified Directory 11g Release 2 PS1 (11.1.2.1.0)
- What's New in Oracle Unified Directory 11g Release 2 (11.1.2)

## What's New in Oracle Unified Directory

This section provides a concise summary of the new features in this release, and contains the following topics:

- What's New in Oracle Unified Directory 11g Release 2 PS1 (11.1.2.1.0)
- What's New in Oracle Directory Services Manager 11g Release 2 PS1 (11.1.2.1.0)

### What's New in Oracle Unified Directory 11*g* Release 2 PS1 (11.1.2.1.0)

This section provides a concise summary of the new features in this release of Oracle Unified Directory, and covers the following topics:

- Support for Macros in ACIs
- Support for `nsuniqueid` Virtual Attribute
- Support for Criticality in Workflows
- Support for Logging Administration Operations
- Introducing the Transformation Framework
- Enhanced External Change Log Properties
- Support for Integrating Oracle Unified Directory and Enterprise User Security with an External LDAP Repository
- Support for Relocating the Root DSE Entry
- Support for RDN Changing
- Support for Directory Plug-Ins

#### Support for Macros in ACIs

Oracle Unified Directory now supports macro expressions to represent a DN in the target section of the ACI, in the bind rule section, or in both.

For more information, see Section 8.6, "Using Macro ACIs for Advanced Access Control."

### Support for `nsuniqueid` Virtual Attribute

Oracle Unified Directory introduces `nsuniqueid` operational virtual attribute that is assigned to each entry in the directory server to resolve naming conflicts while migrating legacy applications using Oracle Directory Server Enterprise Edition as an LDAP database to Oracle Unified Directory.

For more information, see Section 17.10, "Configuring Virtual Attributes."

### Support for Criticality in Workflows

You can now configure criticality at the workflow level by setting the criticality flag.

For more information, see Section 15.1.4.6, "Configuring Criticality in Workflows."

### Support for Logging Administration Operations

Oracle Unified Directory enables you to log administration operations into a separate log file that provides logging information associated with administration traffic.

For more information, see Section 29.3.3, "Logging Operations to Access Log Publishers."

### Introducing the Transformation Framework

Oracle Unified Directory supports transformation through creation of an instance of workflow element.

For more information, see Section 11.6, "Understanding the Transformation Framework."

### Enhanced External Change Log Properties

Oracle Unified Directory provides additional properties, `ecl-include-del-only` and `ecl-blacklist` to configure attributes for external change log (ECL).

For more information, see Section 26.5.5, "Specifying the Attributes to be Included in the External Change Log" and Section 26.5.6, "Specifying the Attributes to be Excluded in the External Change Log."

### Support for Integrating Oracle Unified Directory and Enterprise User Security with an External LDAP Repository

Oracle Unified Directory supports the following external directories:

- Microsoft Active Directory
- Novell eDirectory
- Oracle Directory Server Enterprise Edition

For more information, see Section 25.4, "Integrating with Enterprise User Security and an External LDAP Directory."

### Support for Relocating the Root DSE Entry

Oracle Unified Directory allows you to relocate Root DSE, which is a special entry that provides information about the server's name, version, naming contexts, and supported features.

For more information, see Section 14.1.6.5, "Relocating the Root DSE Entry for a Network Group."

### Support for RDN Changing

Oracle Unified Directory enables you to rename or replace RDN values from the source directory to Oracle Unified Directory using the `RDNChanging` configuration.

For more information, see Section 11.5, "RDN Changing."

### Support for Directory Plug-Ins

Oracle Unified Directory supports Directory plug-in API as a means to extend the existing Directory Server functionality.

For more information, see *Oracle® Fusion Middleware Developer's Guide for Oracle Unified Directory*.

## What's New in Oracle Directory Services Manager 11*g* Release 2 PS1 (11.1.2.1.0)

This section provides a summary of the new features in this release of Oracle Directory Services Manager (ODSM), and covers the following topics:

- Support for IBM WebSphere Application Server
- Enhanced Log Publisher Configuration
- Integration with Macro ACIs
- Support for Criticality Flag to Configure Workflows
- Support for Virtual Attributes
- Support for Transformations
- Support for New Workflow Elements
- Support for Configuring the Root DSE Entry
- Support for Configuring RDN Changing Workflow Element

### Support for IBM WebSphere Application Server

You can install and configure IBM WebSphere Application Server - Network Deployment (ND) to work with Oracle Unified Directory. This is possible only if you are already managing Oracle Unified Directory using the graphical Oracle Directory Service Manager interface. For more information, see "Configuring IBM WebSphere for Oracle Directory Services Manager" in the *Oracle Fusion Middleware Installation Guide for Oracle Unified Directory*.

### Enhanced Log Publisher Configuration

ODSM supports a new parameter to log administration operations in the access logs.

For more information, see Section 29.3.3.2, "Configuring Logged Operations in Access Log Publishers Using ODSM."

### Integration with Macro ACIs

ODSM supports macro expressions to represent a DN in the target section of the ACI, in the bind rule section, or in both.

For more information, see Section 22.4, "Managing Macro ACIs With Oracle Directory Services Manager."

### Support for Criticality Flag to Configure Workflows

ODSM supports a new parameter, the criticality flag to configure workflows.

For more information, see Section 15.2.3, "Configuring Criticality in Workflows With ODSM."

#### Support for Virtual Attributes

ODSM allows you to configure virtual attributes.

For more information, see Section 17.14, "Managing Virtual Attributes With Oracle Directory Services Manager."

#### Support for Transformations

ODSM allows you to define transformations through the creation of transformation workflow element.

For more information, see Section 15.2.4, "Configuring Transformations With ODSM."

#### Support for New Workflow Elements

ODSM now allows you to create the following workflow elements:

- Kerberos Authentication Provider Workflow Element
- RDN Changing Workflow Element
- Transformations Workflow Element

For more information, see Section 14.2.4, "Configuring Workflow Elements With ODSM."

#### Support for Configuring the Root DSE Entry

ODSM supports the ability to configure Enterprise User Security.

For more information, see Section 14.2.7, "Configuring Network Groups With ODSM."

#### Support for Configuring RDN Changing Workflow Element

ODSM allows you to configure the RDN Changing workflow element.

For more information, see Section 14.2.4, "Configuring Workflow Elements With ODSM."

## What's New in Oracle Unified Directory 11*g* Release 2 (11.1.2)

This section provides a concise summary of the new features in this release, and contains the following topics:



### What's New in Oracle Unified Directory 11*g* Release 2 (11.1.2)

This section provides a concise summary of the new features in this release of Oracle Unified Directory, and covers the following topics:

- Support for Oracle's Enterprise User Security (EUS)
- Enhanced Support for Social Networking Applications
- Improved CLI for Configuring External ChangeLog
- Support for Test to Production Environments
- Suppressing Password Display on CLI

#### Support for Deterministic Identity Mapper Evaluation Order

It is imperative to define the order in which identity mappers are evaluated in the network group to avoid conflicts. You can now define priorities for the conflicting identity mappers.

For more information, see Section 12.6, "Ordering Identity Mappers."

#### Support for LDAP Referrals

When a server is unable to handle a client's request, it sends a list of referrals to the client, which point the client to other servers in the topology. The client then performs the operation again on one of the remote servers in the referral list.

For more information, see Section 17.13, "Configuring Referrals."

#### New Bind Mode Parameters

You can now configure proxy LDAP workflow elements with two additional parameters, such as the `never-bind` parameter, `use-proxy-auth` parameter, and the include and exclude lists to tweak the behavior of the server.

For more information, see Section 15.1.2, "Configuring the Bind Mode."

#### Support for Microsoft Active Directory Paging

Oracle Unified Directory now supports Active Directory range retrieval by providing support for Microsoft Active Directory paging.

For more information, see Section 15.1.8, "Configuring Microsoft Active Directory Paging."

#### Support for the Criticality Flag

Oracle Unified Directory now implements criticality configuration, which permits the Oracle Unified Directory proxy server to return partial data to a client if a search operation fails, due to a host error.

For more information, see Section 15.1.4.7, "Configuring Criticality in Workflow Elements."

#### Support for Oracle's Enterprise User Security (EUS)

Integrating Oracle Unified Directory with EUS enables you to store user identities in Oracle Unified Directory for Oracle Database authentication.

In this release, support for EUS is limited to password authentication (certificate authentication and integration with Kerberos are not supported at this stage).

For more information, see Chapter 25, "Integrating With Oracle's Enterprise User Security."

### Enhanced Support for Social Networking Applications

Social networking applications are now supported with two new controls, the Join control and the Proximity control.

For more information, see Section 17.5.3.2, "Searching Using the Join Search Control" and Section 17.5.3.3, "Searching Using the Proximity Search Control."

### Improved CLI for Configuring External ChangeLog

The External Change Log (ECL) functionality allows you to publish all changes that have occurred in a directory server database and is particularly useful for synchronizing the LDAP directory with other subsystems.

You now have a user-friendly CLI to configure external changelog using the `dsreplication` command.

For more information, see Section 26.5, "Using the External Change Log."

### Support for Test to Production Environments

You can now install, configure, customize, and validate Oracle Unified Directory in a test environment. Once the system performs as expected, you can create the production environment by moving a copy of the server and its configuration from the test environment, instead of redoing all the changes that were incorporated into the test environment.

For more information, see Chapter 28, "Moving From a Test to a Production Environment."

### Suppressing Password Display on CLI

Some commands had an option where the password was provided in a clear text format on the CLI. This resulted in security exposure, because one could retrieve the password using the `ps` command on a UNIX machine.

The clear text format is deprecated now and the commands are modified to use the file-based option to store the password by introducing the following option:

```
-j, --bindPasswordFile
```

For more information, see Appendix A, "Oracle Unified Directory Command Line Interface."

### Ability to Encrypt the ADS Trust Store Pin

Oracle Unified Directory allows you to configure ADS trust store pin to determine whether to trust a certificate that is presented to it.

For more information, see Section 20.3, "Configuring Trust Manager Providers."

## What's New in Oracle Directory Services Manager 11*g* Release 2 (11.1.2)

This section provides a concise summary of the new features in this release of Oracle Directory Services Manager (ODSM), and covers the following topics:

- Suffix Configuration for EUS
- New User Interface to Configure Root Users
- Key Manager and Trust Manager Configuration
- Auto-Suggest Feature
- Support for Dynamic Groups

- Support for Virtual Static Groups
- Simplified Tree Structure of the Configuration Tab

#### Suffix Configuration for EUS

ODSM enables you to create and configure suffixes to work with Oracle Enterprise User Security (EUS).

For more information, see Section 14.2.3, "Configuring Suffixes With ODSM."

#### New User Interface to Configure Root Users

ODSM now provides a new user interface (UI) to configure root users.

For more information, see Section 19.2.2, "Configuring Root Users by Using ODSM."

#### Key Manager and Trust Manager Configuration

You can now configure key manager providers and trust manager providers by using ODSM.

For more information, see Section 20.2.6, "Configuring Key Managers With ODSM" and Section 20.3.5, "Configuring Trust Managers With ODSM".

#### Auto-Suggest Feature

ODSM now implements an auto-suggest feature in different tabs that helps streamline configuration and operations.

For more information, see Section 17.15, "Managing Data With Oracle Directory Services Manager."

#### Support for Dynamic Groups

OSDM now enables you to create dynamic groups whose membership is determined by search criteria using an LDAP URL.

For more information, see Section 19.3.2, "Defining Dynamic Groups."

#### Support for Virtual Static Groups

ODSM enables you to create virtual static groups, where each entry behaves like a static group entry by using virtual attributes.

For more information, see Section 19.3.4, "Defining Nested Groups."

#### Simplified Tree Structure of the Configuration Tab

The default view of the configuration tree in the Configuration tab has been simplified to provide a user-friendly view of the naming context (or suffix) configuration. In addition, presence of a contextual menu to launch all the relevant operations for a selected node simplifies user interaction.

For more information, see Section 14.2, "Managing the Server Configuration With Oracle Directory Services Manager."

# Part I

# Introduction to Oracle Unified Directory

This part provides an overview of Oracle Unified Directory and the modes in which it can be installed. The part also provides sample deployment scenarios for each server mode.

This part includes the following topics:

# 1 Overview of Oracle Unified Directory

This chapter provides an overview of Oracle Unified Directory and explains some of the unique features of Oracle Unified Directory.

This chapter contains the following topics:

- Section 1.1, "What is Oracle Unified Directory?"
- Section 1.2, "Overview of Directory Server"
- Section 1.3, "Overview of Proxy Server"
- Section 1.4, "Overview of the Replication Gateway"

## 1.1 What is Oracle Unified Directory?

Oracle Unified Directory is a comprehensive next generation directory service. It is designed to address large deployments and to provide high performance, and is highly extensive. Oracle Unified Directory is easy to deploy, manage, and monitor.

This section contains the following topics:

- Section 1.1.1, "Components of Oracle Unified Directory"
- Section 1.1.2, "Oracle Unified Directory Installation Types"
- Section 1.1.3, "Synchronizing Oracle Unified Directory with Other Directories"

### 1.1.1 Components of Oracle Unified Directory

Oracle Unified Directory includes:

- LDAP directory server, used for storing data

  For more information about directory server, see Section 1.2, "Overview of Directory Server."

- Proxy server, where the server acts as an interface between the client and the directory server that contains the data

  For more information about proxy server, see Section 1.3, "Overview of Proxy Server."

- Replication gateway between Oracle Unified Directory and Oracle Directory Server Enterprise Edition

  For more information about replication gateway, see Section 1.4, "Overview of the Replication Gateway."

For more information about which Oracle Unified Directory server mode you should use, see Section 1.1.2, "Oracle Unified Directory Installation Types."

## 1.1.2 Oracle Unified Directory Installation Types

The mode in which the Oracle Unified Directory server runs depends on how you install the software based on your requirement.

You can choose one of the following installation types when installing Oracle Unified Directory:

- Section 1.1.2.1, "Setting Up the Directory Server"
- Section 1.1.2.2, "Setting Up the Proxy Server"
- Section 1.1.2.3, "Setting Up the Replication Gateway Server"

### 1.1.2.1 Setting Up the Directory Server

If you want to create an LDAP directory server that contains directory data, then install Oracle Unified Directory as a directory server. For more information, see Setting Up the Directory Server chapter in *Oracle Fusion Middleware Installation Guide for Oracle Unified Directory*.

### 1.1.2.2 Setting Up the Proxy Server

If you want the server to act as an interface between the client and the directory server containing the data, then install Oracle Unified Directory as a proxy server. The proxy server does not contain any data. It handles client requests through load balancing or data distribution. For more information about setting up the proxy server, see Setting Up the Proxy Server chapter in *Oracle Fusion Middleware Installation Guide for Oracle Unified Directory*.

### 1.1.2.3 Setting Up the Replication Gateway Server

If you want the Oracle Unified Directory server to replicate information between Oracle Unified Directory and Oracle Directory Server Enterprise Edition, then install Oracle Unified Directory as a replication gateway. For more information, see Setting Up the Replication Gateway chapter in *Oracle Fusion Middleware Installation Guide for Oracle Unified Directory*.

## 1.1.3 Synchronizing Oracle Unified Directory with Other Directories

You can synchronize Oracle Unified Directory with other directories using Oracle Directory Integration Platform.

**Note:** You can obtain Oracle Directory Integration Platform by installing Oracle Identity Management release 11.1.1.6.0 or above.

Oracle Directory Integration Platform consists of a set of services and interfaces that facilitates synchronization and provisioning solutions between the directory and other repositories.

If you want to use Directory Integration Platform to enable synchronization for Oracle Unified Directory, you need to enable the Oracle Unified Directory changelog. For more information about how to enable the changelog in Oracle Unified Directory, see Section 26.5, "Using the External Change Log."

Directory Integration Platform synchronization can be described as follows:

- Section 1.1.3.1, "Synchronization between Oracle Unified Directory and Oracle Internet Directory"
- Section 1.1.3.2, "Synchronization between Oracle Unified Directory and Third-Party Directories"

#### 1.1.3.1 Synchronization between Oracle Unified Directory and Oracle Internet Directory

Oracle Directory Integration Platform 11.1.1.5 and higher supports synchronization between Oracle Internet Directory and Oracle Unified Directory. For more information about the synchronization procedure, see the chapter "Integrating with Oracle Directory Server Enterprise Edition" in the *Directory Integration Platform Administrator's guide*. Oracle Directory Server Enterprise Edition was formerly known as the Sun Java System Directory Server. You need to replace all references of `SJSDS` in the guide to `OUD` for synchronization to work accurately. You can obtain Oracle Directory Integration Platform by installing Oracle Identity Management release 11.1.1.6.0 or above.

#### 1.1.3.2 Synchronization between Oracle Unified Directory and Third-Party Directories

To enable synchronization of data between Oracle Unified Directory and third-party directories, you need to integrate Oracle Directory Integration Platform with Oracle Unified Directory. You can obtain Oracle Directory Integration Platform by installing Oracle Identity Management release 11.1.1.6.0 or above.

## 1.2 Overview of Directory Server

This section provides a brief overview of the directory server component of Oracle Unified Directory server.

The Oracle Unified Directory server is an LDAPv3 compliant directory server written entirely in Java. The directory server includes the following high-level functionality:

- *Full LDAPv3 compliance (RFC 4510-4519) with support for numerous standard and experimental extensions*
- *High performance and space effective data storage*
- *Ease of configuration and administration*
  - A highly extensible administrative framework that enables you to customize most of the features listed below.
  - An administration connector that manages all administration traffic to the server. The administration connector enables the separation of user traffic and administration traffic to simplify logging and monitoring, and to ensure that administrative commands take precedence over commands that manipulate user data.
  - A graphical control panel that displays server status information and enables you to perform basic server and data administration.
  - Several command-line utilities to assist with configuration, administration tasks, basic monitoring, and data management. The main configuration utility (`dsconfig`) provides an interactive mode that walks you through most configuration tasks.

- *An advanced replication mechanism*
  - Enhanced multi-master replication across directory server instances
  - An assured replication feature that ensures high availability of data and immediacy of data availability for specific deployment requirements
  - Fractional replication capabilities
  - Support for an external change log that publicizes all changes that have occurred in a directory server database
- *An extensible security model*
  - Support for various levels of authentication and confidentiality
  - Access to resources based on privileges
  - An advanced access control mechanism

  *Multi-faceted monitoring capabilities*
- *Rich user management functionality*
  - Password policies
  - Identity mapping
  - Account status notification

## 1.3 Overview of Proxy Server

This section provides a brief overview of the proxy component of Oracle Unified Directory. The section covers the following topics:

- Section 1.3.1, "What Is the Proxy Server?"
- Section 1.3.2, "Why Use the Proxy Server?"

### 1.3.1 What Is the Proxy Server?

The Oracle Unified Directory proxy is an LDAPv3 compliant server that does not store data but routes LDAP requests from clients to the directory servers that are spread across an enterprise.

The proxy is the entry point to a directory service deployment spread over multiple directory servers and/or multiple data centers. All client requests are routed by the proxy to the appropriate remote LDAP server. The Oracle Unified Directory proxy component can be used with any LDAP v3-compliant directory server, such as the Oracle Unified Directory server or Oracle Directory Server Enterprise Edition.

In order to route data requests to the remote LDAP servers, the proxy component can be configured to use either *load balancing* or *data distribution*, or both.

You can deploy the Oracle Unified Directory proxy in very simple configurations, or in more complex, replicated scenarios, using `oud-proxy-setup`. Some simple deployments are detailed in Chapter 3, "Example Deployments Using the Proxy Server."

---

**Note:** The proxy component cannot be used directly as a datastore.

---

As the interface between the client and the remote LDAP server, the proxy provides a number of security features, to ensure secure connection if and when required. For more information about security, see Chapter 21, "Configuring Security Between the Proxy and the Data Source."

For an in-depth presentation of the elements that constitute the Oracle Unified Directory proxy, see Chapter 11, "Understanding the Proxy Functionality."

### 1.3.2 Why Use the Proxy Server?

The proxy manages all the connections between a client and a data source (be it a single server, replicated server, or data center). As such, it centralizes all the rules for client connections, including handling load balancing, data distribution and security with the data source.

When you deploy the proxy for load balancing, all requests that the proxy receives are routed to one of the remote LDAP servers based on the load balancing algorithm set during deployment. This enables you to identify the back-end directory servers that the proxy should communicate with and specify the percentage of total client load each directory server should receive. Once configured, the proxy automatically distributes client queries to different directory servers conforming to the load criteria defined in the configuration.

To deploy a *highly available* directory service, you must have at least two replicated directory servers. To ensure that requests that fail to the first server are treated by the backup server, you must ensure that all the clients know the addresses for both data sources, and are coded to treat a failure on the primary server by re-sending the request to the backup server. The proxy handles the failover and *load balancing* of requests, thereby simplifying high availability and scalability.

Typically, if your deployment used only one server to store all the data, you would have performance issues if your data store was too large. You could resolve this issue by replacing the single server with several servers, and splitting the data across these servers. In this case, each client application would need to know which server to search for its data. With the proxy, there is no need to replicate the distribution information for each application, because the proxy manages the distribution of requests to the appropriate data source. Instead, the client application sends a request to the proxy. The proxy knows which partition holds the requested data and handles the request using *distribution*.

By including the proxy in your deployment, you ease the configuration and management of client applications. The proxy centralizes and handles all requests, ensuring load balancing and/or distribution of requests.

The proxy also provides a single access point for managing security in a directory service. You can use the proxy to authorize or restrict access to remote directory servers. In addition, if you want to perform maintenance or back up an LDAP server, you can simply modify your proxy deployment to avoid service interruption.

For a description of sample deployments, see Chapter 3, "Example Deployments Using the Proxy Server."

## 1.4 Overview of the Replication Gateway

This section provides a brief overview of the replication gateway component of Oracle Unified Directory and covers the following topics:

- Section 1.4.1, "What Is the Replication Gateway?"

- Section 1.4.2, "The Role of the Replication Gateway"
- Section 1.4.3, "Limitations of the Replication Gateway"

For information about deploying the replication gateway in a migration scenario, see Section 26.11, "Replicating Between Oracle Directory Server Enterprise Edition and Oracle Unified Directory".

### 1.4.1 What Is the Replication Gateway?

Replication is the mechanism that propagates a change made on one directory server to multiple different directories in a replication topology. The replication gateway translates and propagates replication information effectively between directory servers from Oracle Directory Server Enterprise Edition and directory servers from Oracle Unified Directory. Translations are managed "on the fly" without storing any data on disk.

The main purpose of the replication gateway is to facilitate migration from an existing Oracle Directory Server Enterprise Edition deployment to an Oracle Unified Directory topology. The minimum version for this migration to succeed is Oracle Directory Server Enterprise Edition 11*g* Release 1 (11.1.1).

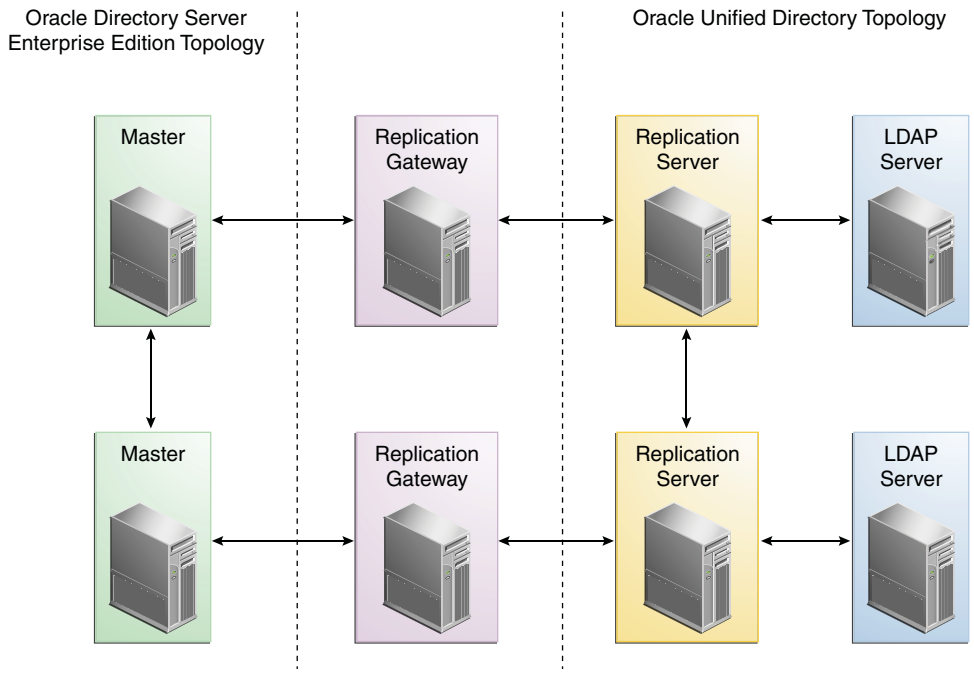
The replication gateway translates the synchronization mechanism specific to each version of the directory, offering two-way replication between the disparate topologies. The replication gateway can be regarded as a *pipe* that propagates updates between heterogeneous replicated topologies.

### 1.4.2 The Role of the Replication Gateway

The following example shows how you can transition an existing Oracle Directory Server Enterprise Edition 11g Release 1 (11.1.1) deployment to an Oracle Unified Directory topology by using the replication gateway between the two topologies.

The replication gateway is responsible for propagating changes made on the disparate servers to the entire replication topology.

Overall Replication Topology




Within the overall replication topology, the replication gateway acts as a two-way forwarding server. It propagates modifications from the Oracle Directory Server Enterprise Edition servers to the Oracle Unified Directory replication topology, and from the Oracle Unified Directory servers to the Oracle Directory Server Enterprise Edition replication topology. In each instance, the replication gateway propagates both ways. You can disable changes from being propagated from the Oracle Unified Directory servers to the Oracle Directory Server Enterprise Edition replication topology, according to your transition scenario.

For high availability, two replication gateway servers are deployed in every transition scenario.

### 1.4.3 Limitations of the Replication Gateway

The replication gateway does not manage the following aspects:

- **Data initialization.** Total update is not supported through the replication gateway. To initialize an Oracle Directory Server Enterprise Edition topology with data from an Oracle Unified Directory server, the data must be exported from the Oracle Unified Directory server and then imported to an Oracle Directory Server Enterprise Edition master server.
- **Schema coherency.** The replication gateway does not ensure that schema is coherent across the disparate servers. The administrator must define coherent schema.
- **Feature translation.** The replication gateway does not translate features between the disparate servers, and assumes that the topologies are heterogeneous, with regard to features. The best way to handle incompatible features (for example, macro ACIs, CoS, password policies) is to filter out the affected object classes and attribute types before replication occurs.

The replication gateway does provide a filtering option, for replication *from* Oracle Directory Server Enterprise Edition *to* Oracle Unified Directory. This option enables you to filter out object classes and attribute types that do not apply to Oracle Unified Directory servers. The default values that are configured for filtering take into account differences in CoS, roles, password policies and conflict resolution.

- **Replication Conflict Resolution.** In the case of single-valued attributes, if different values are added simultaneously to the same single-valued attribute, the Oracle Directory Server Enterprise Edition server and the Oracle Unified Directory server handle the conflict in different ways. The Oracle Directory Server Enterprise Edition server retains the value of the last modify/add operation while the Oracle Unified Directory server retains the oldest value. These values may not always be the same.

# 2

# Example Deployments Using the Directory Server

This chapter provides sample configurations for a replicated topology including multiple instances of the Oracle Unified Directory directory server.

This section covers the following topics:

- Section 2.1, "Small Replicated Topology"
- Section 2.2, "Multiple Data Center Topology"

For a complete understanding of how replication works in Oracle Unified Directory, see Chapter 6, "Understanding the Oracle Unified Directory Replication Model."

## 2.1 Small Replicated Topology

By replicating directory data across servers, you can reduce the access load on a single machine, improving server response time and providing horizontal read scalability. In addition, replication can be used to ensure availability of data in the event of machine failure.

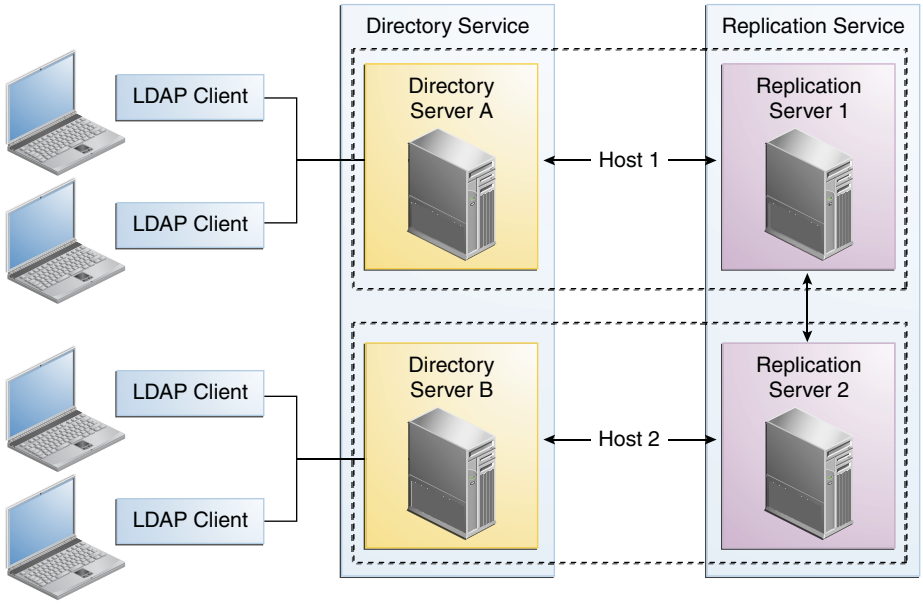
Note that you cannot use replication to scale write operations because a write operation to one directory server results in a write operation to every other server in the topology. The only way to scale write operations horizontally is to split the directory data among multiple databases and place those databases on different servers.

The centralized replication model in Oracle Unified Directory separates user data from replication metadata. In this model, the server that stores the user data is called the directory server. The server that stores the replication metadata is called the replication server. This approach simplifies the management of replication topologies and can improve performance.

For small deployments, you can set up replication by putting the replication servers and directory servers on the same system. You can further simplify administration by running the replication server and the directory server on each system in a single process.

The following figure shows how replication is used to ensure availability and to provide read scalability in a small topology.

*Figure 2–1 Basic Replication Topology*




## 2.1.1 The Role of Directory Servers in a Topology

Directory servers are responsible for the following tasks:

- Persistence of data and serving client requests
- Forwarding changes to specific replication servers

When a change is made on a directory server, that server forwards the change to a selected replication server. The replication server then replays the change to other replication servers in the topology, which in turn replay the change to all other directory servers in the topology.

Each directory server contains the following items:

- A list of the suffix DNs to be synchronized
- For each suffix DN, a list of replication servers to connect to

Applications should typically perform reads and writes on the same directory server instance. This prevents those applications from experiencing consistency problems due to loose consistency.

## 2.1.2 The Role of Replication Servers in a Topology

Replication servers are responsible for the following tasks:

- Managing connections from directory servers
- Connecting to other replication servers
- Listening for connections from other replication servers
- Receiving changes from directory servers
- Forwarding changes to directory servers and to other replication servers
- Saving changes to stable storage, which includes trimming older operations

Each replication server contains a list of all the other replication servers in the replication topology. Replication servers are also responsible for providing other servers with information about the replication topology. Even the smallest deployment must include two replication server instances, to ensure availability in case one of the replication server instances fails. There is usually no need for additional replication server instances unless the directory service must be able to survive more than one failure at a time, or unless the number of directory server instances must be very large.

Although replication servers do not store directory data, they are always LDAP servers or JMX servers. Like directory servers, replication servers can be configured, monitored, backed up and restored.

## 2.2 Multiple Data Center Topology

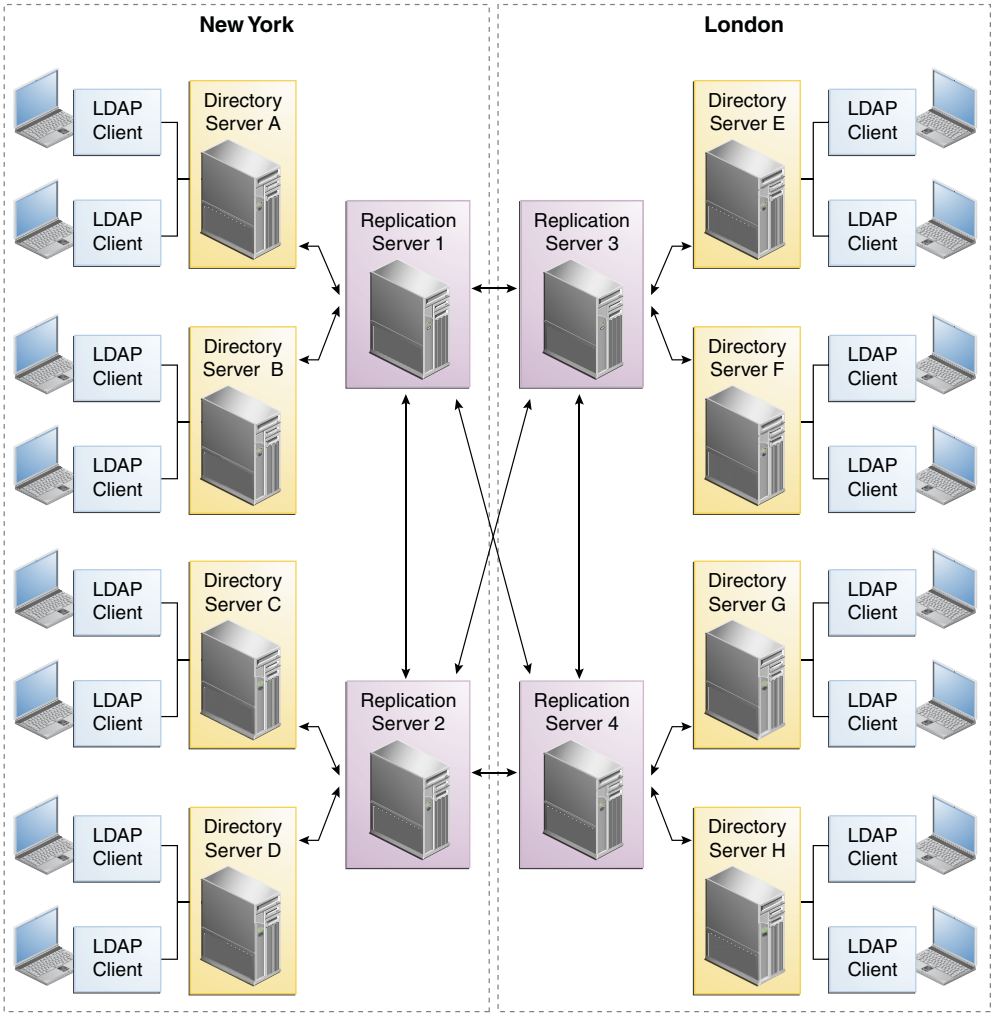
Replication enables geographic distribution of the directory service by providing identical copies of directory data on multiple servers across more than one data center. The basic principles of a replication deployment outlined in the small topology also apply to multiple data center deployments.

The Oracle Unified Directory directory server uses a custom replication protocol that is efficient over a wide area network (WAN). In the following scenario, an enterprise has two major data centers, one in London and the other in New York, separated by a WAN.

This deployment includes two replication server instances for availability in each data center, in case one of the replication server instances fails. The directory servers connect first to local replication servers. Directory servers only access replication servers in another data center if all local replication servers have failed. Client applications always connect to local directory server instances, and perform reads and writes on the same directory server instance.

The Oracle Unified Directory directory server supports an unlimited number of read/write directory servers in a replication topology. The number of directory servers can be scaled according to the read requirements of the organization. Note that increasing the number of directory servers does not scale the number of writes that can be processed because ultimately all servers in the topology must process all the writes. Unless it is acceptable to have a topology that does not converge, the write throughput of the topology is limited to the write throughput of the slowest machine.

*Figure 2–2 Multiple Data Center Topology*




## 2.2.1 Multiple Data Centers and Replication Groups

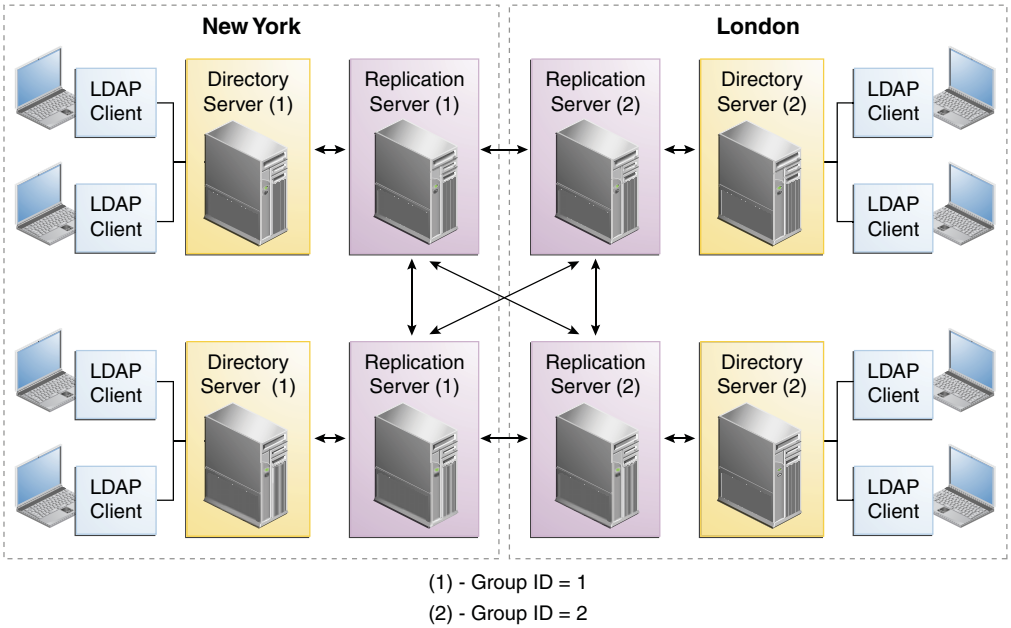
Replication groups enable you to organize a replicated topology according to specific criteria, such as data center location. A replication group is identified by a unique ID that is applied to the replication servers and the directory servers in that group. Group IDs determine how a directory server domain connects to an available replication server. From the list of configured replication servers, a directory server first tries to connect to a replication server that has the same group ID as that of the directory server.

This sample deployment shows the use of replication groups across multiple data centers. The deployment assumes two data centers, connected by a wide area network (WAN), with the following configuration:

- Each replication server and directory server within a single data center has the same group ID.
- There is a unique group ID for the entire data center (one group ID per data center).

Figure 2–3 shows a disaster recovery deployment that includes two data centers with different group IDs.

*Figure 2–3 Replication Groups Over WAN*




In this deployment, each directory server will attempt to connect to a replication server in its own data center, avoiding the latency associated with connection over a WAN. If all the replication servers in a data center fail, the directory server will connect to a remote replication server. This ensures that the replication service is maintained, albeit in a degraded manner (if the connection between data centers is slow). When one or more local replication servers is back online, the directory servers will automatically reconnect to a local replication server.

### 2.2.2 Multiple Data Centers and the Window Mechanism

The Oracle Unified Directory directory server provides a window mechanism which specifies that a certain number of update requests are sent without one server having to wait for an acknowledgement from the recipient server before continuing.

The window size represents the maximum number of update messages that can be sent without immediate acknowledgement from the recipient server. If the topology spans multiple data centers connected by a network with large latency, it might be worth increasing the window size beyond its default value of 100. To assess whether the window size is the limiting factor in replication throughput, monitor the `current-send-window` and `current-rcv-window` attributes below `cn=monitor`.

If a server publishes a `current-send-window` to another server that is consistently zero or close to zero and the corresponding server publishes a `current-rcv-window` that is higher, it means that all the data are currently in the network. In this case, increasing the window size on the recipient server should increase replication speed and reduce replication delay. These improvements will result in the consumption of more resources on the recipient server.

# 3

# Example Deployments Using the Proxy Server

There are many types of deployment in which the Oracle Unified Directory proxy can be used successfully. The following are suggested deployments, which will help familiarize you with how the proxy works.

This chapter covers the following topics:



## 3.1 Deciding Your Proxy Deployment Type

There are two main types of deployment with the proxy, namely load balancing and distribution.

To decide which type of deployment you want, consider this: where and how is your data stored and how much data do you handle?

- If all your data is stored on a replicated data store, then use a deployment with load balancing. See Configuration 1: Simple Load Balancing.
- If your data is partitioned or you have a large database and want to split your data so that it is partitioned on different data sources, then use a deployment with distribution. See Section 3.3, "Configuration 2: Simple Distribution."

More complex deployment scenarios can be defined, which layer load balancing and distribution. The main question will be, do you need load balancing, or distribution, or both?

Other than simple load balancing and simple distribution, the following example deployments will be presented:

- If you want to deploy data centers in different geographical locations, for example, you could deploy failover between two load-balanced data centers. See Section 3.4, "Configuration 3: Failover Between Data Centers."

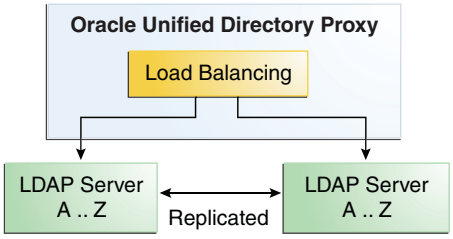
- If you want to use distribution but want the data partitions to be replicated, you can deploy the proxy server using distribution which routes to load balancer. See Section 3.5, "Configuration 4: Distribution with Load Balancing."
- If you want to use distribution with the data partitions replicated, but for availability and disaster recover you want the partitions to not only be replicated in one data center but also want to replicate the data centers in two different geographical locations, then you could deploy an architecture similar to Section 3.6, "Configuration 5: Distribution with Failover Between Data Centers."

You can add a *global index catalog* to deployments using distribution, to map entries to a specific partition. This will help minimize the use of broadcasts. For information on configuring a global index catalog, see Section 15.1.7, "Configuring Global Indexes By Using the Command Line."

## 3.2 Configuration 1: Simple Load Balancing

When you deploy the proxy for load balancing, all requests that the proxy receives are routed to one of the remote LDAP servers. As illustrated in Figure 3–1, the remote LDAP servers are replicated and contain the same data. The number of supported remote LDAP servers is not limited.

***Figure 3–1 Simple Load Balancing***




The requests are routed to one of the remote LDAP servers based on the load balancing algorithm set during deployment.

The load balancing algorithms are:

- `failover`
- `generic`
- `optimal`
- `proportional`
- `saturation`
- `searchfilter`

For more information on the different load balancing algorithms, see Section 11.1, "Load Balancing Using the Proxy."

The algorithm can be bypassed by a client connection affinity. If you set client connection affinity, the proxy uses the load balancing algorithm for the first request, but for the following request will disregard the load balancing algorithm set and will try to reuse the same route for a new operation on the same client connection, for example, depending on the type of client affinity set. For more information, see Section 15.1.3.5.7, "Setting Client Connection Affinity."

The advantages of using load balancing deployment are the high availability of the data, as well as an adapted workload on the remote LDAP servers. For example, if one of the remote LDAP servers in your configuration becomes unavailable, the load balancing will route the request to another remote LDAP server. In this case, the failure is not visible to the client and there is no service disruption.

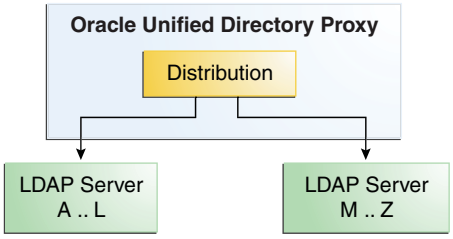
A simple load balancing deployment can be configured easily during the proxy installation.

## 3.3 Configuration 2: Simple Distribution

When you deploy the proxy for simple distribution, the data is split into partitions. Each partition of data is held on a separate remote LDAP server, as illustrated in Figure 3–2. Here, LDAP Server A...L is a server that holds entries for users whose names start with A through L. Similarly, LDAP Server M…Z holds entries for users whose names start with M through Z. All requests that the proxy receives are routed to the remote LDAP server which contains the appropriate data.

The number of remote LDAP servers onto which the data is partitioned depends on the size of the database that you are splitting. Figure 3–2 shows simple distribution algorithm with two partitions, but you can configure more.

***Figure 3–2 Simple Distribution***




The requests are routed to one of the remote LDAP servers based on the distribution algorithm set during deployment.

The distribution algorithms are:

- `capacity`
- `numeric`
- `lexico`
- `dnpattern`

For more information on the different distribution algorithms, see Section 11.2, "Data Distribution Using the Proxy."

The advantage of a deployment using distribution is that you can scale the number of updates per second. To diminish the number of broadcasts when using distribution, you can add a global index catalog. For information about the global index catalog, see Section 15.1.7, "Configuring Global Indexes By Using the Command Line."

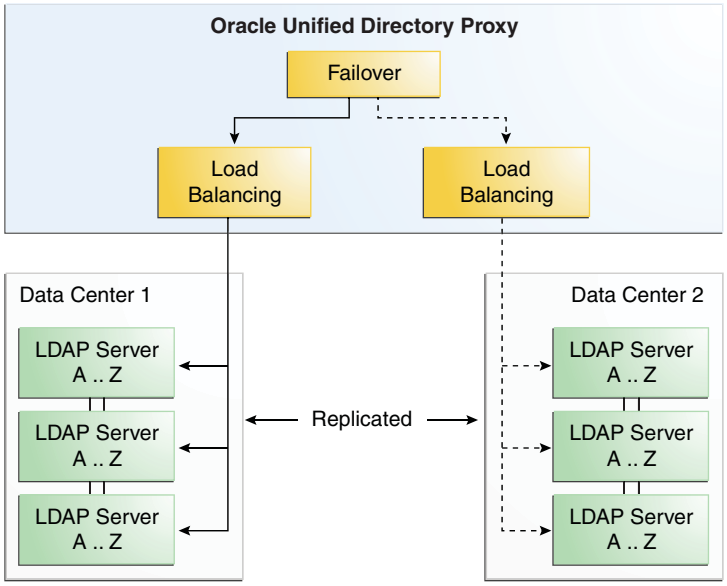
A simple distribution deployment can be easily configured during the proxy installation.

## 3.4 Configuration 3: Failover Between Data Centers

When you configure failover between data centers, you are essentially deploying two levels of load balancers within the proxy. In this deployment, the data centers are replicated and the remote LDAP servers within the data centers are also replicated. The first load balancing element of the deployment can be either failover or saturation. The example assumes failover algorithm is selected for the initial load balancing element.

As illustrated in Figure 3–3, all of the requests are routed by the failover load balancer through the main route, to a second load balancing element, which sends the request to a server within Data Center 1. Here, LDAP Server A...L is a server that holds entries for users whose names start with A through L. If Data Center 1 goes down or is degraded, then the traffic is routed by the failover load balancer to the backup route, to a server in Data Center 2.

***Figure 3–3 Failover Between Data Centers***




The requests are routed to the remote LDAP servers within the data centers based on the load balancing algorithm set. The load balancing algorithm can be different for each data center. For example, you can set the load balancing in Data Center 1 as `proportional`, while the load balancing algorithm in Data Center 2 is set as `saturation`.

This type of deployment is typically used when deploying in two geographical areas. This adds high availability of data to a simple load balancing deployment, since not only are the remote LDAP servers replicated, but the data centers are also replicated.

Typically, you would have the two data centers in two different geographical locations. This way, if there was a problem in one location, the data center in the other location would act as backup. Another example would be setting the first load balancer to `saturation`. This way, if Data Center 1 in one geographical location (for example in one time-zone) becomes saturated, the other data center can pick up the excess traffic.

For more information on the different load balancing algorithms, see Load Balancing Using the Proxy.

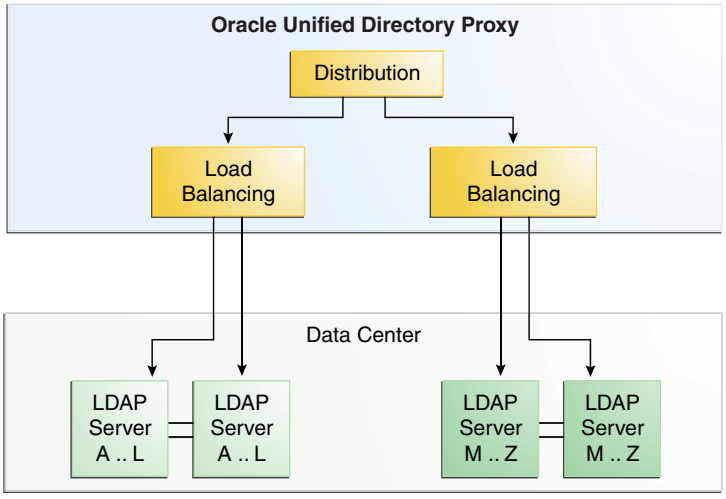
For details on deploying this configuration, see Failover Load Balancing.

## 3.5 Configuration 4: Distribution with Load Balancing

In a deployment that includes distribution and load balancing, the data is split into partitions, and the data is replicated on the remote LDAP servers. Requests sent to the proxy are first distributed to the partition in which the data is stored, then the request is routed to one of the remote LDAP servers, depending on the load balancing algorithm set. The remote LDAP servers holding the partitioned data are replicated.

As illustrated in Figure 3–4, when the proxy receives a requests, it is filtered by the distribution to the correct partition. Here, LDAP Server A...L is a server that holds entries for users whose names start with A through L. Similarly, LDAP Server M…Z holds entries for users whose names start with M through Z. For example, a request for entry with a cn such as `Garry` would be forwarded to partition 1, to the servers with data from `A..L`. The load balancer then forwards the request to one of the replicated remote LDAP servers.

***Figure 3–4 Distribution with Load Balancing***




The requests are routed to the remote LDAP servers within the data centers based on the load balancing algorithm set. For more information on the different load balancing algorithms, see Section 11.1, "Load Balancing Using the Proxy."

The advantages of this deployment are the speed of the updates, because of the distribution of data, and high availability of the data.

For more information on the different distribution algorithms, see Section 11.2, "Data Distribution Using the Proxy."

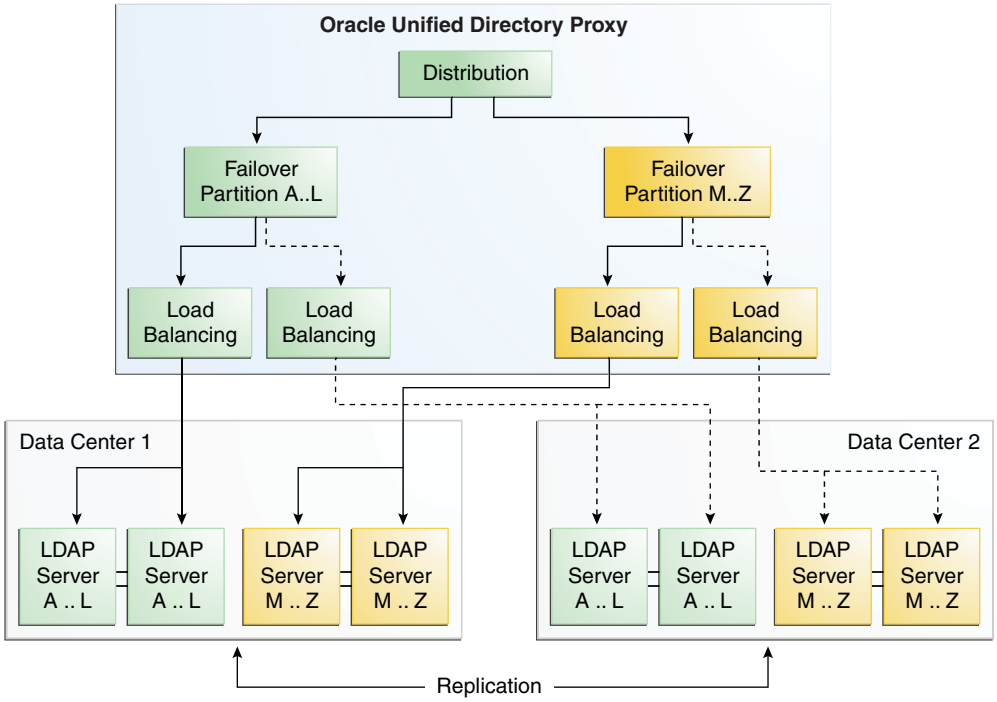
For more information on the different load balancing algorithms, see Section 11.1, "Load Balancing Using the Proxy."

For details on deploying this configuration, see Chapter 11, "Understanding the Proxy Functionality."

## 3.6 Configuration 5: Distribution with Failover Between Data Centers

In a deployment that includes distribution with failover load balancing between two data centers, the data is split into partitions, where each partition is managed through a failover load balancing route. As illustrated in Figure 3–5, not only are the remote LDAP servers holding the partitioned data replicated within the data center, but in addition, the data centers are replicated, with one of the two acting as the backup. Here, LDAP Server A...L is a server that holds entries for users whose names start with A through L. Similarly, LDAP Server M…Z holds entries for users whose names start with M through Z.

***Figure 3–5 Distribution with Failover Between Data Centers***




In other words, requests sent to the proxy are first distributed to the partition in which the data is stored. For example, a request for entry with a cn such as `Garry` would be forwarded to partition 1. The failover load balancer then forwards the request through the main route, depending on the load balancing algorithm set, to one of the one of the remote LDAP servers holding the data for `A..L`.

In the deployment illustrated in Figure 3–5, Data Center 2 acts as a backup, and is only used on failure of the first data center. However, this same deployment could be configured to use `saturation`, rather than a failover load balancer. This way, if Data Center 1 in one geographical location (for example in one time-zone) becomes saturated, the other data center can pick up the excess traffic.

The advantages of this deployment are the speed of the reads through the distribution algorithm, and the high availability offered since the remote LDAP servers are replicated, and one data center acts as a backup.

For more information on the different distribution algorithms, see Section 11.2, "Data Distribution Using the Proxy."

For more information on the different load balancing algorithms, see Section 11.1, "Load Balancing Using the Proxy."

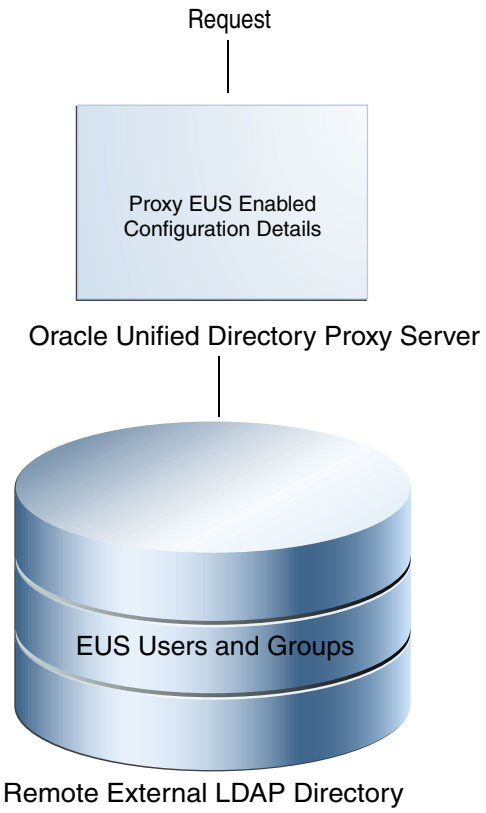
For details on deploying this configuration, see Section 16.5, "Configuring Distribution with Failover Between Data Centers."

## 3.7 Configuration 6: Enterprise User Security

When you deploy the proxy for Enterprise User Security (EUS), the configuration details are stored locally in Oracle Unified Directory and the remote external LDAP directory contains only the Enterprise Users and the Enterprise Groups details.

As illustrated in Figure 3–6 the remote external LDAP directory contains only the Enterprise Users and the Enterprise Groups details.

***Figure 3–6 Proxy Enterprise User Security***




The requests are routed to one of the remote LDAP servers based on the load balancing algorithm set during deployment.

The load balancing algorithms are:

- `failover`
- `optimal`
- `proportional`
- `saturation`

For more information on the different load balancing algorithms, see Section 11.1, "Load Balancing Using the Proxy."

To configure this proxy deployment complete the following steps:

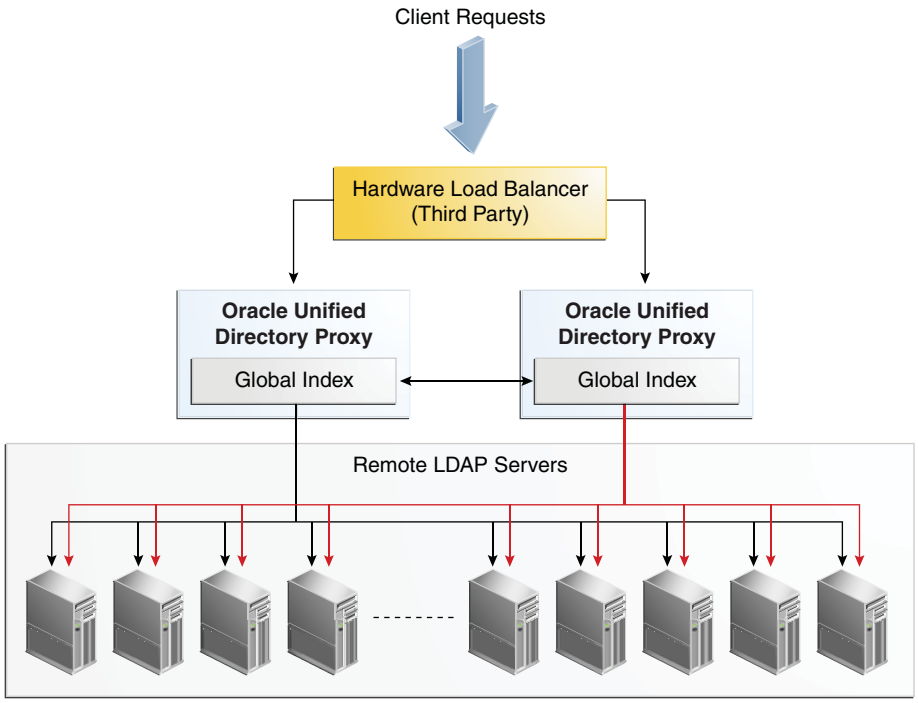
1. Setup the proxy server, as described in the "To Configure Enterprise User Security" section in the *Oracle Fusion Middleware Installation Guide for Oracle Unified Directory*.
2. Integrate Oracle Unified Directory and Enterprise User Security, as described in Chapter 25, "Integrating With Oracle's Enterprise User Security".

## 3.8 Multiple Replicated Proxies

To prevent a Single Point of Failure, you should ensure that your deployment is redundant. Typically, this can be done by installing a third party hardware load balancer, as illustrated in Figure 3–7.

Using a hardware load balancer, you can manage multiple proxy instances on separate physical machines or in different geographical locations.

***Figure 3–7 Multiple Proxy Instances***




When running multiple proxy instances in a distribution deployment with a global index catalog, the global index catalog should be replicated. For more information on replicating the global index catalog, see Section 15.1.7.2, "Replication of Global Index Catalogs."

To configure this proxy deployment, see "Setting Up the Proxy Server" section in the *Oracle Fusion Middleware Installation Guide for Oracle Unified Directory.*

# Part II

# Oracle Unified Directory Concepts and Architecture

This part describes the details of how Oracle Unified Directory works. These chapters cover the architecture of Oracle Unified Directory and the various components that make up that architecture.

In general, you do not need a thorough understanding of all of these concepts in order to administer Oracle Unified Directory, but an overview of these chapters might help to make your administration easier.

This part includes the following chapters:

# 4

# Understanding Oracle Unified Directory Concepts and Architecture

Oracle Unified Directory is a next-generation unified directory solution that integrates storage, synchronization, and proxy functionality to help you manage the critical identity information that drives your business applications. These capabilities enable you to meet the evolving needs of an enterprise architecture.

This chapter provides conceptual descriptions of the basic components of Oracle Unified Directory and discusses Oracle Unified Directory architecture. This chapter covers the following topics:

- Section 4.1, "Oracle Unified Directory Components"
- Section 4.2, "Architecture of Oracle Unified Directory"

## 4.1 Oracle Unified Directory Components

Oracle Unified Directory integrates three key components: Network Groups, Workflows, and Workflow Elements. This section provides an overview of each component and contains the following topics:

- Section 4.1.1, "Network Groups"
- Section 4.1.2, "Workflows"
- Section 4.1.3, "Workflow Elements"

### 4.1.1 Network Groups

Network groups are the entry point of all client requests handled by Oracle Unified Directory.

Network groups handle all client interactions and dispatch them to local backend workflows or proxy workflows, based on:

- Criteria

  Criteria can include security authentication level, port number, client IP mask, client bind DN, bind ID, domain name, and other criteria.

- Quality of Service (QoS) policies

  QoS policies can include LDAP referral policy, request filtering, client connection affinity, and resource limits.

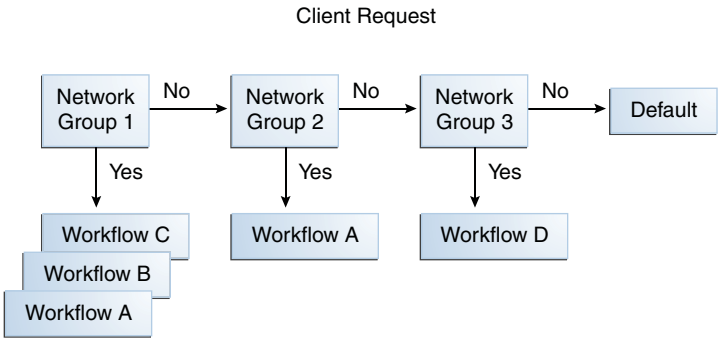
You can define more than one network group, each with different properties and different priorities. However, an incoming client connection can only be attached to

one network group at a time. An incoming client connection is attached to the first network group for which the connection complies with the criteria defined for that network group.

The client connection is assessed by each network group, in order of priority, until it complies with all the criteria of that network group. As illustrated in Figure 4–1, the request is first sent to the network group with the highest priority: `Network Group 1`. `Network Group 1` assesses if the request matches all the required criteria. If it does not match all of the criteria, it forwards the request to the next network group in the list: `Network Group 2`.

If the request matches all the properties of a network group, the network group assesses if the client connection matches the QoS policies of that network group. If it matches the QoS policies, it is routed to the associated *workflow*.

***Figure 4–1 Network Group Selection***




A network group can be associated with one or more workflows, each workflow corresponding to a different naming context. For more information about workflows, see Section 4.1.2, "Workflows". If the client connection matches the criteria of a network group, but not the QoS policies of that network group, the connection is not forwarded to the workflow, nor is it sent to the next network group. Instead, an error is returned, indicating the QoS policy that caused the error.

If a network group has no workflows attached to it, the request is not handled. Instead, the server returns an error message of the sort: `No such entry`.

For information about managing network groups, see Section 14.1.6, "Configuring Network Groups With `dsconfig`".

***Example 4–1 Using Network Group Criteria to Route to Different Workflows***

Assume an Oracle Unified Directory configuration with the following network groups:

- `Network Group 1`: criteria set with bind DN `**,dc=example,dc=com`

  This network group is associated with `Workflow 1`, with naming context `dc=example,dc=com`

- `Network Group 2`: criteria set with bind DN `**,dc=test,dc=com`

  This network group is associated with `Workflow 2`, with naming context `dc=test,dc=com`

Depending on the bind DN, a search would be routed through `Network Group 1` or `Network Group 2`. For example, if the bind DN was `uid=user.1,dc=test,dc=com`, the request would not be accepted by `Network`

Group 1, but would be forwarded to and accepted by Network Group 2, and forwarded to Workflow 2.

#### *Example 4–2 Using a Network Group QOS Policy to Filter Requests*

Assume an Oracle Unified Directory configuration with the following network groups:

- Network Group 1: criteria set with bind DN **,ou=admin,dc=example,dc=com

  QoS policy set with resource limits size limit=0, time limit=0. Therefore, for admin group, there are no limits.

  This network group is associated to Workflow 1, with naming context dc=example,dc=com.

- Network Group 2: criteria set with bind DN **,dc=example,dc=com

  QoS policy set with resource limits size limit=100, time limit=30 s. Therefore, for all connections other than admin group, there are limits set on the resources used.

  This network group is also associated to Workflow 1, with naming context dc=example,dc=com.

Therefore, as long as the bind DN is dc=example,dc=com, the requests will be forwarded to Workflow 1. The QoS policy set for Network Group 2 gives restricted access to Workflow 1, for anyone that is not admin. Anyone who binds as admin will access Workflow 1 through Network Group 1, and will have no limitations on resource limits.

### 4.1.2 Workflows

A workflow is defined by a *naming context* (base DN) and a workflow element that define how Oracle Unified Directory should handle an incoming request. A workflow must be registered with at least one network group, but can be attached to several network groups.

A network group can point to *several* workflows if the naming contexts of the workflows are different. However, several network groups can point to the *same* workflow when the network group QoS policies are different, but the naming context of the workflow is the same.

#### *Example 4–3 A Network Group Routing to Several Workflows*

Assume an Oracle Unified Directory configuration with the following network groups (as illustrated in Figure 4–1), where:

- Network Group 1 with a bind DN of **,l=fr,dc=example,dc=com

  This network group is associated to Workflow 1, with naming context l=fr,dc=oracle,dc=com

- Network Group 2 with a bind DN of **,l=uk,dc=example,dc=com

  This network group is associated to Workflow 2, with naming context l=uk,dc=example,dc=com

- Network Group 3 with a bind DN of **,dc=example,dc=com

  This network group is associated to Workflow 1 and Workflow 2, with naming context dc=example,dc=com

A search with bind DN `**,l=uk,dc=oracle,dc=com` would be handled by `Network Group 2` and sent to `Workflow 2`.

A search with bind DN `**,dc=oracle,dc=com` would be handled by `Network Group 3` and sent to `Workflow 1` and `Workflow 2`.

### 4.1.3 Workflow Elements

Each workflow contains at least one *workflow element*. Workflow elements are part of a routing structure.

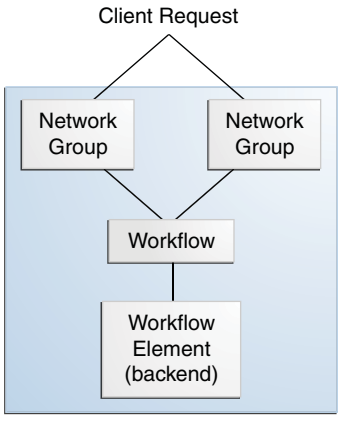
Oracle Unified Directory supports several different types of workflow elements:

- Leaf workflow elements: This comprises the local backend workflow elements and proxy workflow elements.
- Routing workflow elements: This comprises the load balancing workflow elements and distribution workflow elements.
- Virtual workflow element: This comprises the DN renaming workflow elements, RDN changing workflow elements, and Transformation workflow elements.
- EUS workflow element: This comprises the Enterprise User Security (EUS) workflow elements.
- EUS context workflow element: This comprises the EUS context workflow elements.
- LDIF workflow element: This comprises the LDIF local backend workflow elements.
- Memory backend workflow element: This comprises the memory local backend workflow elements.

For a directory server, the workflow element is the DB local backend, as illustrated in Figure 4–2.

For a proxy server, the workflow elements can be chained with load balancing workflow elements or distribution workflow elements that act as a pointer, routing the request along a specific path. The proxy workflow element provides direct access to the remote data source.

***Figure 4–2 Client Request for a Directory Server***




Oracle Unified Directory has a number of preconfigured workflow elements that should not be modified or deleted.

## 4.2 Architecture of Oracle Unified Directory

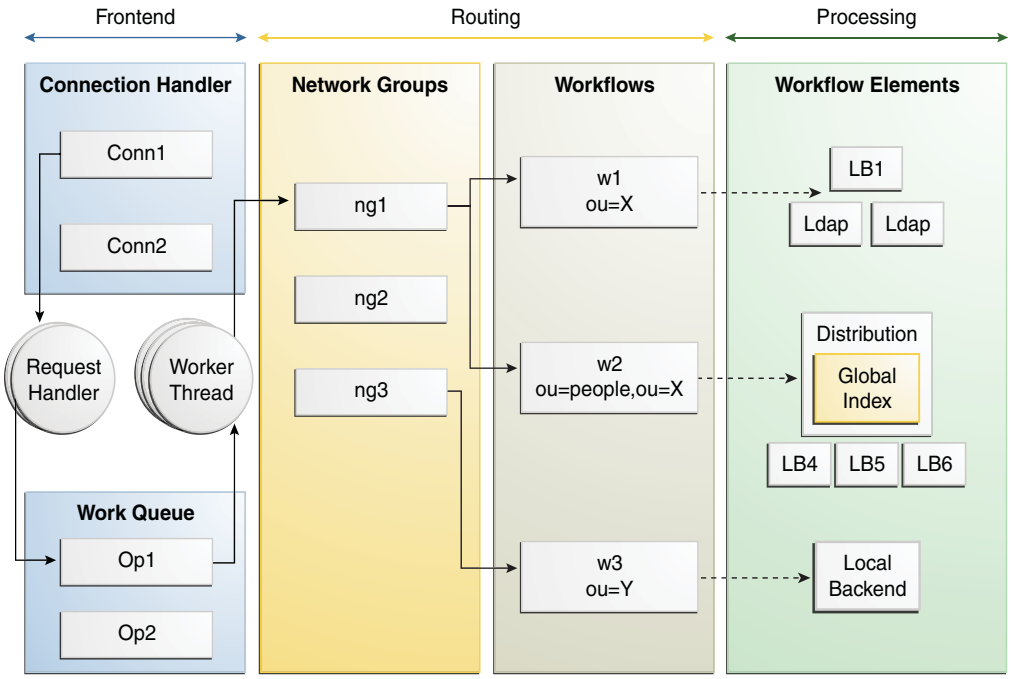
This section presents the high-level architecture of Oracle Unified Directory.

As illustrated in Figure 4–3, a client request is managed by Oracle Unified Directory before being forwarded to the data source. In this scenario, there are three network groups, such as ng1, ng2, and ng3. The first network group ng1 contains two workflows while ng3 contains a single workflow. A workflow is defined by a suffix. The suffix for `w1` is `ou=X` and a workflow points to a tree of workflow elements. The tree of workflow elements determines the processing to apply on an operation.

A client request pursues the following path:

1. The request handlers place the incoming LDAP requests in the work queue from where the worker thread grabs them.
2. The operation is routed to a network group based on the network group criteria assigned. An operation must comply with the network group QoS policies regardless of the server profile, directory server or proxy server.
3. The network group forwards the operation to a workflow, which defines the naming context. The determination of the workflow is based on the match between the request base DN and the workflow naming context.
4. The workflow forwards the operation to its tree of workflow elements, which defines how to treat the request. The content of the tree of workflow elements depends on the server profile as follows:
   - For a directory server, you can only configure the workflow element as the local backend workflow element (a storage).
   - For a proxy server, you can configure the workflow element as a distribution workflow element, a load balancing workflow element, a DN renaming workflow element, or an LDAP proxy workflow element.
5. After the request has gone through the assigned processing, the request is sent to the data source.

***Figure 4–3 High-Level Presentation of Oracle Unified Directory Components***



# 5

# Understanding Oracle Unified Directory High Availability Deployments

As more and more businesses and mission-critical applications connect with identities being centrally managed, it has become imperative to have LDAP service available all the time. High availability in conjunction with performance has become the distinguishing feature of all extranet and enterprise deployments.

This chapter contains the following topics:



## 5.1 What is High Availability?

High availability is a system design approach and its associated implementation that ensures an agreed level of operational performance during a given measurement period for your directory service.

Agreed service levels vary from one organization to another. Service levels also depend on several factors such as the time of day systems are accessed, whether or not systems can be brought down for maintenance, and the cost of downtime to the organization. Failure or downtime in this context, is defined as periods when a system is unavailable and prevents from providing the agreed level of service.

Oracle Unified Directory provides elaborate cost-effective and easy-to-use high availability features, which eliminate the downtime and maximize the time when the system is available.

## 5.2 Availability and Single Points of Failure

Oracle Unified Directory deployments that provide highly available service can recover from failures and maintain service within agreed level of service. With a high availability deployment, component failures might impact individual directory queries but does not result into a complete system failure.

A single point of failure (SPOF) is a system component, which on failure renders an entire system unavailable or unreliable. When you design a highly available deployment, you identify potential SPOFs and investigate how to mitigate these SPOFs.

This section contains the following topics:

## 5.2.1 Types of SPOFs

SPOFs can be divided into three categories:



### 5.2.1.1 Hardware Failure

You can broadly categorize the hardware SPOFs as follows:

- Network failures
- Failure of the physical servers on which Directory Server or Directory Proxy Server are running
- Hardware load balancer failures
- Storage subsystem failures
- Power supply failures

### 5.2.1.2 Software Failure

Directory server or proxy server failure can be categorized as follows:

- Slow response time
- Write overload
  - Maximized file descriptors
  - Maximized file system
  - Poor storage configuration
  - Too many indexes
- Read overload
- Cache issues
- CPU constraints
- Replication issues
  - Out of sync
  - Replication propagation delay
  - Replication flow
  - Replication overload
- Large wildcard searches

## 5.2.2 Common Approach to Mitigate SPOFs - Redundancy

You can implement redundancy to ensure that failure of a single component does not cause an entire directory service to fail. Redundancy involves providing redundant software components, hardware components, or both. Examples of this strategy include deploying multiple, replicated instances of Directory Server on separate hosts and/or using redundant arrays of independent disks (RAID) for storage of Directory

Server data. Redundancy with replicated Directory Servers is the most efficient way to achieve high availability.

## 5.3 Using Redundancy for High Availability

To ensure reliability and continued services for directory service, you must maintain a high level of system availability, with a seamless transition to redundant systems during a system failure.

Redundancy works for both Directory and proxy servers and allows you to mitigate:

- Hardware failures, because the traffic can be redirected to another hardware component.
- Software failures, when the failure cannot be reproduced systematically.

Redundancy handles failure in the following ways:

- Section 5.3.1, "Redundancy at the Hardware Level"
- Section 5.3.2, "Redundancy at Directory Server Level Using Replication"
- Section 5.3.3, "Using Directory Proxy Server as Part of a Redundant Solution"
- Section 5.3.4, "Using Application Isolation for High Availability"
- Section 5.3.5, "Using Replication Gateway for High Availability"

### 5.3.1 Redundancy at the Hardware Level

This section provides an overview of hardware redundancy. Many publications provide comprehensive information about using hardware redundancy for high availability. You must specifically, see "Blueprints for High Availability" published by John Wiley & Sons, Inc.

Failure at the network level can be mitigated by having redundant network components. When designing your deployment, consider having redundant components for the following:

- Internet connection
- Network interface card
- Network cabling
- Network switches
- Gateways and routers

You can mitigate the hardware load balancer as an SPOF by including a redundant hardware load balancer in your architecture.

You can mitigate against SPOFs in the storage subsystem by using redundant server controllers. You can also use redundant cabling between controllers and storage subsystems, redundant storage subsystem controllers, or redundant arrays of independent disks.

If you have only one power supply, loss of this supply could make your entire service unavailable. To prevent this situation, consider providing redundant power supplies for hardware, where possible, and diversifying power sources. Additional methods of mitigating SPOFs in the power supply include using surge protectors, multiple power providers, and local battery backups, and emergency local power generators.

Failure of an entire data center can occur if, for example, a natural disaster strikes a particular geographic region. In this instance, a well-designed multiple data center replication topology can prevent a distributed directory service from becoming unavailable. For more information, see Section 5.4, "Sample Topologies Using Redundancy for High Availability."

### 5.3.2 Redundancy at Directory Server Level Using Replication

A common method to implement redundancy in Oracle Unified Directory Servers is to use Replication. Redundant solutions are usually less expensive, easier to implement, and easier to manage than clustering solutions. This is because in clustering model you often have to configure at least two servers to serve the same application workload, one node is active while the other is passive, on standby. Note that replication, as part of a redundant solution, has numerous functions other than availability. While the main advantage of replication is the ability to split the read across multiple servers, this advantage needs to be balanced with the task to manage the additional server. Replication also offers scalability on read operations and, with proper design, scalability on write operations, within certain limits. For an overview of replication concepts, see Chapter 6, "Understanding the Oracle Unified Directory Replication Model."

The SPOFs described in Section 5.2.1.2, "Software Failure" can be mitigated by having redundant instances of Directory Server. This involves the use of replication. Replication ensures that the redundant servers remain synchronized, and that requests can be rerouted with no downtime.

Replication is used to prevent the loss of a single server from causing your directory service to become unavailable. A reliable replication topology ensures that the most recent data is available to clients across data centers, even in the case of a server failure. At a minimum, your local directory tree needs to be replicated to at least one backup server. Directory architects recommend you to replicate three times per physical location for maximum data reliability. When the data is replicated at least thrice, then in the event of a failure of a directory server the configuration remains highly available and protected against failure. In deciding how much to use replication for fault tolerance, consider the quality of the hardware and networks used by your directory. Unreliable hardware requires more backup servers.

The Oracle Unified Directory replication model is a loosely consistent, multi-master model. In other words, all directory servers in a replicated topology can process both read and write operations. For more information about replication, see Chapter 6, "Understanding the Oracle Unified Directory Replication Model."

Do not use replication as a replacement for a regular data backup policy. Replication is designed to maintain service within a given service level agreement. It is not designed to protect against incorrect data stored in the directory by applications or users. For information about backing up directory data, see Section 17.3, "Backing Up and Restoring Data."

To maintain the ability to read data in the directory with the expected Service Level Agreement, a suitable load balancing strategy must be put in place. Both software and hardware load balancing solutions exist to distribute read load across multiple replicas. Each of these solutions can also determine the state of each replica and to manage its participation in the load balancing topology. The solutions might vary in terms of completeness and accuracy.

To maintain write failover over geographically distributed sites, you can use multiple data center replication over WAN. This entails setting up at least two master servers in each data center, and configuring the servers to be fully meshed over the WAN. This

strategy prevents loss of service if any of the masters in the topology fail. Write operations must be routed to an alternative server if a writable server becomes unavailable.

### 5.3.3 Using Directory Proxy Server as Part of a Redundant Solution

Directory Proxy Server is designed to support high availability directory deployments. The proxy provides automatic load balancing as well as automatic failover and fail back among a set of replicated Directory Servers. Should one or more Directory Servers in the topology become unavailable, the load is proportionally redistributed among the remaining servers.

Proxy servers can also be made redundant by using several instances of proxy. Yet another approach to provide highly available directory service.

Directory Proxy Server actively monitors the Directory Servers to ensure that the servers are still online. The proxy also examines the status of each operation that is performed. Servers might not all be equivalent in throughput and performance. If a primary server becomes unavailable, traffic that is temporarily redirected to a secondary server is directed back to the primary server as soon as the primary server becomes available.

Note that when data is distributed, multiple disconnected replication topologies must be managed, which makes administration more complex. In addition, Directory Proxy Server relies heavily on the proxy authorization control to manage user authorization. A specific administrative user must be created on each Directory Server that is involved in the distribution. These administrative users must be granted proxy access control rights.

### 5.3.4 Using Application Isolation for High Availability

Directory Proxy Server can also be used to protect a replicated directory service from failure due to a faulty client application. To improve availability, a limited set of masters or replicas is assigned to each application.

Suppose a faulty application causes a server shutdown when the application performs a specific action. If the application fails over to each successive replica, a single problem with one application can result in failure of the entire replicated topology. To avoid such a scenario, you can restrict failover and load balancing of each application to a limited number of replicas. The potential failure is then limited to this set of replicas, and the impact of the failure on other applications is reduced.

### 5.3.5 Using Replication Gateway for High Availability

The replication gateway propagates changes between Oracle Directory Server Enterprise Edition and Oracle Unified Directory topologies. The replication gateway is designed to provide a highly available deployment solution by allowing you to use redundant replication gateway servers for propagating changes made on disparate servers to the entire replication topology. For more information about replication gateway, see Section 1.4.2, "The Role of the Replication Gateway."

## 5.4 Sample Topologies Using Redundancy for High Availability

The following sample topologies show how redundancy and replication is used to provide continued service in the event of failure:

- Chapter 2, "Example Deployments Using the Directory Server"

- Chapter 3, "Example Deployments Using the Proxy Server"
- Section 15.2.1, "Configuring Load Balancing With ODSM"
- Section 15.1.7.2, "Replication of Global Index Catalogs"

# 6

# Understanding the Oracle Unified Directory Replication Model

Oracle Unified Directory replication uses a loosely consistent multi-master model. All directory servers that are part of a replication topology can accept read and write operations.

The following architectural topics are targeted at developers and at users who want to understand the internal of the replication mechanism. It is not necessary to read these topics just to be able to use replication. For information about configuring and using replication, see Chapter 26, "Replicating Directory Data".

The following topics describe the architecture of the Oracle Unified Directory replication functionality.

- Section 6.1, "Overview of the Replication Architecture"
- Section 6.2, "How Replication Works"
- Section 6.3, "Historical Information and Conflict Resolution"
- Section 6.4, "Schema Replication"
- Section 6.5, "Replication Status"
- Section 6.6, "Replication Groups"
- Section 6.7, "Assured Replication"
- Section 6.8, "Fractional Replication"

## 6.1 Overview of the Replication Architecture

The Oracle Unified Directory replication model is a loosely consistent, multi-master model. In other words, all directory servers in a replicated topology can process both read and write operations.

Replication is built around a centralized publish-subscribe architecture. Each directory server communicates with a central service, and uses the central service to publish its own changes and to receive notification about changes on other directory servers. This central service is called the *replication service*.

The replication service can be made highly available by using multiple server instances running on multiple hosts. Within the replication architecture, a server instance that provides the replication service is called a *replication server*. A server instance that provides the directory service is called a *directory server*.

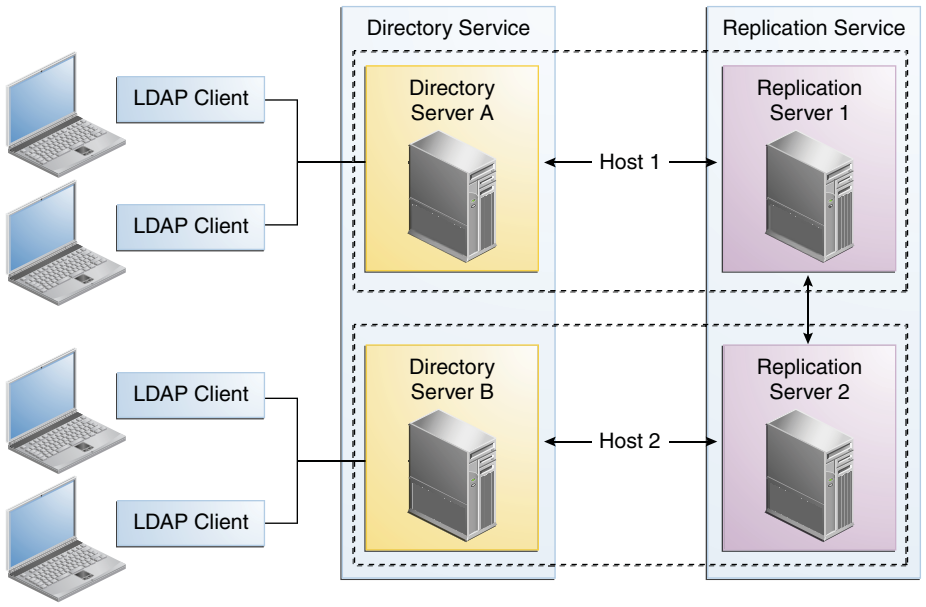
The parties in a replication session authenticate to each other using SSL certificates. A connection is accepted if the certificate that is presented is in the ADS trust store. No access control or privileges are enforced.

The topics in this section describe the replication architecture and the various elements that make up this architecture.

- Section 6.1.1, "Basic Replication Architecture"
- Section 6.1.2, "Replication Servers"
- Section 6.1.3, "Replication Change Numbers"
- Section 6.1.4, "Replication Server State"
- Section 6.1.5, "Operation Dependencies"

## 6.1.1 Basic Replication Architecture

The basic replication architecture is shown in the following illustration.




At startup time, each directory server selects a single replication server and connects to it. The directory server sends all changes that it processes to that replication server, and receives all changes from other servers in the topology through that replication server. Each replication server is connected to every other replication server in the topology.

When a replication server receives a change from a directory server, the replication server forwards the change to all the other replication servers in the topology. These replication servers in turn forward the change to all the directory servers to which they are connected. When a replication server receives a change from another replication server, the replication server forwards the change to the directory servers to which it is connected, but not to other replication servers. A directory server never sends a change directly to another directory server. This architecture ensures that all changes are forwarded to all servers without requiring complex negotiation.

Every change is assigned a *change number* by the directory server that originally processed the change. The change number is used to identify the change throughout its processing. A replication server maintains changes in persistent storage so that

older changes can be resent to directory servers that were not connected when the change occurred or that fell behind, becoming temporarily unable to receive all the changes at the time they were processed. For more information, see Section 6.1.3, "Replication Change Numbers".

The current update state of each directory server is maintained by keeping a record of the last changes that the directory server processed. When a directory server connects to a replication server, the replication server uses this record to determine the first change in the list of updates to send to the directory server.

Because multiple directory servers can process updates simultaneously, an update operation on one directory server can conflict with another update operation that affects the same entries on another directory server. Each directory server resolves conflicts when it replays operations from other directory servers, so that all directory server data eventually converges.

Conflicts can occur because of conflicting modify operations, called *modify conflicts*. Conflicts can also occur because of conflicting add, delete, or modRDN operations, called *naming conflicts*. To resolve conflicts in a coherent way, directory servers maintain a history of successive changes to each entry. This history is called *historical information*. Historical information is stored as an operational attribute inside the entry on which the changes occurred. For more information, see Section 6.3, "Historical Information and Conflict Resolution".

### 6.1.2 Replication Servers

A replication server performs the following tasks:

- Manages connections from directory servers
- Connects to other replication servers
- Listens for connections from other replication servers
- Receives changes from directory servers
- Forwards changes to directory servers and other replication servers
- Saves changes to stable storage, which includes trimming older operations

Replication servers are not the same as directory servers. However, like directory servers, replication servers use a configuration file, can be configured and monitored online, and can be backed up and restored. Replication servers are therefore always LDAP servers or JMX servers, even though replication servers do not store directory data.

When you configure a directory server instance for replication, a replication server is created automatically, unless you specify otherwise. The replication server and the directory server can run in the same JVM, or in separate JVMs.

In a small topology (up to four directory servers) it makes sense to configure each server to function as both a directory server and a replication server. In a large topology (more than twenty directory servers) it is advisable to separate the directory server and replication server instances into separate JVMs, and to limit the number of replication servers.

Between these two extremes, you can decided on the configuration that works best for your requirements. Having all servers functioning as both directory servers and replication servers is generally a simpler topology and easier to administer. Separating the directory servers and replication servers lowers the disk requirements of the directory server instances because they do not need to store a replication change log.

### 6.1.3 Replication Change Numbers

Change numbers uniquely identify changes that are made on an LDAP directory server. Change numbers also provide a consistent ordering of changes. The change number order is used to resolve conflicts and to determine the order in which forwarded changes should be replayed.

A *change number* consists of the following elements:

- **Time stamp, in milliseconds.** Time stamps are generated using the system clock. The change number is also generated such that each change number is always greater than all the change numbers that have already been processed by the server. Constantly increasing change numbers guarantees that operations that depend on previous operations are consistently replayed in the correct order. An example of an operation that depends on a previous operation is a modify operation that directly follows the add operation for that entry.
- **Sequence number.** A sequential number, increment for each change that occurs within the same millisecond.
- **Replica identifier.** A unique integer identifier that is assigned to each replica in a topology. (A *replication topology* is the set of all replicas of a given data set. For example, the replication topology for `example.com` might be all copies of the `dc=example,dc=com` suffix across a directory service.)

  The replica identifier ensures that two different servers do not assign the same identifier to two different changes. In a future directory server release, an algorithm might be used to assign replica identifiers automatically.

### 6.1.4 Replication Server State

When a directory server connects to a replication server, the replication server must determine how up to date the directory server data is before the replication server can send changes that the directory server has not yet seen. This "up to date" state of the directory server is called the *replication server state*.

A server might have missed relatively old changes from another remote server, yet might already have seen and processed more recent changes from a server that is close by. Server state is therefore maintained by recording the last change number processed by each replica, according to the replica identifier.

Because administrators can stop and restart servers, the server state must be saved to stable storage. Ideally saving the server state would be done after each local or replicated change is made. Saving information to the database after each change would add significant overhead, however. Server state is therefore kept in memory and saved to the database on a regular basis, and when the server is properly shut down.

In the event of brutal interruptions such as kills and crashes can cause the server to lose track of changes that have already been processed. This can result in the need to fix inconsistencies when the server restarts. For an explanation of how crash recovery is managed, see Section 6.2.6, "Directory Server Crashes."

### 6.1.5 Operation Dependencies

Sometimes an operation cannot be replayed until another operation is complete. For example, when an add operation is followed by a modify operation on the same entry, the server must wait for the add operation to be completed before starting the modify operation.

Such dependencies are quite rare and are generally necessary for a few operations only. Usually operations do not have dependencies, since they are modify operations. Therefore, in such cases, it is necessary to replay operations in parallel to obtain the best performance with multi-CPU servers.

The replication model is built on the assumption that operation dependencies are rare. The replication mechanism therefore always tries to replay operations in parallel, and only switches to processing operation dependencies if an operation fails to replay.

## 6.2 How Replication Works

The topics in this section describe the mechanics involved in the replication process and how specific functionality is achieved.

- Section 6.2.1, "Replication Initialization"
- Section 6.2.2, "Directory Server Change Processing"
- Section 6.2.3, "Replication Server Selection"
- Section 6.2.4, "Change Replay"
- Section 6.2.5, "Auto Repair"
- Section 6.2.6, "Directory Server Crashes"
- Section 6.2.7, "Replication Server Crashes"

### 6.2.1 Replication Initialization

Before a server can participate in a replicated topology, that server must be initialized with data. That is, a complete data set must be copied onto the server in some way. For information about the ways in which a server can be initialized with data, see Section 26.4, "Initializing a Replicated Server With Data."

#### 6.2.1.1 Replicating Configuration Data Manually

Replication is automatic for data, but it has to be manually triggered for configuration.

Oracle Unified Directory configuration is specified in the file instance-path/config/oud.ldif. This section lists the specific configuration attributes that you have to replicate from the old instance to the new instance manually.

You can migrate the values of the following configuration attributes:

- Global configuration attributes, for instance writability mode, size and time limit, and so on.
- Security configuration attributes, for instance crypto manager, key manager, trust manager, ID mapping, and SASL.
- Connection handlers.
- Performance tuning attributes, for instance cache, threads, and other database configuration parameters.
- Replication configuration attributes.
- Password policy configuration attributes.
- Plug-In configuration attributes.
- Feature configuration attributes, for instance identity mapping, indexes, and so on.

## 6.2.2 Directory Server Change Processing

When a modification occurs on a directory server, replication code on the directory server performs the following tasks:

- Assigns a change number
- Generates historical information
- Forwards the change to a replication server
- Updates the server state

Historical information is stored in the entry and must therefore be included in the operation before the server writes to the back end. The server uses the change number when generating historical information. The change number is therefore generated before the historical information. Both the change number and the historical information are performed as part of the pre-operation phase.

The operation is sent to the replication server before an acknowledgment for the update is sent to the client application that requested the operation. This ensures that a synchronous, assured replication mode can be implemented. For more information, see Section 6.7, "Assured Replication". The acknowledgment is therefore sent as part of the post-operation phase.

Changes are sent in the order defined by their change numbers. The correct order enables replication servers to make sure that all the changes are forwarded to other directory servers.

Because a directory server is multi-threaded, post-operation plug-ins can be called in a different order to pre-operation plug-ins, for the same operation. The replication code maintains a list of *pending changes*. This list includes changes that have started, and for which change numbers have already been generated, but that have not yet been sent to the replication server. Changes are added to the list of pending changes in the pre-operation phase. Changes are removed from the list when they are sent to the replication server. If a specific operation reaches the post-operation phase ahead of its change number-defined position, that operation waits until previous operations are sent before being sent to the replication server.

The server state is updated when the operation is sent to the replication server. For more information, see Section 6.1.4, "Replication Server State."

## 6.2.3 Replication Server Selection

When a directory server starts (or when the replication server to which it is connected is stopped), the directory server selects a suitable replication server for publishing and receiving changes. This section describes how the replication server is selected.

### 6.2.3.1 Replication Server Selection Algorithm

The directory server uses the following principles to select a suitable replication server:

- **Filtering.** To begin, the directory server creates a list of eligible replication servers, from all of the configured replication servers in the topology. The list is created based on the following criteria:

  1. Replication servers that have the same group ID (or geographic identifier) as the directory server.
  2. Replication servers that have the same generation ID (initial data set) as the directory server.

3. Replication servers that include all of the latest updates that were generated from the directory server.
4. Replication servers that run in the same virtual machine as the directory server.

---

**Note:** These criteria are listed in order of preference. So, for example, if a replication server has the same generation ID (criterion 2) as the directory server but does not have the same group ID (criterion 1), it will not be included in the list, unless no replication server in the topology has the same group ID as the directory server.

---

- **Load Balancing.** When the directory server has compiled a list of eligible replication servers, it selects a replication server in a manner that balances the load across all the replication servers in the topology. This selection is made in accordance with the *replication server weight* in the topology. For more information, see Section 6.2.3.2, "Replication Server Load Balancing."

#### 6.2.3.2 Replication Server Load Balancing

In large topologies with several directory servers and several replication servers, it is more efficient to spread the directory servers out across the replication servers in a predefined manner. This is particularly important if the replication servers are running on different types of machines, with different capabilities. If the estimated "global power" of the machines differs significantly from one replication server to another, it is useful to balance the load on the replication servers according to their power.

You can configure the proportional *weight* of a replication server so that the number of directory servers connecting to each replication server is balanced efficiently. Replication server weight is defined as an integer (`1..n`). Each replication server in a topology has a default weight of 1. This weight only has meaning in its comparison to the weights of other replication servers in the topology.

The replication server weight determines the proportion of the directory servers currently in the topology that should connect to this particular replication server. The replication server weight is configured as a fraction of the estimated global power of all the replication servers in the topology. For example, if replication server A is estimated to be twice as powerful as replication server B, the weight of replication server A should be twice the weight of replication server B.

The weight of a particular replication server can be represented as $({}^{n}/_{d})$ where $n$ is the weight of the replication server and $d$ is the sum of the weights of all the replication servers in the topology.

For information about configuring the replication server weight, see Section 26.3.12, "Configuring the Replication Server Weight."

### 6.2.4 Change Replay

The replay of changes on replicated directory servers is efficient on multi-core and multi-CPU systems. On a directory server, multiple threads read the changes sent by the replication server.

Dependency information is used to decide whether an operation can be replayed immediately. The server checks the server state and the list of operations on which the

current operation depends to determine whether those operations have been replayed. If the operations have not been replayed, the server puts the operation in a queue that holds dependency operations. If the operation can be replayed, the server builds an internal operation from information sent by replication servers. The server then runs the internal replay operation.

Internal replay operations built from the operations that are sent by a replication server can conflict with prior operations. Such internal operations cannot therefore always be replayed as if they were taking place on the original directory server. The server checks for conflicts when processing the *handleConflictResolution* phase.

In the majority of cases, the internal replay operations do not conflict with prior operations. In such cases, the handleConflictResolution phase does nothing. The replication code is therefore optimized to return quickly.

When a conflict does occur, the handleConflictResolution code takes the appropriate action to resolve the conflict. For modify conflicts, the handleConflictResolution code changes the modifications to retain the most recent changes.

When conflict resolution is handled, historical information is updated as for local operations. The operation can then be processed by the core server. Finally, at the end of the operation, the server state is updated.

After completing an operation, the server thread processing the operation checks whether an operation in the dependency queue was waiting for the current operation to complete. If so, that operation is eligible to be replayed, so the thread starts the replay process for the eligible operation. If not, the thread listens for further operations from the replication server.

### 6.2.5 Auto Repair

Despite efforts to keep servers in sync, directory servers can begin to show incoherent data. Typically, this occurs in the following circumstances:

- A disk error taints the stored data
- A memory error leads to an error in processing data
- A software bug leads to bad data or missing changes

In such cases, tracking and replaying changes is not sufficient to synchronize the incoherent data.

An automatic repair mechanism is provided, which leverages historical information inside entries to determine what the coherent data should be. The replication mechanism then repairs the data on directory servers where the data is bad or missing. The *auto repair* mechanism is implemented as an LDAP application, and runs on the hosts that run replication servers.

The auto repair application can run in different modes. Depending on the mode in which it is run, the auto repair application performs the following tasks:

- Repairs inconsistencies manifested as an error when the server was replaying modifications
- Repairs inconsistencies detected by the administrator
- Periodically scans directory entries to detect and repair inconsistencies

**Note:** In the current directory server release, the auto repair mechanism must be run manually. For more information, see Section 26.8, "Detecting and Resolving Replication Inconsistencies."

### 6.2.6 Directory Server Crashes

If a directory server crashes, its connection to the replication server is lost. Recent changes that the directory server has processed and committed to its database might not yet have been transmitted to any replication server.

When a directory server restarts, therefore, it must compare its state with the server state of the replication servers to which the directory server connects. If the directory server detects that changes are missing and not yet sent to a replication server, the directory server constructs fake operations from historical information. The directory server sends these fake operations to its replication server.

Because the local server state is not saved after each operation, the directory server cannot trust its saved server state after a crash. Instead, it recalculates its server update state, based on historical information.

### 6.2.7 Replication Server Crashes

If a replication server crashes, directory servers connect to another replication server in the topology. The directory servers then check for and, if necessary, resend missing changes.

## 6.3 Historical Information and Conflict Resolution

The topics in this section describe how historical information is retained and used to resolve replication conflicts.

- Section 6.3.1, "What is a Replication Conflict?"
- Section 6.3.2, "Resolving Modify Conflicts"
- Section 6.3.3, "Resolving Naming Conflicts"
- Section 6.3.4, "Purging Historical Information"

### 6.3.1 What is a Replication Conflict?

A conflict occurs when one or more entries are updated simultaneously on multiple servers and the changes are incompatible, or causes some interaction between the updates. Conflict occurs because no update operation is carried out simultaneously on every replica in the replication topology. Instead, updates are first processed on one server, then replicated to other servers.

The following example describes a conflict that occurs when an attribute is modified at the same time on two different directory servers.

Consider a topology with two read-write replicas. A modify operation changes the surname, `sn`, attribute of an entry to `Smith` on one server. Before the server that is processing the change can synchronize with the other server, the `sn` attribute value for that entry is replaced with the value `Jones` on the other server. Unless the conflict is managed, replication would replay the change (`Smith`) on the server that now contains the value `Jones`. At the same time, replication would replay the change (`Jones`) on the server that contains the value `Smith`. The servers would therefore end up with inconsistent values for the `sn` attribute on the modified entry.

The following list describes additional conflicts that can occur.

- An entry is deleted on one server while one of its attribute values is modified on another server.
- An entry is renamed on one server while one of its attribute values is remodified on another server.
- An entry is deleted and another entry with the same Distinguished Name (DN) is added on one server while one of its attribute values is modified on another server.
- A parent entry is deleted and a child of that entry is created on another server, either through an add operation or a rename operation.
- Two different entries with the same DN are added at the same time on two different servers.
- Two different values are used to replace a single-valued attribute on the same entry on different servers at the same time.

Conflicts that involve only modifications of the same entry are called *modify conflicts*. Conflicts that involve at least one operation other than modify are called *naming conflicts*.

All modify conflicts and the vast majority of naming conflicts can be solved automatically by replaying the operations in their order of occurrence. However, the following naming conflicts, which have very little chance of occurring, cannot be solved automatically.

- Two entries with the same DN are created at the same time on different servers, either by adding new entries or by renaming existing entries.
- A parent entry is deleted and a child of the parent entry is created at the same time. The child entry can be created either when a new entry is added or when an existing entry is renamed.

### 6.3.2 Resolving Modify Conflicts

Modify conflicts only occur with modification operations.

Operations are globally and logically ordered to determine the outcome of a given set of operations. Change numbers are used to define the order.

The replication conflict resolution functionality ensures that all servers eventually reach the same state, as if all operations were replayed everywhere in the order defined by the change numbers. This remains true even though changes might be replayed in a different order on different servers. In the modify conflict example with the `sn` values of `Smith` and `Jones`, described previously, assume that the value was set to `Jones` on the second server *after* it was set to `Smith` on the first server. The resulting attribute value should be `Jones` on both servers, even after the replace modification of `Smith` is replayed on the second server.

Historical information about each entry is retained to check whether a conflicting operation has already been played using a change number newer than that of a current conflicting operation. For each modify operation, historical information is used, first to check if there is a conflict, and, if there is a conflict, to determine the correct result of the operation.

When a modify conflict occurs, the server determines whether the current attribute values must be retained or whether the modification must be applied. The current attribute values alone are not sufficient to make this assessment. The server also

determines when (at which change number) prior modifications were made. Historical information therefore includes the following elements:

- The date when the attribute was last deleted
- The date when a given value was last added
- The date when a given value was last deleted

When an attribute is deleted or fully replaced, older information is no longer relevant. At that point the older historical information is removed.

Historical information undergoes the following processing:

- Saved in the `ds-sync-hist` attribute (This attribute can only be viewed by an administrator.)
- Updated (but not used) for normal operations
- Updated and used for replicated operations

Conflict resolution is carried out when operations are replayed, after the pre-operation during the `handleConflictResolution` phase.

Conflict resolution is carried out by changing the `List<Modification>` field of the `modifyOperation` to match the actual modifications required on the user attributes of the entry, and to change the `ds-sync-hist` attribute that is used to store historical information.

### 6.3.3 Resolving Naming Conflicts

Naming conflicts only happen for replayed operations. The server uses the following methods to resolve naming conflicts:

- Uses unique IDs to identify entries, including entries that have been renamed
- Tries to replay each operation first and only takes action if a conflict occurs
- Checks during the pre-operation phase for conflicts that cannot be detected when operations are replayed
- Retains no *tombstone entries*, which are entries that have been marked for deletion but not yet removed

Because directory entries can be renamed, the DN is not an immutable value of the entry. DNs cannot therefore be used to identify the entry for replication purposes. A unique and immutable identifier is therefore generated when an entry is created, and added as an operational attribute of the entry. This unique ID is used, instead of the DN, to identify the entry in changes that are sent between directory servers and replication servers.

A replication context is attached to the operation. The replication context stores private replication information such as change number, entry ID, and parent entry ID that is required to solve the conflict.

### 6.3.4 Purging Historical Information

Historical information is stored in the server database. Historical information therefore consumes space, I/O bandwidth, and cache efficiency. Historical information can be removed as soon as more recent changes have been seen from all the other servers in the topology.

Historical information is purged in the following ways:

- When a new change is performed on the entry.
- By a purge process that can be triggered at regular intervals. The purge process saves space, at the cost of more CPU for processing the purge. The purge process is therefore configurable. For more information, see Section 26.3.2, "Changing the Replication Purge Delay."

## 6.4 Schema Replication

This section describe how schema replication is implemented. and is aimed at users who require an in-depth understanding of the schema replication architecture.

Schema describe the entries that can be stored in a directory server. Schema management is a core feature of the directory service. Replication is also a central feature of the directory service and is essential to a scalable, highly available service.

Any changes made to the schema of an individual directory server must therefore be replicated on all the directory servers that contribute to the same service.

Schema replication occurs when the schema is modified in any of the following ways:

- By modifying the `cn=schema` suffix when the server is online
- By using a dedicated task to perform dynamic schema updates by means of a file when the server is online
- By modifying the underlying back-end files directly when the server is offline

Generally, schema modifications occur only when deploying new applications or new types of data. The rate of change for schema is therefore low. For this reason, the schema replication implementation favors simplicity over scalability.

Schema replication is enabled by default. In certain specific cases, it might be necessary to have different schema on different directory servers, even when the servers share all or part of their data. In such cases you can disable schema replication, or specify a restricted list of servers that participate in schema replication. For more information, see Section 26.6, "Configuring Schema Replication."

### 6.4.1 Schema Replication Architecture

The schema replication architecture relies on the general replication architecture. You should therefore have an understanding of the general replication architecture before reading this section. For more information, see Section 6.1, "Overview of the Replication Architecture."

Directory servers notify replication servers about any changes to their local schema. As in the case of data replication, the replication servers propagate schema changes to other replication servers, which in turn replay the changes on the other directory servers in the topology.

Schema replication shares the same replication configuration used for any subtree:

```
dn: cn=example,cn=domains,cn=Multimaster Synchronization,\
  cn=Synchronization Providers,cn=config
objectClass: top
objectClass: ds-cfg-replication-domain
cn: example
ds-cfg-base-dn: cn=schema
ds-cfg-replication-server: <server1>:<port1>
ds-cfg-replication-server: <server2>:<port2>
ds-cfg-server-id: <unique-server-id>
```

Schema replication differs from data replication in the following ways:

- **Entry Unique ID.** A unique ID is required for data replication because entries can be renamed or deleted.

  In the case of the schema, there is only one entry and that entry cannot be deleted or renamed. The unique ID used for the schema entry is therefore the DN of the schema entry.

- **Historical information.** Historical information is used to save a history of relevant modifications to an entry. This information makes it possible to solve modification conflicts.

  For schema replication, the only possible operations are adding values and deleting values. Historical information is therefore not maintained for modifications to the schema.

- **Persistent server state.** When a directory server starts up, the replication plug-in establishes a connection with a replication server. The replication server looks for changes in its change log and sends any changes that have not yet been applied to the directory server.

  In order to know where to start in the change log, the replication plug-in stores information that is persistent across server stop and start operations. This persistent information is stored in the replication `base-dn` entry.

  The schema back end allows the specific operational attribute used to store the persistent state, `ds-sync-state`, to be modified.

## 6.5 Replication Status

Each replicated domain in a replicated topology has a certain *replication status*, depending on its connections within the topology, and on how up to date it is with regard to the changes that have occurred throughout the topology.

Knowledge of a domain's replication status enables a replicated topology to do the following:

- Manage certain aspects of assured replication
- Enable certain administrative tasks
- Administer and monitor replication effectively

For more information, see Section 29.7, "Monitoring a Replicated Topology."

The following sections outline the different statuses that a replicated domain can have.


- Section 6.5.1, "Replication Status Definitions"
- Section 6.5.2, "Degraded Status"
- Section 6.5.3, "Full Update Status and Bad Generation ID Status"


### 6.5.1 Replication Status Definitions

The following list provides a description of each possible replication status that can be held by a replicated domain.

`NOT_CONNECTED_STATUS`
The local replicated domain is not connected to any replication server. Replication cannot occur until a connection to a replication server is established. This is the only possible status if there is no connection to a replication server.

NORMAL_STATUS
The local replicated domain is almost in sync with its peers (that is, with the updates received on the replication server). The client LDAP requests have been processed normally.

DEGRADED_STATUS
The local replicated domain is too late regarding updates that have been queued by the replication server. What constitutes *too late* is defined by the *degraded status threshold*, that is, the number of changes that the replication server has in its queue for the directory server. With this status, the local directory server might be slow in replaying changes. This can have an impact on assured replication.

FULL_UPDATE_STATUS
An online full update is currently being performed on the local replicated domain (in other words, the domain is receiving entries from a remote directory server). The full update must be completed before the status can be changed and before the replicated domain can participate in replication again.

BAD_GEN_ID_STATUS
The local replicated domain does not have the same generation ID as the replication server to which it is connected. Replication cannot run until the local domain is initialized with a data set that has the same generation ID as its replication server. To initialize the local domain, perform an online full update, an LDIF import, or a binary copy of the database, retaining the domain entries.

### 6.5.2 Degraded Status

A directory server that is slow in replaying changes is assigned a DEGRADED_STATUS. The stage at which the server is regarded as "too slow" is defined by the *degraded status threshold* and is configurable, based on the number of updates queued in the replication server for that directory server.

When the degraded status threshold is reached, the directory server assumes a degraded status and is considered to be unable to send acknowledgments in time. A server with this status can have an impact on assured replication, as replication servers no longer wait for an acknowledgment from this server before returning their own acknowledgments.

### 6.5.3 Full Update Status and Bad Generation ID Status

Apart from being assigned a degraded status, a directory server can change status if an administrator performs one of the following tasks on the topology:

- **Full update.** When a replicated domain is initialized online from another server in the topology, the directory server status for that domain changes to FULL_UPDATE_STATUS. When the full update has completed, the directory server reinitializes its connection to the topology, and the status is reset to NORMAL_STATUS.
- **Local import or restore.** When a replicated domain is reinitialized by using a local import or restore procedure, the directory server status for that domain changes to NOT_CONNECTED_STATUS.
- **Resetting the generation ID.** If a replicated domain connects to a replication server with a generation ID that is different from its own, the domain is assigned a BAD_GEN_ID status. A domain can also be assigned this status if a reconnection occurs after a full online update, a local import, or a restore with a set of data that has a different generation ID to that of the replication server.

In addition, you might need to reset the generation ID of all the replication servers in the topology by running the reset generation ID task on the directory server. This causes all the replication servers in the topology to have a different ID to the ID of the directory servers to which they are connected. In this case, the directory servers are assigned a `BAD_GEN_ID` status.

## 6.6 Replication Groups

*Replication groups* are designed to support multi-data center deployments and disaster recovery scenarios. Replication groups are defined by a group ID. A group ID is a unique number that is assigned to a replicated domain on a directory server (one group ID per replicated domain). A *group ID* is also assigned to a replication server (one group ID for the whole replication server).

Group IDs determine how a directory server domain connects to an available replication server. From the list of configured replication servers, a directory server first tries to connect to a replication server that has the same group ID as that of the directory server. If no replication server with a compatible group ID is available, the directory server connects to a replication server with a different group ID. This selection process is described in greater detail in the following section.

For information about how to configure replication groups, see Section 26.3.8, "Configuring Replication Groups."

**Note:** Assured replication does not cross group boundaries. For more information, see Section 6.7, "Assured Replication."

## 6.7 Assured Replication

Before you read the following sections, you should have an understanding of basic replication concepts. You must know what a replication server is, as opposed to a directory server, and have an understanding of how replication servers work in a replicated topology. If this is not the case, read at least the Section 6.1, "Overview of the Replication Architecture" to obtain an understanding of how regular replication works in the directory server.

In a standard replicated topology, changes are replayed to other replicated servers in a "best effort" mode. A change made on an LDAP server is replayed on the other servers in the topology as soon as possible, but in an unsynchronized manner. This is convenient for performance but does not ensure that a change has been propagated to other servers when the initial LDAP client call is finished.

In some deployments this might be acceptable, that is, the time period between the change on the first server and the replay on peer servers might be short enough to fulfill the requirements of the deployment. For example, an international organization might store employee user accounts in a replicated topology across various geographical locations. If a new employee is hired and a new account is created for him on one LDAP server in a specific location, it might be acceptable that the replay of the creation occurs in other LDAP servers a few milliseconds after the LDAP client call terminates. The user is unlikely to perform a host login that would access one of the other LDAP servers in the same second that the user account is created.

However, there might be cases in which more synchronization is required from the replication process. If a specific host fails, it might be imperative that any changes made on that host have been propagated elsewhere in the topology. In addition, the deployment might require assurance that once the LDAP client call of a change is

returned by a server, all of the peer servers in the topology have received that change. Any other clients that read the entry from anywhere in the topology would be sure to obtain the modification.

*Assured replication* is a method of making regular replication work in a more synchronized manner. The topics in this section describe how assured replication works, from an architectural perspective. For information about configuring assured replication, see Section 26.3.9, "Configuring Assured Replication."

The following sections describe the implementation of assured replication:

- Section 6.7.1, "Assured Replication Modes"
- Section 6.7.2, "Assured Replication Connection Algorithm"
- Section 6.7.3, "Assured Replication and Replication Status"
- Section 6.7.4, "Assured Replication Monitoring"

## 6.7.1 Assured Replication Modes

The directory server currently supports two assured replication modes, depending on the level of synchronization that is required, the goal of the replicated topology, and the acceptable performance impact.

- Section 6.7.1.1, "Safe Data Mode"
- Section 6.7.1.2, "Safe Read Mode"
- Section 6.7.1.3, "Safe Read Mode and Replication Groups"

### 6.7.1.1 Safe Data Mode

In safe data mode, any change is propagated to a specified number of servers in the topology before the LDAP client call returns. If the LDAP server on which the change was made fails, it is guaranteed that the change has already been propagated to at least the specified number of servers.

This specified number of servers (N) defines the *safe data level*. The safe data level is based on acknowledgments from the replication servers only. In other words, an update message that is sent from an LDAP server must be acknowledged by at least N (N>=1) replication servers before the LDAP client call that initiated the update returns.

The higher the safe data level, the greater the number of machines that are assured to have the update and, consequently, the more reliable the data. However, as the safe data level increases, the overall performance decreases because additional acknowledgments are required before the LDAP client call returns.

The safe data level functions in best effort mode. That is, if the safe data level is set to 3 and there are temporarily only two replication servers available in the topology, an acknowledgment from the third (unavailable) replication server will not be expected until this server is available again.

Safe data mode is affected by the use of *replication groups*. Because assured replication does not cross group boundaries, a replication server with a group ID of 1 waits for an acknowledgment from other replication servers with the same group ID but not for acknowledgments from replication servers with a different group ID. For more information, see Section 6.6, "Replication Groups."

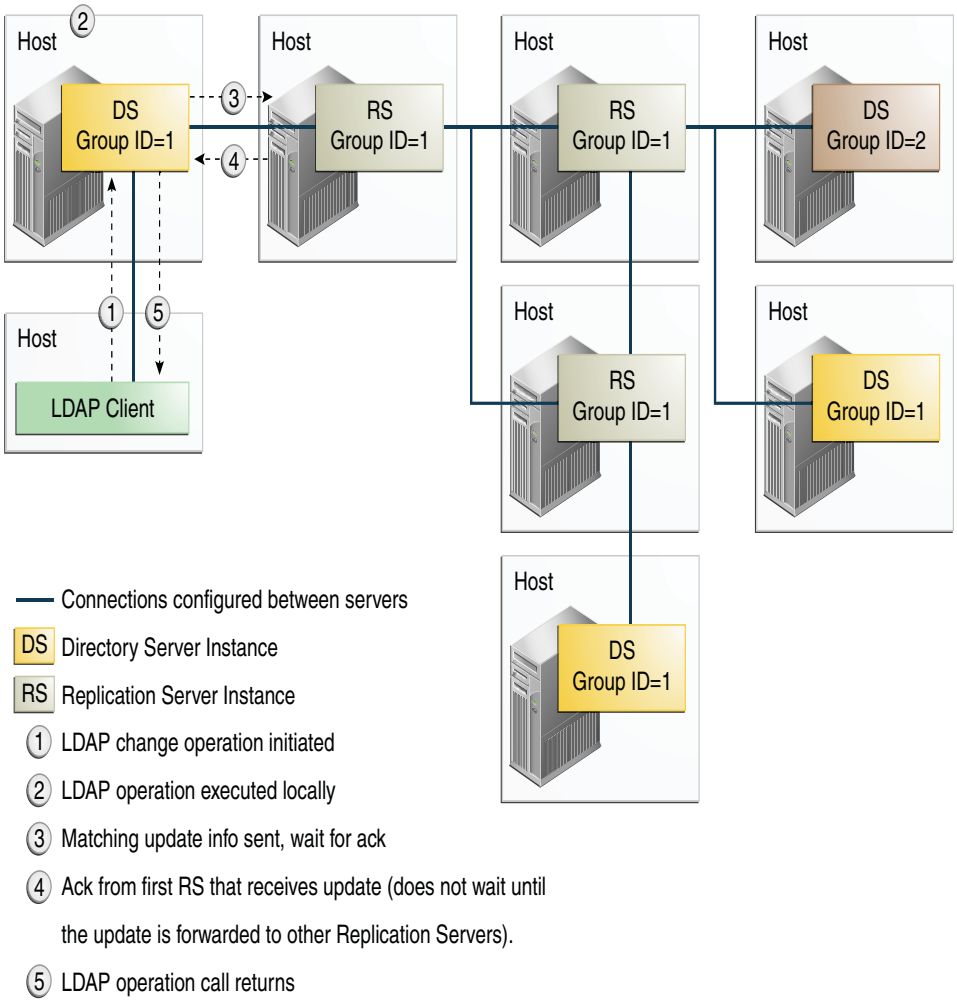
**Note:** In the current replication implementation, the `setup` and `dsreplication` commands support only a scenario in which the main replication server is physically located in the same VM as the LDAP server (that is, on the same machine). However, the fundamental replication design is to support deployments where the replication servers run on separate machines, to increase reliability.

Such deployments can currently be configured only by using the `dsconfig` command and are not supported by the `setup` and `dsreplication` commands. However, these deployments provide better failover and availability, and are expected to be supported in the future. In such deployments, if the safe data level is set to 1 (acknowledgment of only one replication server is expected), this replication server *must* run on a separate machine to the LDAP server.

***Example 6–1 Safe Data Level = 1***

Setting the safe data level to 1 ensures that the first replication server returns an acknowledgment to the directory server immediately after receiving the update. The replication server does not wait for acknowledgments from other replication servers in the topology. The modification is guaranteed to exist on one additional server (other than the directory server on which the change was made).

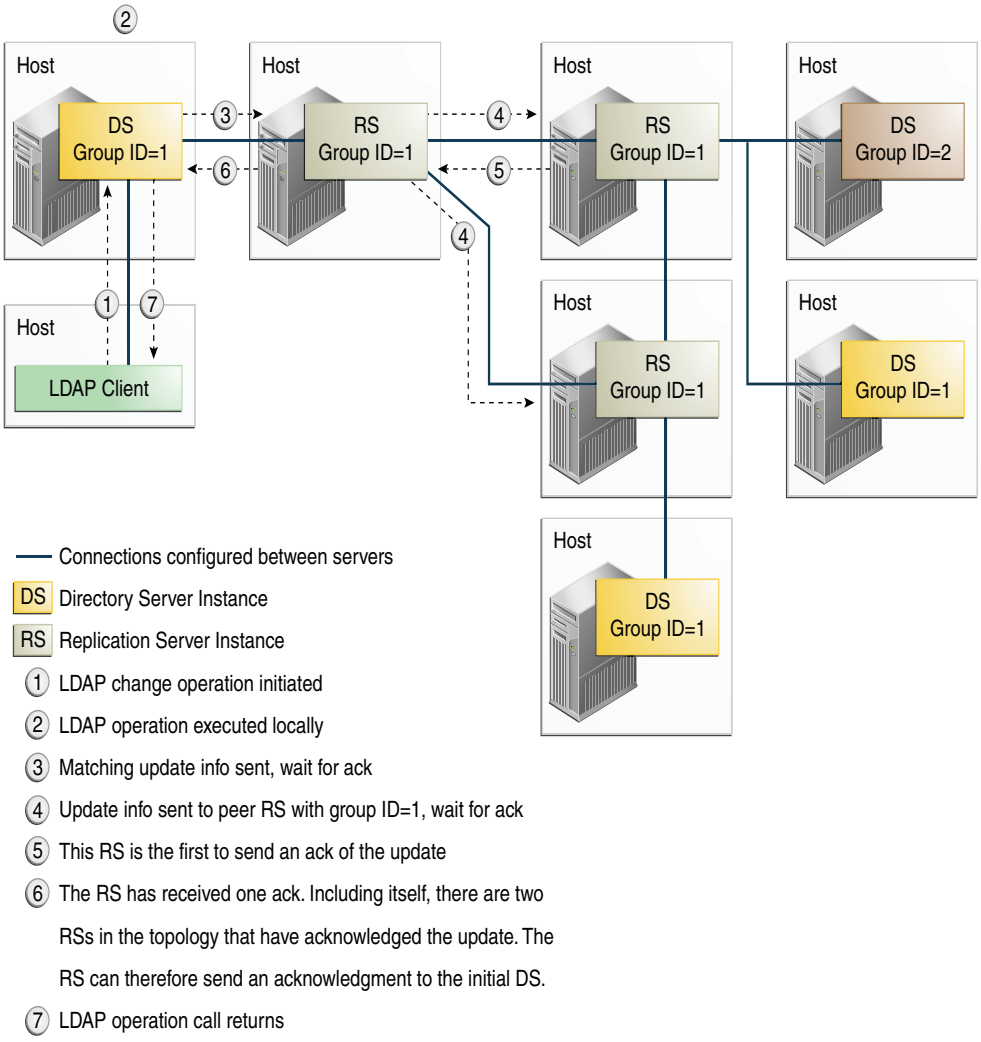
This example can only be configured with `dsconfig` and is not yet supported by the `setup` or `dsreplication` commands.




***Example 6–2 Safe Data Level = 2 (RS and DS on Different Hosts)***

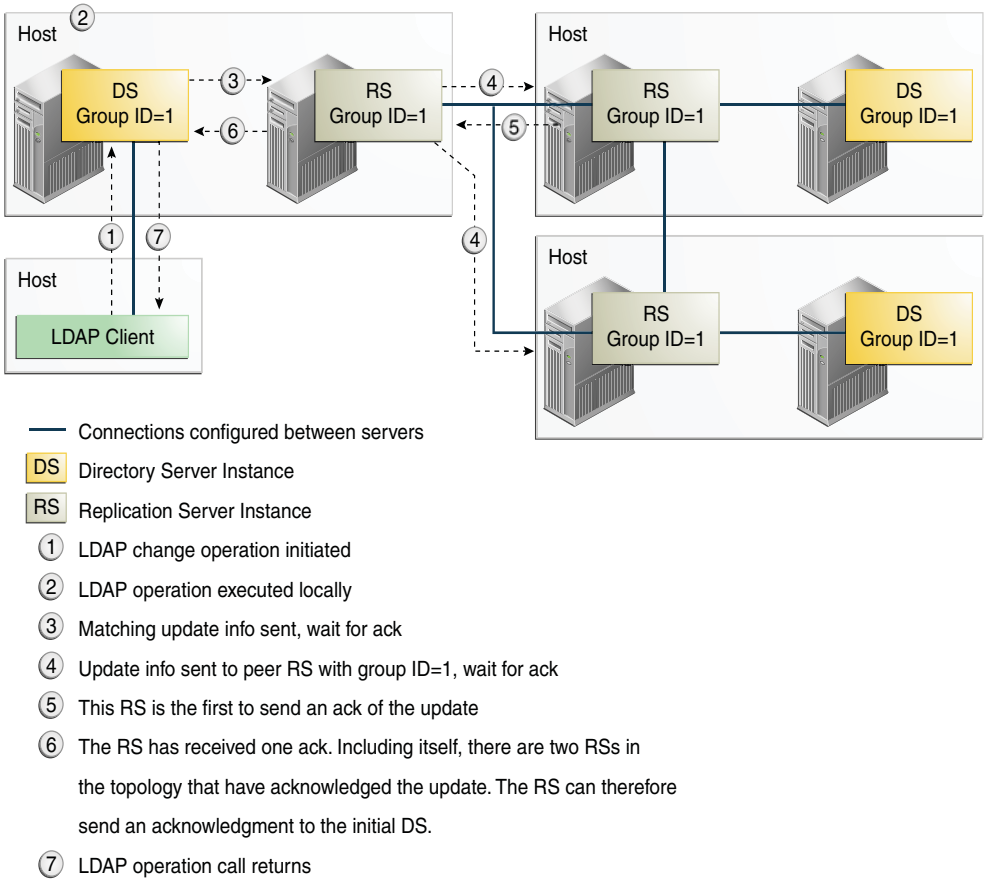
Setting the safe data level to 2 ensures that the first replication server will wait for an acknowledgment from one peer replication server before returning an acknowledgment to the directory server. The modification is guaranteed to exist on two additional servers (other than the directory server on which the change was made).

This example can only be configured with `dsconfig` and is not yet supported by the `setup` or `dsreplication` commands.




***Example 6–3 Safe Data Level = 2 (RS and DS on Same Host)***

In the current replication implementation, the `setup` and `dsreplication` commands only support configurations in which the replication is on the same machine as the directory server. With this implementation, if you want to ensure that a change is sent to at least one additional host, you must set the safe data level to 2.




### 6.7.1.2 Safe Read Mode

Safe read mode ensures that any modification made on a specific directory server has been replayed to all other directory servers within the topology before the LDAP call returns. In this mode, if another LDAP client performs a read operation on another directory server in the topology, that client is assured of reading the modification that has just been performed. Safe read mode is the most synchronized manner in which replication can be configured. However, this mode also has the biggest performance impact in terms of write time.

Safe read mode is based on acknowledgments from the LDAP servers rather than the replication servers in a topology. When a modification is made on a directory server, the update is sent to the corresponding replication server. The replication server then forwards the update to the other replication servers in the topology. These replication servers wait for acknowledgment of the modification being replayed on all the directory servers to which the modification is forwarded. When the modification has been replayed on all directory servers in the topology, the replication servers send their acknowledgment back to the first replication server, which in turn sends an acknowledgment to the original directory server.

The first replication server also waits for an acknowledgment from any other directory servers that are directly connected to it before sending the acknowledgment to the original directory server. Only when the original directory server has received an acknowledgment from its replication server does it finally return the end of the operation call to the LDAP client.

At this point, all directory servers in the topology contain the modification. If an LDAP client reads the data from any directory server, it is therefore certain of obtaining the modification.

#### 6.7.1.3 Safe Read Mode and Replication Groups

*Replication groups* support multi-data center replication and high availability. For more information about replication groups, see Section 6.6, "Replication Groups." In the context of assured replication, replication groups enable a set of directory servers to work together in safe read mode. All directory servers that work together in a synchronized manner require the same group ID. This group ID should also be assigned to all the replication servers working in the synchronized topology. Assured replication does not cross group boundaries.

When a change occurs on a directory server with certain group ID (N), the LDAP call is not returned before every other directory server with group ID N has returned an acknowledgment of the change.

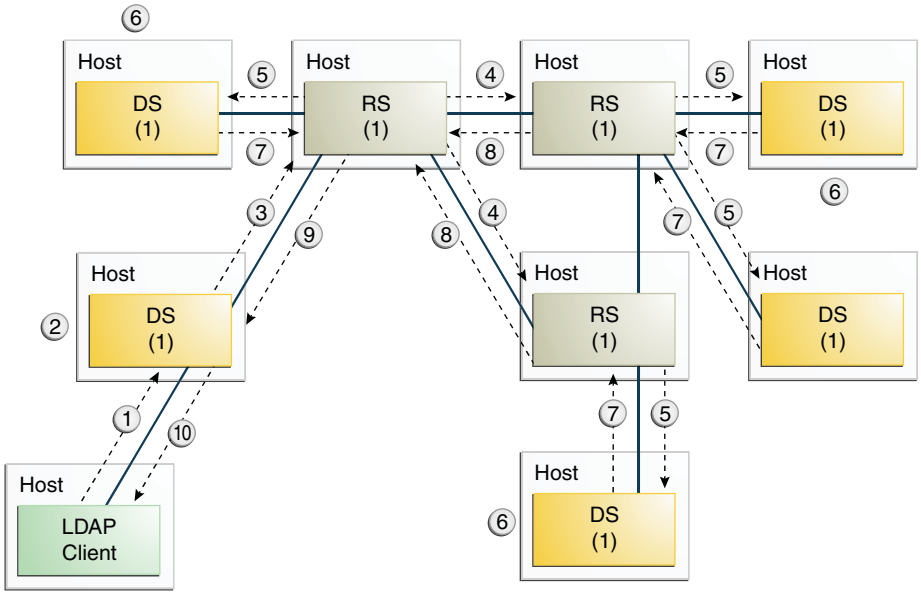
The use of replication groups provides more flexibility in a replicated topology that uses safe read mode.

- In a single data center deployment, you can define a subset of the directory servers that should be fully synchronized. Only the directory servers with the same group ID will wait for an acknowledgment from their peers with the same group ID. All the replication servers will have the same group ID.
- In a multi-data center deployment, you can specify that all the directory servers within a single data center are fully synchronized. A directory servers will wait for acknowledgment only from its peers located in the same data center before returning an LDAP call. Acknowledgment is expected only if the directory server is connected to a replication server with the same group ID.

***Example 6–4 Safe Read Mode in a Single Data Center With One Group***

The following illustration shows a deployment in which all nodes are in the same data center and are part of the same replication group. Each directory server and replication server has the same group ID. Any modification must be replayed on every directory server in the topology before an LDAP client call returns. Any subsequent LDAP read operation on any directory server in the topology is assured of reading the modification.

Such a scenario might be convenient, for example, if there is an LDAP load balancer in front of the replicated directory server pool. Because it is impossible to determine the directory server to which the load balancer will redirect an LDAP modification, a subsequent read operation is not necessarily routed to the directory server on which the modification was made. In this case, it is imperative that the change is made on all servers in the topology before the LDAP client call is returned.

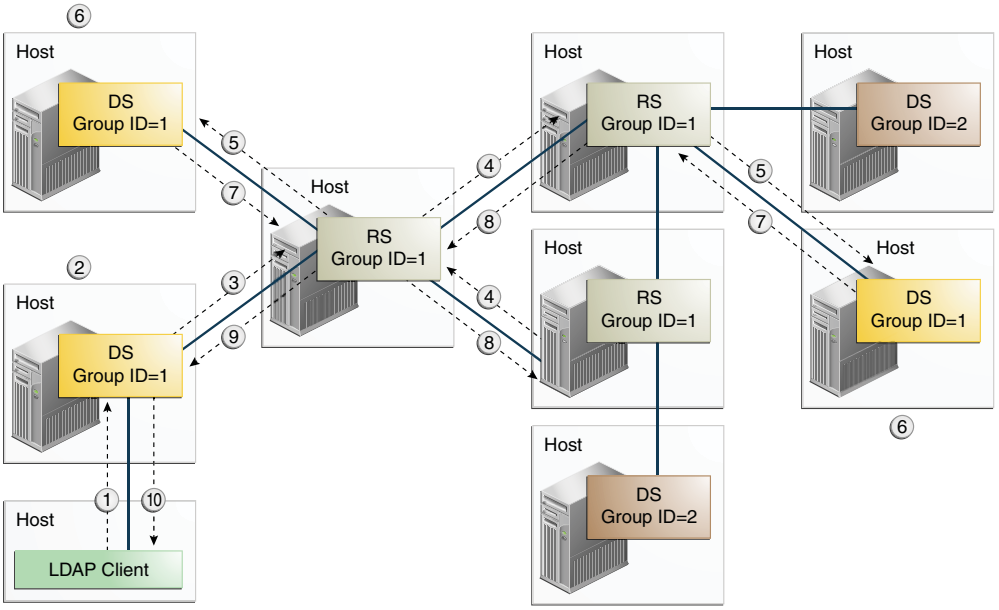



***Example 6–5 Safe Read Mode in a Single Data Center With More Than One Group***

The following illustration shows a deployment in which all nodes are in the same data center but in which assured replication is configured on only a subset of the directory servers. This subset of servers constitutes a replication group, and each server is assigned the same group ID (1). When a change is made on one of the directory servers in the replication group, an acknowledgment must be received from all the directory servers in the group before the initial LDAP call is returned to the client. The remaining directory servers in the topology will still replay the change, but their acknowledgment is not required before the LDAP call is returned. If a change made on one of the servers outside of the group, no acknowledgment from other directory servers is required before the LDAP call is returned to the client.

In this example, the replication server that is connected to directory servers outside of the replication group is still assigned a group ID of 1. This configuration ensures failover in the case of another replication server being offline. In this case, if a directory server within the replication group connects to this particular replication server, assured replication must still work. For the purpose of failover, any replication server

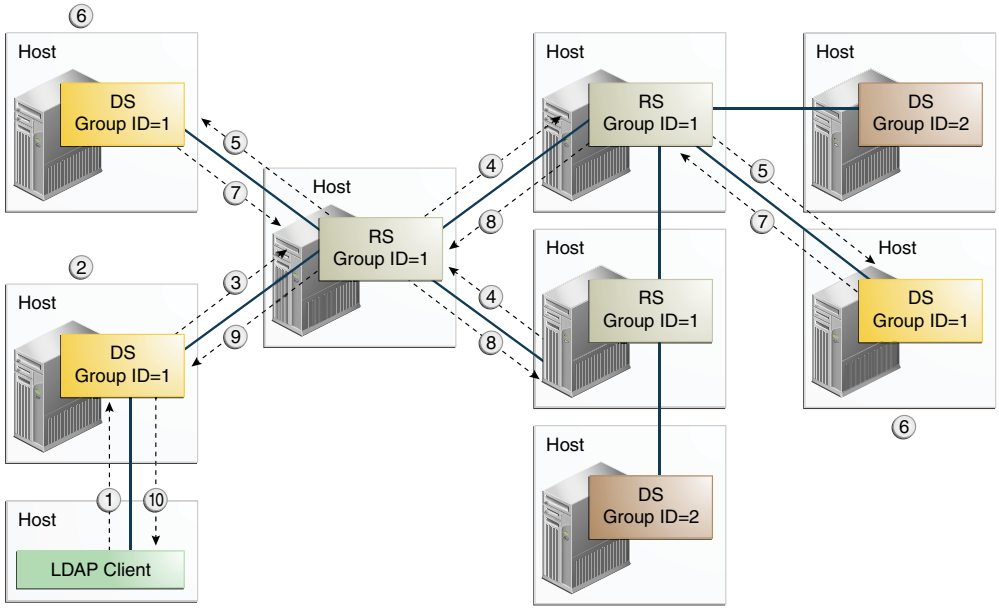
must be assigned the same group ID if there is a chance that a directory server within the group might connect to it at some stage.




***Example 6–6 Safe Read Mode in a Multi-Data Center Deployment***

The following illustration shows a deployment with two data centers (in different geographical locations). Each data center has safe read mode configured locally within the data center. All of the directory servers and the replication servers within the same data center are assigned the same group ID (1 for the first data center and 2 for the second data center). The directory servers within the same data center operate in a more tightly consistent synchronized manner. Any change made on a directory server must be replayed and acknowledged from all directory servers within that data center before the LDAP client call returns.

In this example, data is synchronized between the two data centers, but any change made on a specific directory server is immediately visible on all other directory servers within the same data center. This scenario is convenient if there is an LDAP load balancer in front of the directory servers of a data center. The performance impact in terms of writes is not too great because no acknowledgments are requested from the servers of the remote data center.




Connections configured between servers

1. LDAP change operation initiated
2. LDAP operation executed locally
3. Matching update info sent, wait for ack
4. Update info sent to peer RS with group ID=1, wait for ack
5. Update info sent to peer DS with group ID=1, wait for ack
6. LDAP operation replayed locally
7. DS sends ack back to RS
8. RS sends ack initial RS
9. Initial RS sends ack to initial DS
10. LDAP operation call returns

The group ID of the replication server is important in this scenario. If a change arrives from a directory server with group ID N, the replication server compares N with its own group ID and takes the following action:

- If the replication server has the same group ID (N), it forwards the change to all the replication servers and directory servers to which it is directly connected. However, it waits for an acknowledgment only from the servers with the same group ID (N) before sending its own acknowledgment back to the original directory server.
- If the replication server has a different group ID, it forwards the change to all the replication servers and directory servers but does not wait for any acknowledgment.

## 6.7.2 Assured Replication Connection Algorithm

In implementing the scenarios described in the previous sections, a directory server in a topology uses the following algorithm to select the replication server to which that directory server should connect:

1. Connect to each replication server in the list of configured replication servers and obtain its server state and group ID.

2. From the list of replication servers that are up to date with the changes on the directory server, and that have same group ID as the directory server, select the one that has the most updates from other directory servers in the topology. If no replication server exists with the same group ID as the directory server, select the replication server that is most up to date.

This algorithm ensures that a higher priority is given to replication servers with the same group ID as the directory server's group ID. A directory server will therefore favor a replication server located in its own data center.

Connecting to a replication server with the same group ID (in the same data center) provides the safe read mode functionality. Connecting to a replication server with a different group ID provides failover to another data center (if all the replication servers in the local data center fail). In this case, safe read mode is disabled as no acknowledgment is requested when sending update messages to replication servers with a different group ID. Replication continues, but in degraded mode (that is, the safe read mode requested at configuration time is not applied.)

To return replication to normal, a directory server periodically polls the configuration list for the arrival of replication servers with the same group ID as its own. If the directory server detects that a replication server with its own group ID is available, it disconnects from the current replication server (with a different group ID), and reconnects to the recovered replication server with the same group ID. Safe read mode is thus re-enabled and replication returns to the mode in which it was configured.

### 6.7.3 Assured Replication and Replication Status

When a replication server detects that a directory server is out of sync regarding the overall updates made in the topology, that directory server is said to have a *degraded status*. A directory server that is out of sync is unlikely to be able to send the expected acknowledgments in time for the replication server to avoid a time-out situation. The server therefore has a degraded status until it has an acceptable level of updates. With a degraded status, a directory server is no longer expected to send acknowledgments to the replication server, until it returns to having a *normal status*.

Because a directory server with a degraded status is unable to send acknowledgments, the synchronization of an LDAP operation in safe read mode cannot be assured. In other words, data read from this directory server might not contain the modifications made on another directory server in the topology.

For more information, see Section 6.5.1, "Replication Status Definitions."

### 6.7.4 Assured Replication Monitoring

The assured replication mechanism includes several attributes defined to monitor how well the mechanism is working. The following tables list the monitoring attributes defined on the directory servers and on the replication servers in a topology.

On a directory server, the attributes are located under the monitor entry for that replicated DN. For example, monitoring information related to the replicated domain `dc=example,dc=com` is located under the monitoring entry `cn=Replication Domain,dc=example,dc=com,server-id,cn=monitor`.

On a replication server, the monitoring information related to assured replication is on a per connection basis. Monitoring attributes are found in the monitoring entry of a directory server or replication server that is connected to the current replication server. For example, on a particular replication server, the monitoring information related to a connected directory server would be under the monitoring entry `cn=Directory Server dc=example,dc=com ds-host,server-id,cn=monitor`. The

monitoring information related to a connected replication server would be under the monitoring entry `cn=Remote Replication Server dc=example,dc=com repl-server-host:repl-port,server-id,cn=monitor`.

***Table 6–1 Monitoring Attributes on the Directory Server***

| Attribute Name | Attribute Type | Purpose |
|---|---|---|
| `assured-sr-sent-updates` | Integer (0..N) | Number of updates sent in assured replication, safe read mode |
| `assured-sr-acknowledged-updates` | Integer (0..N) | Number of updates sent in assured replication, safe read mode, that have been successfully acknowledged |
| `assured-sr-not-acknowledged-updates` | Integer (0..N) | Number of updates sent in assured replication, safe read mode, that have not been successfully acknowledged (either because of timeout, wrong status, or error at replay) |
| `assured-sr-timeout-updates` | Integer (0..N) | Number of updates sent in assured replication, safe read mode, that have not been successfully acknowledged because of timeout |
| `assured-sr-wrong-status-updates` | Integer (0..N) | Number of updates sent in assured replication, safe read mode, that have not been successfully acknowledged because of wrong status |
| `assured-sr-replay-error-updates` | Integer (0..N) | Number of updates sent in assured replication, safe read mode, that have not been successfully acknowledged because of replay error |
| `assured-sr-server-not-acknowledged-updates` | String | Multiple values allowed: number of updates sent in assured replication, safe read mode, that have not been successfully acknowledged (either because of timeout, wrong status or error at replay) for a particular server (directory server or replication server). String format: *server id:number of failed updates* |
| `assured-sr-received-updates` | Integer (0..N) | Number of updates received in assured replication, safe read mode |
| `assured-sr-received-updates-acked` | Integer (0..N) | Number of updates received in assured replication, safe read mode that have been acknowledged without errors |
| `assured-sr-received-updates-not-acked` | Integer (0..N) | Number of updates received in assured replication, safe read mode, that have been acknowledged with errors |
| `assured-sd-sent-updates` | Integer (0..N) | Number of updates sent in assured replication, safe data mode |
| `assured-sd-acknowledged-updates` | Integer (0..N) | Number of updates sent in assured replication, safe data mode, that have been successfully acknowledged |

*Table 6–1 (Cont.) Monitoring Attributes on the Directory Server*

| Attribute Name | Attribute Type | Purpose |
|---|---|---|
| assured-sd-timeout-updates | Integer (0..N) | Number of updates sent in assured replication, safe data mode, that have not been successfully acknowledged because of timeout |
| assured-sd-server-timeout-updates | String | Multiple values allowed: number of updates sent in assured replication, safe data mode, that have not been successfully acknowledged (because of timeout) for a particular RS. String format: *server id*:*number of failed updates* |

*Table 6–2 Monitoring Attributes on the Replication Server*

| Attribute Name | Attribute Type | Purpose |
|---|---|---|
| assured-sr-received-updates | Integer (0..N) | Number of updates received from the remote server in assured replication, safe read mode |
| assured-sr-received-updates-timeout | Integer (0..N) | Number of updates received from the remote server in assure replication, safe read mode, that timed out when forwarding them |
| assured-sr-sent-updates | Integer (0..N) | Number of updates sent to the remote server in assured replication, safe read mode |
| assured-sr-sent-updates-timeout | Integer (0..N) | Number of updates sent to the remote server in assured replication, safe read mode, that timed out |
| assured-sd-received-updates | Integer (0..N) | Number of updates received from the remote server in Assured Replication, Safe Data |
| assured-sd-received-updates-timeout | Integer (0..N) | Number of updates received from the remote server in assured replication, safe date mode, that timed out when forwarding them. This attribute is meaningless if the remote server is a replication server. |
| assured-sd-sent-updates | Integer (0..N) | Number of updates sent to the remote server in assured replication, safe data mode. This attribute is meaningless if the remote server is a directory server. |
| assured-sd-sent-updates-timeout | Integer (0..N) | Number of updates sent to the remote server in assured replication, safe data mode, that timed out. This attribute is meaningless if the remote server is a directory server. |

## 6.8 Fractional Replication

The fractional replication feature enables you to restrict certain attributes from being included when modify operations are replayed on specific servers in a topology. For

information about configuring fractional replication, see Section 26.3.10, "Configuring Fractional Replication."

This section describes the architecture of the fractional replication mechanism and covers the following topics:

- Section 6.8.1, "Fractional Data Set Identification"
- Section 6.8.2, "Fractional Replication Filtering"
- Section 6.8.3, "Fractional Replication and Local Operations"

### 6.8.1 Fractional Data Set Identification

A fractional data set is identified by the following operational attributes that are stored in the root entry of the replicated domain:

- `ds-sync-fractional-exclude`
- `ds-sync-fractional-include`

The syntax and meaning of these attributes is identical to their corresponding configuration attributes, described in Section 26.3.10, "Configuring Fractional Replication." The role of these operational attributes is to *tag* a data set as fractional: their presence in a domain implies "this data set is a fractional domain and does not contain the following specific attributes...".

The fractional configuration stored in the root entry of the domain, combined with the generation ID (`ds-sync-generation-id`) and the replication state (`ds-sync-state`), can be seen as the *fractional signature* of the data set.

When a domain is enabled (for example, after its fractional configuration is modified), the server compares the fractional configuration of the domain (under `cn=config`) with the fractional configuration attributes in the root entry of the domain. If both configurations match, the domain assumes a *normal* status and LDAP operations can be accepted. If the configurations do not match, the domain assumes a *bad generation ID* status and the data set must be synchronized (by importing a data set) before LDAP operations can be accepted.

The data set that is imported must either:

- have the same fractional configuration in its root entry as the local domain has under `cn=config`. In this case, the data set is imported as is.
- have no fractional configuration in its root entry. In this case, the data set is imported and filtered according to the attribute filtering rules defined in the fractional configuration of the local domain (under `cn=config`). The `ds-sync-fractional-exclude` or `ds-sync-fractional-include` attributes are then created in the root entry of the imported data, by copying the fractional configuration of the local domain.

### 6.8.2 Fractional Replication Filtering

When a domain is configured as fractional, all `ADD`, `MODIFY`, and `MODIFYDN` operations that arrive from the network to be replayed are filtered. These operations can end up being abandoned if all of the attributes in the operation are filtered attributes according to the fractional configuration.

### 6.8.3 Fractional Replication and Local Operations

If an LDAP client performs an operation directly on a fractional replica and the operation does not match the fractional configuration, the operation is forbidden and the server returns an "unwilling to perform" error.

For example, if a fractional replica is configured with `fractional-exclude: *:jpegPhoto` and an LDAP client attempts to add a new entry that contains a `jpegPhoto` attribute, the operation is rejected with an "unwilling to perform" error. This behavior ensures that the domain remains consistent with its fractional configuration definition, which implies that no `jpegPhoto` attribute can exist on the domain.

# 7

# Understanding the Oracle Unified Directory Indexing Model

Like a book index, Oracle Unified Directory indexes speed up searches by associating search strings with the contents of a directory.

This chapter describes the various index types and the way in which searches are evaluated and includes the following sections:

- Section 7.1, "Overview of Indexes"
- Section 7.2, "Index Types"
- Section 7.3, "Index Entry Limit"
- Section 7.4, "Search Evaluation"
- Section 7.5, "Maintaining Indexes"

For information about configuring indexes, see Section 17.7, "Indexing Directory Data".

## 7.1 Overview of Indexes

Oracle Unified Directory uses indexes to speed up search operations by associating lookup information with Oracle Unified Directory entries. Each search operation includes a search key that specifies the entries to return. During a search operation the server uses the index to find entries that match the search key. If indexes are not configured, then the server must check every entry in a suffix to locate potential matches for the search key.

Navigating through all entries in the directory is resource-intensive, especially for large directories. In addition, unindexed searches might not be allowed to non-privileged users. For more information about assigning privilege for unindexed search, see Chapter 10, "Understanding Root Users and the Privilege Subsystem." To make searches more efficient, you can configure indexes to correspond to the searches that clients need to perform.

This section contains the following topics:

- Section 7.1.1, "What is an Index?"
- Section 7.1.2, "Understanding the Importance of Indexing"

### 7.1.1 What is an Index?

An index is a mechanism used by the Directory Server database to efficiently find entries matching the search criteria. An index maps a search key to an ID list, which is a set of entry IDs for the entries that match that index key.

### 7.1.2 Understanding the Importance of Indexing

- The most efficient methodology to improve search operations against the directory server is to configure indexes, combined with defining an index entry limit on search results.
- An index stores the values of specified attributes for an entry without storing any other detail about the entry. This saves space and makes search faster by organizing the index around that attribute. If you perform a search on an attribute that has been indexed, Oracle Unified Directory quickly locates the index for the entries that meet the search criteria.

## 7.2 Index Types

Oracle Unified Directory supports the following index types:

- Section 7.2.1, "Approximate Indexes"
- Section 7.2.2, "Equality Indexes"
- Section 7.2.3, "Ordering Indexes"
- Section 7.2.4, "Presence Indexes"
- Section 7.2.5, "Substring Indexes"

### 7.2.1 Approximate Indexes

An approximate index is used to match values that sound like the values that are provided in the search filter. The purpose of an approximate index is to locate entries that match values similar to the search filter. For example, an approximate index on the `cn` attribute allows client applications to locate entries even when the names are misspelled.

### 7.2.2 Equality Indexes

An equality index identifies which entries are exactly equal to the value that is provided in a search filter. An equality index can only be maintained for attributes that have a corresponding equality matching rule.

### 7.2.3 Ordering Indexes

An ordering index keeps track of the relative order of values for an attribute. It is similar to an equality index except that it uses an ordering matching rule instead of an equality matching rule to normalize the values. Ordering indexes can not be maintained for attributes that do not have a corresponding ordering matching rule.

### 7.2.4 Presence Indexes

A presence index keeps track of the entries that have at least one value for a specified attribute. There is only a single presence index key per attribute, and its ID list contains the entry ID for all entries that contain the specified attribute. The `aci`

attribute is indexed for presence by default to enable quick retrieval of entries with ACIs.

### 7.2.5 Substring Indexes

A substring index keeps track of which entries contain specific substrings. Index keys for a substring index consist of six-character substrings taken from attribute values and the corresponding values are an ID list containing the entry ID of the entries containing those substrings. The attribute's substring matching rule is used to normalized value the values for the index keys, and substring indexes cannot be defined for attributes that do not contain substring matching rules.

## 7.3 Index Entry Limit

The index entry limit is a configuration limit that can be used to control the maximum number of entries that is allowed to match any given index key (that is, the maximum size of an ID list). This provides a mechanism for limiting the performance impact for maintaining index keys that match a large percentage of the entries in the server. In cases where large ID lists might be required, performing an unindexed search can often be faster than one that is indexed.

## 7.4 Search Evaluation

To process an LDAP search operation, the server applies each assertion of the search filter to generate a list of candidate entries, which are then combined to form an initial set of candidate entry IDs.

If a candidate set is obtained, the search is considered to be **indexed.** Each candidate entry is fetched from the entry database and returned to the client if it matches the search scope and filter.

If no candidate set is obtained (because of a lack of indexes or some of the index values having exceeded the index entry limit), the search is considered to be **not-indexed.** In this case, a cursor is opened on the DN database at the base entry to iterate through all records in scope, fetching and filtering the corresponding entries until all the entries under the search base have been processed.

Whenever the number of candidate entry IDs from the indexes is found to be 10 or less, no further attempt is made to reduce the number of candidates. Instead those entries are immediately fetched from the entry database and filtered, on the assumption that this is quicker than continuing to read the index databases. This can pay off for AND search filters in which the first component is the most specific.

Search AND filters are also rearranged so that components that are slow to evaluate (greater-than-or-equal, less-than-or-equal) come after components that are generally faster (for example, equality).

## 7.5 Maintaining Indexes

You can consider the following key points for maintaining indexes:

- Run the `verify-index` command to check the consistency between the index and the entry data within the directory server database.

  For more information about the command, see Section A.3.16, "verify-index."

- Run the `rebuild-index` command to rebuild the directory server indexes, if you create a new index or when the `index-entry-limit` property of an index changes.

  For more information about the command, see Section A.3.13, "rebuild-index."

- Configure a Virtual List View (VLV) index, which is a mechanism used by the Directory Server database to efficiently process searches with VLV controls. A VLV index effectively notifies the server that a virtual list view, with specific query and sort parameters, will be performed. This index also allows the server to collect and maintain the information required to make using the virtual list view faster. A VLV index stores sorted blocks of ID lists, which are a set of entry IDs and the attribute values of the entry to sort on.

  For more information about configuring VLV indexes, see Section 17.7.2, "Configuring VLV Indexes."

- Configure an extensible match index to accelerate search operations using an extensible match search filter. Index keys are values that have been normalized using a specified matching rule, and the corresponding ID list contains the entry ID for all entries that match the value according to that matching rule.

  For more information about extensible match search filter, see Section D.5.13, "extensible match search filter."

# 8

# Understanding the Oracle Unified Directory Access Control Model

This chapter contains reference information about the directory server access control model. For information about configuring access control in the directory server, see Chapter 22, "Controlling Access To Data."

This chapter covers the following topics:

- Section 8.1, "Access Control Principles"
- Section 8.2, "ACI Syntax"
- Section 8.3, "Bind Rules"
- Section 8.4, "Bind Rule Syntax"
- Section 8.5, "Compatibility With the Oracle Directory Server Enterprise Edition Access Control Model"
- Section 8.6, "Using Macro ACIs for Advanced Access Control"

## 8.1 Access Control Principles

This section describes the principles of the access control mechanism provided with the directory server.

- Section 8.1.1, "Access Control Overview"
- Section 8.1.2, "ACI Structure"
- Section 8.1.3, "Directory Server Global ACIs"
- Section 8.1.4, "ACI Evaluation"
- Section 8.1.5, "ACI Limitations"
- Section 8.1.6, "Access Control and Replication"

See also Section 22.1, "Managing Global ACIs With `dsconfig`".

### 8.1.1 Access Control Overview

When the directory server receives a request, it uses the authentication information provided by the user in the bind operation, and the access control instructions (ACIs) defined in the server to allow or deny access to directory information. The server can allow or deny permissions such as read, write, search, or compare. The permission level granted to a user might depend on the authentication information that the user provides.

Using access control, you can control access to the entire directory, a subtree of the directory, specific entries in the directory (including entries that define configuration tasks), a specific set of entry attributes, or specific entry attribute values. You can set permissions for a particular user, for all users who belong to a specific group or role, or for all users of the directory. Finally, you can define access for a specific client, identified by its IP address or DNS name.

## 8.1.2 ACI Structure

Access control instructions are stored in the directory as attributes of entries. The `aci` attribute is an operational attribute that is available for use on every entry in the directory, regardless of whether it is defined for the object class of the entry. This attribute is used by the directory server to evaluate what rights are granted or denied when the directory server receives an LDAP request from a client. The `aci` attribute is returned in an `ldapsearch` operation only if it is specifically requested.

An ACI statement includes three main parts:

**Target**
Determines the entry or attributes to which permissions apply.

**Permission**
Defines what operations are allowed or denied.

**Bind Rule**
Determines who is subject to the ACI, based on their bind DN.

The permission and bind rule portions of the ACI are set as a pair, also called an Access Control Rule (ACR). The specified permission to access the target is granted or denied depending on whether the accompanying rule is evaluated to be true. For more information, see Section 8.2, "ACI Syntax."

If an entry that contains an ACI does not have child entries, the ACI applies to that entry only. If the entry has child entries, the ACI applies to the entry itself and to all entries below it. Therefore, when the directory server evaluates access permissions to an entry, it verifies the ACIs for every entry between the one that was requested and the base of its root suffix.

The `aci` attribute is multivalued, which means that you can define several ACIs for the same entry or subtree.

You can create an ACI on an entry that does not apply directly to that entry but to some or all of the entries in the subtree below it. The advantage of this is that you can place at a high level in the directory tree a general ACI that effectively applies to entries that are more likely to be located lower in the tree. For example, at the level of an `organizationalUnit` entry or a `locality` entry, you could create an ACI that targets entries that include the `inetorgperson` object class.

You can use this feature to minimize the number of ACIs in the directory tree by placing general rules at high-level branch points. To limit the scope of more specific rules, place them as close as possible to leaf entries.

---

**Note:** ACIs that are placed in the root DSE entry (with the DN `""`) apply only to that entry.

---

### 8.1.3 Directory Server Global ACIs

You can configure access control centrally by using `dsconfig` to modify the properties of the Access Control Handler.

The following default global ACIs apply to all suffixes that are defined in the directory server because the rules do not specify a `target` expression:

```
Property   : Value(s)
-----------:---------------------------------------------------------------------
global-aci : "(targetattr!="userPassword||authPassword")(version 3.0; acl
: "Anonymous read access"; allow (read,search,compare)
: userdn="ldap:///anyone";)", (targetattr="*")(version 3.0; acl
: "Self entry modification"; allow (write) userdn="ldap:///self";),
: "(targetattr="createTimestamp||creatorsName||modifiersName||modify
: Timestamp||entryDN||entryUUID||subschemaSubentry")(version 3.0;
: acl "User-Visible Operational Attributes"; allow
: (read,search,compare) userdn="ldap:///anyone";)",
```

For more information, see Section 22.1, "Managing Global ACIs With `dsconfig`."

### 8.1.4 ACI Evaluation

To evaluate the access rights to a particular entry, the server compiles a list of the ACIs present on the entry itself and on the parent entries back up to the base of the entry's root suffix. During evaluation, the server processes the ACIs in this order. ACIs are evaluated in all of the suffixes and subsuffixes between an entry and the base of its root suffix, but not across chained suffixes on other servers.

> **Note:** Access control does not apply to any user who has the `bypass-acl` privilege. The Directory Manager has this privilege. When a client is bound to the directory as the Directory Manager, the directory server does not evaluate any ACIs before performing operations. As a result, performance of LDAP operations as Directory Manager is not comparable to the expected performance of other users. You should always test directory performance with a typical user identity.

By default, if no ACI applies to an entry, access is denied to all users except those with the `bypass-acl` privilege. Access must be explicitly granted by an ACI for a user to access any entry in the directory. The default ACIs define anonymous read access and allow users to modify their own entries, except for attributes needed for security. For more information, see Section 22.1.1, "Default Global ACIs."

Although the directory server processes the ACIs that are closest to the target entry first, the effect of all ACIs that apply to an entry is cumulative. Access granted by any ACI is allowed unless any other ACI denies it. ACIs that deny access, no matter where they appear in the list, take precedence over ACIs that allow access to the same resource.

For example, if you deny write permission at the directory's root level, none of the users can write to the directory regardless of the specific permissions you grant them. To grant a specific user write permissions to the directory, you must restrict the scope of the original denial for write permission so that it does not include that user.

### 8.1.5 ACI Limitations

Be aware of the following limitations when you create an access control policy for your directory service:

- If your directory tree is distributed over several directory servers, some restrictions apply to the keywords that you can use in access control statements. ACIs that depend on group entries (`groupdn` keyword) must be located on the same directory server as the group entry. If the group is dynamic, all members of that group must also have an entry on the directory server. If the group is static, the members' entries can be located on remote directory servers. However, you can do value matching of values stored in the target entry with values stored in the entry of the bind user (for example, using the `userattr` keyword). Access is evaluated normally even if the bind user does not have an entry on the directory server that holds the ACI.
- Access control rules are always evaluated on the local directory server. You must not specify the host name or port number of the directory server in LDAP URLs used in ACI keywords. If you do, the LDAP URL is not taken into account at all.

### 8.1.6 Access Control and Replication

ACIs are stored as attributes of entries, so if an entry containing ACIs is part of a replicated suffix, the ACIs are replicated like any other attribute.

## 8.2 ACI Syntax

ACIs are complex structures with many possible variations. The following sections describe the syntax of an ACI in detail.

- Section 8.2.1, "ACI Syntax Overview"
- Section 8.2.2, "Defining Targets"
- Section 8.2.3, "Defining Permissions"

See also Section 8.4, "Bind Rule Syntax."

### 8.2.1 ACI Syntax Overview

The `aci` attribute has the following syntax:

```
aci: (target)(version 3.0;acl "name";permissionBindRules;)
```

where:

- *target* specifies the entry, attributes, or set of entries and attributes for which you want to control access. The target can be a distinguished name, one or more attributes, or a single LDAP filter. The target is optional. When the target is not specified, the ACI applies to the entire entry where it is defined and all of its children.
- `version 3.0` is a required string that identifies the ACI version.
- *name* is a name for the ACI. The name can be any string that identifies the ACI. The ACI name is required and should describe the effect of the ACI. Although there are no restrictions on the name, it is good practice to use unique names for ACIs. If you use unique names, the Get Effective Rights control enables you to determine which ACI is in force.

- *permission* specifically states what rights you are either allowing or denying, for example read or search rights.
- *bindRules* specify the credentials and bind parameters that a user has to provide to be granted access. Bind rules can also be based on user or group membership or connection properties of the client.

You can specify multiple targets and permission-bind rule pairs. This allows you to refine both the entry and attributes being targeted and efficiently set multiple access controls for a given target, as shown here:

```
aci: (target)...(target)(version 3.0;acl "name"; permissionBindRule;
permissionBindRule; ...; permissionBindRule;)
```

The following example shows a complete LDIF ACI:

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

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

## 8.2.2 Defining Targets

The target identifies what the ACI applies to. When a client requests an operation on attributes in an entry, the directory server evaluates the target to see if the ACI must be evaluated to allow or deny the operation. If the target is not specified, the ACI applies to all attributes in the entry containing the `aci` attribute and to the entries below it.

The following sections describe how to define targets:

- Section 8.2.2.1, "Targeting a Directory Entry"
- Section 8.2.2.2, "To Target Attributes"
- Section 8.2.2.3, "To Target an Entry and Attributes"
- Section 8.2.2.4, "To Target Entries or Attributes Using LDAP Filters"
- Section 8.2.2.5, "To Target Attribute Values Using LDAP Filters"
- Section 8.2.2.6, "To Target a Single Directory Entry"
- Section 8.2.2.7, "To Specify the Scope of an ACI"
- Section 8.2.2.8, "To Target LDAP Controls"
- Section 8.2.2.9, "To Target LDAP Extended Operations"

The general syntax for a target is one of the following:

```
(keyword = "expression")
(keyword != "expression")
```

where:

- *keyword* indicates the type of target. The following types of targets are defined by the keywords in Table 8–1:
  - A directory entry or its subtree
  - The attributes of an entry
  - A set of entries or attributes that match an LDAP filter

- An attribute value or combination of values that match an LDAP filter
- The scope of the ACI
- An LDAP control
- An extended operation

- The equal sign (=) indicates that the target is the object specified in the expression, and not equal (!=) indicates that the target is any object not specified in the expression.

> **Note:** The not-equal operator is not supported for the `targattrfilters` and `targetscope` keywords.

- *expression* is dependent on the keyword and identifies the target. The quotation marks (`""`) around *expression* are syntactically required, although the current implementation accepts expressions like `targetattr=*`. In future versions, syntax checking might become more strict, so you should always use quotation marks.

The following table lists each keyword and the associated expressions.

***Table 8–1 LDIF Target Keywords***

| Keyword | Valid Expressions | Wildcard Allowed? |
|---|---|---|
| `target` | `ldap:///`*distinguishedName* | Allowed |
| `targetattr` | *attribute* | Allowed |
| `targetfilter` | *LDAPfilter* | Allowed |
| `targattrfilters` | *LDAPoperation*`:`*LDAPfilter* | Allowed |
| `targetscope` | `base, onelevel, subtree, subordinate` | Not Allowed |
| `targetcontrol` | *oid* | Not Allowed |
| `extop` | *oid* | Not Allowed |

### 8.2.2.1 Targeting a Directory Entry

Use the target keyword and a DN inside an LDAP URL to target a specific directory entry and any entries below it. The targeted DN must be located in the entry where the ACI is defined or in the subtree below the entry. The target expression has the following syntax:

```
(target = "ldap:///distinguishedName")
(target != "ldap:///distinguishedName")
```

The distinguished name must be located in the entry where the ACI is defined or in the subtree below the entry. For example, the following target can be used in an ACI on `ou=People,dc=example,dc=com`:

```
(target = "ldap:///uid=bjensen,ou=People,dc=example,dc=com")
```

The keyword `target` is optional. If it is not present, the default target for the ACI is the entry where the ACI is stored.

**Note:** The DN of the entry must be a distinguished name in string representation (defined in RFC 4514 (http://www.ietf.org/rfc/rfc4514.txt)). Therefore, characters syntactically significant for a DN, such as commas, must be escaped with a single backslash (\). For example:

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

You can also use a wildcard in the DN to target any number of entries that match the LDAP URL. The following are legal examples of wildcard usage:

- `(target="ldap:///uid=*,dc=example,dc=com")` Matches every immediate child of the `example.com` branch entry that has the `uid` attribute in the entry's RDN, as shown in this example.

  ```
  uid=tmorris,dc=example,dc=com
  uid=yyorgens,dc=example,dc=com
  uid=bjensen,dc=example,dc=com
  ```

- `(target="ldap:///uid=*,**,dc=example,dc=com")` Matches every entry more than one level below the `example.com` branch entry that has the `uid` attribute in the entry's RDN, as shown in this example.

  ```
  uid=tmorris,ou=sales,dc=example,dc=com
  uid=yyorgens,ou=marketing,dc=example,dc=com
  uid=bjensen,ou=eng,ou=east,dc=example,dc=com
  ```

- `(target="ldap:///uid=*Anderson,ou=People,dc=example,dc=com")` Matches every entry immediately below the `ou=People` branch entry with a `uid` ending in Anderson.
- `(target="ldap:///*=*Anderson,ou=People,dc=example,dc=com")` Matches every entry immediately below the `ou=People` branch whose RDN ends with Anderson, regardless of the naming attribute.

Multiple wildcards are allowed, such as in `uid=*,ou=*,dc=example,dc=com`, which matches every entry in the `example.com` tree whose distinguished name contains the `uid` and `ou` attributes in the specified positions.

#### 8.2.2.2 To Target Attributes

In addition to targeting directory entries, you can also target one or more attributes (or all but one or more attributes) that occur in the targeted entries. This functionality is useful when you want to deny or allow access to partial information about an entry. For example, you can allow access to only the common name, surname, and telephone number attributes of a given entry. Similarly, you can deny access to sensitive information such as personal data.

If no `targetattr` rule is present, no attributes can be accessed by default. To access all attributes, the rule must be `targetattr="*"`.

The targeted attributes do not need to exist on the target entry or its subtree, but the ACI applies whenever they do. The attributes you target do not need to be defined in the schema. The absence of schema checking makes it possible to implement an access control policy before importing your data and its schema.

To target attributes, use the `targetattr` keyword and provide the attribute names. The `targetattr` keyword uses the following syntax:

```
(targetattr = "attribute")
(targetattr != "attribute")
```

You can target multiple attributes by using the `targetattr` keyword with the following syntax:

```
(targetattr = "attribute1 || attribute2 ... || attributeN")
(targetattr != "attribute1 || attribute2 ... || attributeN")
```

For example, to target an entry's common name, surname, and UID attributes, you would use the following:

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

To target all of an entry's user attributes, except `carlicense`, you would use the following target:

```
(targetattr != "carlicense")
```

The preceding example does not return operational attributes.

Targeted attributes include all subtypes of the named attribute. For example, `(targetattr = "locality")` also targets `locality;lang-fr`. You can also target subtypes specifically, for example, `(targetattr = "locality;lang-fr-ca")`.

You can use a wildcard as a stand-alone character in a `targetattr` rule (such as `targetattr="*"`), but this use is discouraged because it serves no particular purpose and can have a negative performance impact.

#### 8.2.2.3 To Target an Entry and Attributes

By default, the entry targeted by an ACI containing a `targetattr` keyword is the entry on which the ACI is placed. That is, if you apply the ACI `aci: (`*targetattr* `= "`*uid*`")(`*accessControlRules*`;)` to the `ou=Marketing, dc=example,dc=com` entry, then the ACI applies to the entire Marketing subtree. However, you can also explicitly specify a target using the target keyword, as shown in the following example:

```
aci: (target="ldap:///uid=*,ou=Marketing,dc=example,dc=com")
(targetattr="uid") (accessControlRules;)
```

The order in which you specify the target and the `targetattr` keywords is irrelevant.

#### 8.2.2.4 To Target Entries or Attributes Using LDAP Filters

You can use LDAP filters to target a set of entries that match certain criteria. To do this, use the `targetfilter` keyword with an LDAP filter. The ACI applies to all entries that match the filter at the level of the target DN and in the subtree below it.

The `targetfilter` keyword uses this syntax:

```
(targetfilter = "LDAPfilter")
```

where *LDAPfilter* is a standard LDAP search filter. For more information about filter syntax, see Appendix D.18.9, "search filter."

For example, suppose that all entries representing employees have a status of salaried or contractor and an attribute representing the number of hours worked, as a percentage of a full-time position. To target all the entries representing contractors or part-time employees, you could use the following filter:

```
(targetfilter = "(|(status=contractor)(fulltime<=79))")
```

The Netscape extended filter syntax is not supported in ACIs. For example, the following target filter is not valid:

```
(targetfilter = "(locality:fr:=<= Qu?bec)")
```

Target filters select whole entries as targets of the ACI. You can associate the `targetfilter` and the `targetattr` keywords to create ACIs that apply to a subset of attributes in the targeted entries.

The following LDIF example allows members of the Engineering Admin group to modify the `departmentNumber` and `manager` attributes of all entries in the Engineering business category. This example uses LDAP filtering to select all entries with `businessCategory` attributes set to `Engineering`:

```
dn: dc=example,dc=com
objectClass: top
objectClass: organization
aci: (targetattr="departmentNumber || manager")
(targetfilter="(businessCategory=Engineering)")
(version 3.0; acl "eng-admins-write"; allow (write)
groupdn ="ldap:///cn=Engineering Admins, dc=example,dc=com";)
```

Although using LDAP filters can be useful when you are targeting entries and attributes that are spread across the directory, the results are sometimes unpredictable because filters do not directly name the object for which you are managing access. The set of entries targeted by a filtered ACI is likely to change as attributes are added or deleted. Therefore, if you use LDAP filters in ACIs, you should verify that they target the correct entries and attributes by using the same filter in an `ldapsearch` operation.

### 8.2.2.5 To Target Attribute Values Using LDAP Filters

You can use access control to target specific attribute values. This means that you can grant or deny permissions on an attribute if that attribute's value meets the criteria defined in the ACI. An ACI that grants or denies access based on an attribute's value is called a value-based ACI.

For example, you can grant all users in your organization permission to modify the `roomNumber` attribute in their own entries. However, you would also want to ensure that they do not give themselves reserved room numbers, all of which begin with `12`. LDAP filters are used to check that the conditions on attribute values are satisfied.

To create a value-based ACI, you must use the `targattrfilters` keyword with the following syntax:

`(targattrfilters="`*Op*`=`*attr1*`:`*F1*`[(&&` *attr2*`:`*F2*`)*][;`*Op*`=`*attr*`:`*F*`[(&&` *attr*`:`*F*`)*]")`

where:

- *Op* is either an `add` or `delete` operation:
  - `add` represents the operation of creating an attribute.
  - `delete` represents the operation of deleting an attribute.
- *attr* represents the target attributes.
- *F* represents Appendix D.18.9, "search filter" that apply only to the associated attribute.

When creating an entry, if a filter applies to an attribute in the new entry, then all values of that attribute must satisfy the filter. When deleting an entry, if a filter applies to an attribute in the entry, then all values of that attribute must also satisfy the filter.

When modifying an entry, if the operation adds an attribute, then the add filter that applies to that attribute must be satisfied. If the operation deletes an attribute, then the delete filter that applies to that attribute must be satisfied. If individual values of an attribute already present in the entry are replaced, then both the add and delete filters must be satisfied.

The following example attribute filter allows users to add any `roomNumber` attribute to their own entries except the reserved room numbers, which have a `12` prefix. It also allows users to add a telephone number with a `123` prefix.

```
(targattrfilters="add=roomNumber:(!(roomNumber=12*)) &&
telephoneNumber:(telephoneNumber=123*)")
```

#### 8.2.2.6 To Target a Single Directory Entry

There is no explicit way to target a single entry. However, you can achieve this in one of two ways:

- By creating a bind rule that matches user input in the bind request with an attribute value stored in the targeted entry
- By using the `targetfilter` keyword

With the `targetfilter` keyword you can specify an attribute value that appears only in the desired entry. For example, during the installation of the directory server, the following ACI is created:

```
aci: (targetattr="*")(targetfilter=(o=example))
(version 3.0; acl "Default anonymous access";
allow (read, search) userdn="ldap:///anyone";)
```

This ACI can apply only to the `o=example` entry, because that is the only entry with an attribute `o` having the value `example`.

The risk associated with these methods is that your directory tree can change in the future, and you would have to remember to modify this ACI.

#### 8.2.2.7 To Specify the Scope of an ACI

Usually an ACI has subtree scope. You can restrict the scope of an ACI by using the `targetscope` keyword with the following syntax:

`(targetscope="`*expression*`")`

where *expression* is one of the following:

**`base`**
The ACI applies to the target resource only.

**`onelevel`**
The ACI applies to the target resource's first-generation children.

**`subtree`**
The ACI applies to the target resource and the subtree below it.

**`subordinate`**
The ACI applies only to the subtree below the target resource.

If the `targetscope` is not specified, the default value is `subtree`. The following example restricts the ACI target match only to the entry with the distinguished name `uid=bjensen,ou=People,dc=example,dc=com` and any of the children one level below it:

```
(target =
"ldap:///uid=bjensen,ou=People,dc=example,dc=com")(targetscope="onelevel")
```

**Note:** The not-equal operator is not supported for the `targetscope` keyword.

### 8.2.2.8 To Target LDAP Controls

To target LDAP controls, use the `targetcontrol` keyword and provide the control Appendix D.14.3, "object identifier." The `targetcontrol` keyword uses the following syntax:

```
(targetcontrol="oid")
```

```
(targetcontrol!="oid")
```

You can target multiple LDAP controls by using the `targetcontrol` keyword with the following syntax:

```
(targetcontrol="oid1 || oid2... || oidN")
```

```
(targetcontrol!="oid1 || oid2... || oidN")
```

For example, to target both the Appendix D.7.2, "get effective rights control" and the Appendix D.15.20, "proxied authorization control", use the following `targetcontrol` expression:

```
(targetcontrol = "1.3.6.1.4.1.42.2.27.9.5.2 || 2.16.840.1.113730.3.4.18")
```

**Note:** The get effective rights control has OID value of 1.3.6.1.4.1.42.2.27.9.5.2. The proxy authorization V2 control has OID value of 2.16.840.1.113730.3.4.18.

### 8.2.2.9 To Target LDAP Extended Operations

To target extended operations, use the `extop` keyword and provide the operation Appendix D.14.3, "object identifier." The `extop` keyword uses the following syntax:

```
(extop= "oid")
```

```
(extop!= "oid")
```

You can target multiple extended operations by using the `extop` keyword with the following syntax:

```
(extop = "oid1 || oid2... || oidN")
```

```
(extop!= "oid1 || oid2... || oidN")
```

For example, to target both the Appendix D.18.23, "StartTLS extended operation" and the Appendix D.15.5, "Password Modify extended operation", use the following `extop` expression:

```
(extop = "1.3.6.1.4.1.1466.20037 || 1.3.6.1.4.1.4203.1.11.1.")
```

**Note:** Access control using the `extop` keyword with a StartTLS extended operation target must always be done using Global ACIs. The authorization entry in the StartTLS extended operation is null.

## 8.2.3 Defining Permissions

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

There are two parts to setting permissions:

- Allowing or denying access
- Assigning rights

The following sections describe how to define permissions:


- Section 8.2.3.1, "To Allow or Deny Access"
- Section 8.2.3.2, "To Assign Rights"
- Section 8.2.3.3, "Rights Required for LDAP Operations"
- Section 8.2.3.4, "Permissions Syntax"


### 8.2.3.1 To Allow or Deny Access

You can explicitly allow or deny access permissions by using the `allow` or the `deny` keyword.

### 8.2.3.2 To Assign Rights

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

**Read**
Indicates whether users can read the directory entries and the attributes of entries specified in the ACI. This permission applies only to the search operation. (Compare the Read permission with the description of the Search permission that follows.)

**Write**
Indicates whether users can modify an entry by adding, modifying, or deleting attributes. This permission applies to the modify and modRDN operations.

**Add**
Indicates whether users can create entries. This permission applies only to the add operation.

**Delete**
Indicates whether users can delete entries. This permission applies only to the delete operation.

**Search**
Indicates whether users can search on the targets specified in the ACI. This permission applies only to the search operation. The Search right is checked once, and after the search is allowed or denied, it is not checked again. If the search is allowed, the read right is then applied to each entry to be returned as a result of the search and to each attribute of each entry.

**Compare**
Indicates whether users can compare data they supply with data stored in the directory. With compare rights, the directory returns a success or failure message in response to an inquiry, but the user cannot see the value of the entry or attribute. This permission applies only to the compare operation.

**Selfwrite**
Indicates whether users can add or delete their own DN in an attribute of the target entry. The syntax of this attribute must be a distinguished name. This right is used only for group management. Selfwrite works with proxy authorization: it grants the right to add or delete the proxy DN from the group entry (not the DN of the bound user).

**Proxy**
Indicates whether the specified DN can access the target with the rights of another entry. You can grant proxy access using the DN of any user in the directory except the Directory Manager DN. Moreover, you cannot grant proxy rights to the Directory Manager. An example is provided in Section 22.6, "Proxy Authorization ACIs."

**Import**
Used by the modify DN operation. This access right indicates whether an entry can be imported to the specified DN.

**Export**
Used by the modify DN operation. This access right indicates whether an entry can be exported from the specified DN.

**All**
Indicates that the specified DN has the following rights to the targeted entry: read, write, search, delete, compare, and selfwrite. The All access right does not give the following rights to the target entry: proxy, import, and export.

Rights are granted independently of one another. This means, for example, that a user who is granted add rights can create an entry but cannot delete it if delete rights have not been specifically granted. Therefore, when planning the access control policy for your directory, you must ensure that you grant rights in a way that makes sense for users. For example, it does not usually make sense to grant write permission without granting read and search permissions.

#### 8.2.3.3 Rights Required for LDAP Operations

This section describes the rights that you need to grant to users depending on the type of LDAP operation that you want to authorize them to perform.

- Adding an entry
  - Grant add permission on the entry being added.
  - Grant write permission on the value of each attribute in the entry. This right is granted by default but could be restricted using the `targattrfilters` keyword.
- Deleting an entry
  - Grant delete permission on the entry to be deleted.
  - Grant write permission on the value of each attribute in the entry. This right is granted by default but could be restricted using the `targattrfilters` keyword.
- Modifying an attribute in an entry
  - Grant write permission on the attribute type.
  - Grant write permission on the value of each attribute type. This right is granted by default but could be restricted using the `targattrfilters` keyword.

- Modifying the RDN of an entry
  - Grant write permission on the entry.
  - Grant write permission on the attribute type used in the new RDN.
  - Grant write permission on the attribute type used in the old RDN, if you want to grant the right to delete the old RDN.
  - Grant write permission on the value of the attribute type used in the new RDN. This right is granted by default but could be restricted using the `targattrfilters` keyword.
- Moving an entry to another subtree
  - Grant export permissions on the entry that you want to move.
  - Grant import permission on the new superior entry of the entry that you want to move.
- Comparing the value of an attribute
  - Grant compare permission on the attribute type.
- Searching for entries
  - Grant search permission on each attribute type used in the search filter.
  - Grant read permission on at least one attribute type used in the entry to ensure that the entry is returned.
  - Grant read permission an each attribute type to be returned with the entry.

The permissions you need to set up to allow users to search the directory are more readily understood with an example. Consider the following search:

```
$ ldapsearch -h host -p port -D "uid=bjensen,dc=example,dc=com" \
  -j pwd-file -b "dc=example,dc=com" \
  "(objectclass=*)" mail
```

The following ACI is used to determine whether user `bjensen` can be granted access for searching her own entry:

```
aci: (targetattr = "mail")(version 3.0; acl "self access to \
mail"; allow (read, search) userdn = "ldap:///self";)
```

The search result list is empty because this ACI does not allow `bjensen` the right to search on the `objectclass` attribute. For the search operation to be successful, you must modify the ACI, as shown in the following example:

```
aci: (targetattr = "mail || objectclass")(version 3.0; acl \
"self access to mail"; allow (read, search) userdn = \
"ldap:///self";)
```

#### 8.2.3.4 Permissions Syntax

In an ACI statement, permissions use the following syntax:

`allow|deny (`*rights*`)`

where *rights* is a list of comma-separated keywords enclosed within parentheses. Valid keywords are `read`, `write`, `add`, `delete`, `search`, `compare`, `selfwrite`, `proxy`, `import`, `export`, or `all`.

The `all` access right does not give the following rights to the target entry: `proxy`, `import`, and `export`.

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

```
aci: (target="ldap:///dc=example,dc=com") (version 3.0;acl \
"example"; allow (read, search, compare) bindRule;)
```

## 8.3 Bind Rules

Depending on the ACIs defined for the directory, for certain operations, you need to bind to the directory. The following sections describe how bind rules are used to control access:

- Section 8.3.1, "Bind Rules Overview"
- Section 8.3.2, "Using Boolean Bind Rules"

### 8.3.1 Bind Rules Overview

Binding means logging in or authenticating yourself to the directory by providing a bind DN and password, or, if using SSL, a certificate. The credentials provided in the bind operation and the circumstances of the bind determine whether access to the directory is allowed or denied.

Every permission set in an ACI has a corresponding bind rule that details the required credentials and bind parameters.

A simple bind rule might require that the person accessing the directory belong to a specific group. A complex bind rule can state that a person must belong to a specific group and must log in from a machine with a specific IP address between 8 a.m. and 5 p.m.

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

- Users, groups, and roles that are granted access
- Location from which an entity must bind (The location from which a user authenticates can be spoofed and can therefore not be trusted. Do not base ACIs on this information alone.)
- Time or day on which binding must occur
- Type of authentication that must be in use during binding
- Security strength factor (that is, the length of encryption key currently in use)

Additionally, bind rules can be complex constructions that combine these criteria by using Boolean operators, as described in Section 8.4, "Bind Rule Syntax."

The directory server evaluates the logical expressions used in ACIs according to a three-valued logic similar to the one used to evaluate LDAP filters, as described in RFC 4511 (`http://www.ietf.org/rfc/rfc4511.txt`) Lightweight Directory Access Protocol (LDAP): The Protocol. In summary, this means that if any component in the expression evaluates to Undefined (for example if the evaluation of the expression aborted due to a resource limitation), then the directory server handles this case correctly: it does not erroneously grant access because an Undefined value occurred in a complex Boolean expression.

### 8.3.2 Using Boolean Bind Rules

Bind rules can be complex expressions that use the Boolean expressions `AND`,`OR`, and `NOT` to set very precise access rules. When creating boolean bind rules, always use parentheses to define the order in which rules are to be evaluated. A trailing semicolon is a required delimiter that must appear after the final rule.

For example, to bind with `bindRuleA`, and with either `bindRuleB`, or with either `bindRuleC` and `bindRuleD`, use the following syntax:

```
(bindRuleA and (bindRuleB or (bindRuleC and bindRuleD));)
```

Using another example, the following bind rule is evaluated to be true if the bind DN client is accessed from within the `example.com` domain and is a member of either the administrators group or both the mail administrators and calendar administrators groups.

```
(dns = "*.example.com" and (groupdn =
"ldap:///cn=administrators,dc=example,dc=com" or
(groupdn = "ldap:///cn=mail administrators,dc=example,dc=com" and
groupdn = "ldap:///cn=calendar administrators,dc=example,dc=com"));)
```

The || operator is allowed only in the `groupdn` bind rule keyword expression. For all other bind rule expressions, the `or` operator must be used.

## 8.4 Bind Rule Syntax

Whether access is allowed or denied depends on whether an ACI's bind rule is evaluated to be true. The following sections describe the bind rule syntax and the various keywords that can be used to allow or deny access.

- Section 8.4.1, "Bind Rule Syntax Overview"
- Section 8.4.2, "Defining User Access (`userdn` Keyword)"
- Section 8.4.3, "Defining Group Access (`groupdn` Keyword)"
- Section 8.4.4, "Defining Access Based on Value Matching (`userattr` Keyword)"
- Section 8.4.5, "Defining Access From a Specific IP Address (`ip` Keyword)"
- Section 8.4.6, "Defining Access From a Specific Domain (`dns` Keyword)"
- Section 8.4.7, "Defining Access at a Specific Time of Day or Day of Week (`timeofday` and `dayofweek` Keywords)"
- Section 8.4.8, "Defining Access Based on Authentication Method (`authmethod` Keyword)"
- Section 8.4.9, "Defining Access Based on a Connection's Security Strength Factor (`ssf` Keyword)"

### 8.4.1 Bind Rule Syntax Overview

Bind rules use one of the following patterns:

- *keyword* `="` *expression* `";`
- *keyword* `!="` *expression* `";`

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

The quotation marks ("") around the expression and the delimiting semicolon (;) are required. The expressions you can use depend on the associated keyword.

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

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

| Keyword | Valid Expressions | Wildcard Allowed? |
|---|---|---|
| Defining User Access (`userdn` Keyword) | `ldap:///`*distinguishedName*<br>`ldap:///all`<br>`ldap:///anyone`<br>`ldap:///self`<br>`ldap:///parent`<br>`ldap:///`*suffix*`??sub?(`*filter*`)` | Allowed, in DN only |
| Defining Group Access (`groupdn` Keyword) | `ldap:///`*DN* | Not Allowed |
| Defining Access Based on Value Matching (`userattr` Keyword) | *attribute*`#`*bindType* or *attribute*`#`*value* | Not Allowed |
| Defining Access From a Specific IP Address (`ip` Keyword) | *IPaddress* | Allowed |
| Defining Access From a Specific Domain (`dns` Keyword) | *DNShostName* | Allowed |
| Defining Access at a Specific Time of Day or Day of Week (`timeofday` and `dayofweek` Keywords) | `sun`<br>`mon`<br>`tue`<br>`wed`<br>`thu`<br>`fri`<br>`sat` | Not Allowed |
| Defining Access at a Specific Time of Day or Day of Week (`timeofday` and `dayofweek` Keywords) | *hhmm* where *hh* is in the range `00-24` and *mm* is in the range `00-60` | Not Allowed |
| Defining Access Based on Authentication Method (`authmethod` Keyword) | `none`<br>`simple`<br>`ssl`<br>`sasl`<br>*authenticationMethod* | Not Allowed |
| Defining Access Based on a Connection's Security Strength Factor (`ssf` Keyword) | 0-256 | Not Allowed |

The following sections provide additional information about the bind rule syntax for each keyword.

## 8.4.2 Defining User Access (`userdn` Keyword)

User access is defined using the `userdn` keyword. The `userdn` keyword requires one or more valid distinguished names in the following format:

```
userdn = "ldap:///dn [|| ldap:///dn]..."
```

```
userdn!= "ldap:///dn [|| ldap:///dn]..."
```

where *dn* can be a DN or one of the expressions `anyone`, `all`, `self`, or `parent`. These expressions refer to the following users:

**`userdn = "ldap:///anyone"`**
Both anonymous and authenticated users

**`userdn = "ldap:///all"`**
Only authenticated users

**`userdn = "ldap:///self"`**
Only the same user as the target entry of the ACI

**`userdn = "ldap:///parent"`**
Only the parent entry of the ACI target

The `userdn` keyword can also be expressed as an LDAP filter in this form:

```
userdn = ldap:///suffix??sub?(filter)
```

Characters that are syntactically significant for a DN, such as commas, must be escaped with a single backslash (`\`).

The following sections describe how to define user access with the `userdn` keyword:

- Section 8.4.2.1, "Defining General Access (`all` Keyword)"
- Section 8.4.2.2, "Defining Anonymous Access (`anyone` Keyword)"
- Section 8.4.2.3, "Defining Self Access (`self` Keyword)"
- Section 8.4.2.4, "Defining Parent Access (`parent` Keyword)"
- Section 8.4.2.5, "Specifying Users With LDAP URLs"
- Section 8.4.2.6, "Specifying Users With Wildcards"
- Section 8.4.2.7, "Specifying Users With a Logical OR of LDAP URLs"
- Section 8.4.2.8, "Excluding Specific LDAP URLs"

### 8.4.2.1 Defining General Access (`all` Keyword)

You can use bind rules to indicate that a permission applies to anyone who has successfully bound to the directory. The `all` keyword therefore allows access by all authenticated users. This allows general access while preventing anonymous access.

For example, to grant read access to the entire tree to all authenticated users, create the following ACI on the `dc=example,dc=com` node:

```
aci: (version 3.0; acl "all-read"; allow (read)
userdn="ldap:///all";)
```

### 8.4.2.2 Defining Anonymous Access (`anyone` Keyword)

Granting anonymous access to the directory means that anyone can access it without providing a bind DN or password, regardless of the circumstances of the bind. You can limit anonymous access to specific types of access (for example, access for read or

access for search) or to specific subtrees or individual entries within the directory. Anonymous access using the `anyone` keyword also allows access by any authenticated user.

For example, to allow anonymous read and search access to the entire `example.com` tree, create the following ACI on the `dc=example,dc=com` node:

```
aci: (version 3.0; acl "anonymous-read-search";
allow (read, search) userdn = "ldap:///anyone";)
```

#### 8.4.2.3 Defining Self Access (`self` Keyword)

Specifies that users are granted or denied access to their own entries. In this case, access is granted or denied if the bind DN matches the DN of the targeted entry. For example, to grant all users in the `example.com` tree write access to their `userPassword` attribute, create the following ACI on the `dc=example,dc=com` node.

```
aci: (targetattr = "userPassword") (version 3.0; acl
"modify own password"; allow (write) userdn = "ldap:///self";)
```

#### 8.4.2.4 Defining Parent Access (`parent` Keyword)

Specifies that users are granted or denied access to the entry only if their bind DN is the parent of the targeted entry. For example, to allow users to modify any child entries of their bind DN, create the following ACI on the `dc=example,dc=com` node:

```
aci: (version 3.0; acl "parent access";
allow (write) userdn="ldap:///parent";)
```

#### 8.4.2.5 Specifying Users With LDAP URLs

You can dynamically target users in ACIs using a URL with a filter as shown in the following example:

```
userdn = "ldap:///suffix??sub?(filter)"
```

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

```
userdn = "ldap:///dc=example,dc=com??sub?(|(ou=eng)(ou=acct))"
```

Do not specify a host name or port number within the LDAP URL. LDAP URLs always apply to the local directory server.

#### 8.4.2.6 Specifying Users With Wildcards

You can also specify a set of users by using the wildcard character (`*`). For example, specifying a user DN of `uid=b*,dc=example,dc=com` indicates that only users with a bind DN beginning with the letter `b` is allowed or denied access based on the permissions you set.

#### 8.4.2.7 Specifying Users With a Logical OR of LDAP URLs

Specify several LDAP URLs or keyword expressions to create complex rules for user access. For example:

```
userdn = "ldap:///uid=b*,c=example.com ||
ldap:///cn=b*,dc=example,dc=com";
```

The bind rule is evaluated to be true for users binding with either of the DN patterns.

#### 8.4.2.8 Excluding Specific LDAP URLs

Use the not-equal (!=) operator to define user access that excludes specific URLs or DNs. For example:

```
userdn != "ldap:///uid=*,ou=Accounting,dc=example,dc=com";
```

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

### 8.4.3 Defining Group Access (`groupdn` Keyword)

Members of a specific group can access a targeted resource. This is known as group access. Group access is defined using the `groupdn` keyword to specify that access to a targeted entry is granted or denied if the user binds using a DN that belongs to a specific group.

The `groupdn` keyword requires the distinguished name of one or more groups in the following format:

```
groupdn="ldap:///groupDN [|| ldap:///groupDN]..."
```

The bind rule is evaluated to be true if the bind DN belongs to a group specified by any of the group DNs. The following section give examples using the `groupdn` keyword.

Characters that are syntactically significant for a DN, such as commas, must be escaped with a single backslash (\).

#### 8.4.3.1 Specifying a Group With a Single LDAP URL

```
groupdn = "ldap:///cn=Administrators,dc=example,dc=com";
```

The bind rule is evaluated to be true if the bind DN belongs to the Administrators group. For example, to grant the Administrators group permission to write to the entire directory tree, create the following ACI on the `dc=example,dc=com` node:

```
aci: (version 3.0; acl "Administrators-write"; allow (write)
groupdn="ldap:///cn=Administrators,dc=example,dc=com";)
```

#### 8.4.3.2 Specifying a Group With a Logical OR of LDAP URLs

```
groupdn = "ldap:///cn=Administrators,dc=example,dc=com ||
ldap:///cn=Mail Administrators,dc=example,dc=com";
```

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

### 8.4.4 Defining Access Based on Value Matching (`userattr` Keyword)

The `userattr` keyword can be used to specify which attribute values must match between the entry used to bind (bind entry) and the targeted entry. A `userattr` expression has two formats, a bind-type format and an attribute-value format.

The following sections describe how to define access based on value matching:

- Section 8.4.4.1, "Bind-Type Format"
- Section 8.4.4.2, "Attribute-Value Format"
- Section 8.4.4.3, "`USERDN` Bind Type Example"

#### 8.4.4.1 Bind-Type Format

This format is named the bind-type format because it uses the bind DN and possibly the bind entry when evaluating a match. It is the more complicated of the two formats. The bind-type format can be used in the following three ways:

- Treat a target entry attribute value as a DN that must match the bind DN
- Treat a target entry attribute value as a group DN that the bind DN must be a member of
- Require that both the bind DN and the bind entry match an LDAP URL specified in a target entry attribute value

The bind-type `userattr` format uses this syntax:

```
userattr = "attrName#bindType"
```

where:

***attrName***
Is the name of the attribute in the target entry.

***bindType***
Must be one of the following:

- `USERDN` — The value of *attrName* must match the bind DN.
- `GROUPDN` — The value of *attrName* is a group that must contain the bind DN.
- `LDAPURL` — The value of *attrName* is an Appendix D.20.5, "URL" that is treated as a search that the bind DN and entry must match. To satisfy the search, the URL's *dn* value is used as a base DN that the bind DN must match or have as a parent DN. The URL's *scope* value restricts how far below the base DN the bind DN can match. Finally, the bind entry must match the URL's *filter* value.

The bind type `userattr` format has a special parent keyword that allows targeting of entries levels below the current target entry. See Section 8.4.4.7, "Inheritance" for more details on this keyword.

#### 8.4.4.2 Attribute-Value Format

The attribute-value format requires the following two conditions to match:

- An attribute specified in the `userattr` expression must exist in both the target and bind entries.
- The values of both of these attributes must match a string value specified in the `userattr` expression. This string value cannot be one of the bind type keywords (`USERDN`, `GROUPDN`, `LDAPURL`).

The attribute value `userattr` format uses this syntax:

```
userattr = "attrName#attrValue"
```

where:

***attrName***
The name of the attribute in both the target and bind entries.

***attrValue***
The string representing the attribute value (not `USERDN`, `GROUPDN` or `LDAPURL`).

#### 8.4.4.3 `USERDN` Bind Type Example

The following example of a bind rule `userattr` keyword expression specifies a match between the bind DN and the value of the target entry attribute `manager`.

```
userattr = "manager#USERDN"
```

This bind rule is evaluated to be true if the bind DN matches the value of the `manager` attribute in the target entry. The `manager` attribute in the target entry must match the bind DN. Wildcards are not allowed.

The following example ACI grants a manager full access to all user attributes of entries located in the subtree under the DN `dc=example, dc=comm`:

```
aci: (target="ldap:///dc=example,dc=com")(targetattr="*")
(version 3.0;acl "manager all access";
allow (all) userattr = "manager#USERDN";)
```

#### 8.4.4.4 `GROUPDN` Bind Type Example

This is an example of a bind rule `userattr` keyword expression specifying an attribute that contains a group DN that the bind DN must be a member of.

```
userattr = "owner#GROUPDN"
```

The bind rule is evaluated to be true if the bind DN is a member of the group specified in the `owner` attribute of the target entry.

#### 8.4.4.5 `LDAPURL` Bind Type Example

This is an example of a bind rule `userattr` keyword expression specifying an attribute that contains an LDAP URL that is treated as a search that the bind DN and entry must match.

```
userattr = "aciurl#LDAPURL"
```

The attribute `aciurl` is an example only.

The bind rule is evaluated to true if the bind DN and bind entry satisfy all of the search requirements specified in the LDAP URL. For example, if the value of `aciurl` is `ldap:///dc=example,dc=com??one?(cn=joe*)`, then the bind DN must satisfy a one-level search under the base DN of `dc=example,dc=com` and the bind entry must satisfy the filter `(cn=joe*)`.

#### 8.4.4.6 Attribute Value Example

The following example of the bind rule `userattr` keyword expression specifies an attribute value that both the bind entry and target entry must match.

```
userattr = "favoriteBeverage#Water"
```

The bind rule is evaluated to be true if the bind and target entries include the `favoriteBeverage` attribute with a value of `Water`.

#### 8.4.4.7 Inheritance

When you use the `userattr` keyword to associate the entry used to bind with the target entry, the ACI applies only to the target specified and not to the entries below it. In some circumstances, you might want to extend the application of the ACI several levels below the targeted entry. This is possible by using the parent keyword and specifying the number of levels below the target that should inherit the ACI.

When you use the `userattr` keyword in association with the parent keyword, the syntax is as shown in the following example:

```
userattr = "parent[[inheritanceLevel].attribute#bindType"
```

where:

- *inheritanceLevel* is a comma-separated list that indicates how many levels below the target inherit the ACI. You can include ten levels [0,1,2,3,4,..,9] below the targeted entry. Zero (0) indicates the targeted entry.
- *attribute* is the attribute targeted by the `userattr`.
- *bindType* can be either `USERDN` or `GROUPDN`. The `LDAPURL` bind type is not supported with inheritance.

For example, the `userattr = "parent[[0,1].manager#USERDN"` bind rule is evaluated to be true if the bind DN matches the `manager` attribute of the target entry. Also, the bind rule is evaluated to be true for all entries immediately below the target entry (one level below the target) that have `manager` attributes matching the bind DN.

#### 8.4.4.8 Inheritance Example

The following example indicates that user `bjensen` is allowed to read and search the `cn=Profiles` entry as well as the first level of child entries, which includes `cn=mail` and `cn=news`.

```
cn=Profiles
aci:(targetattr="*")(version 3.0, acl "profiles access" allow(read, search)
userattr="parent[[0,1].owner#USERDN;)
owner=cn=bjensen, ou=people, dc=example, dc=com
cn=mail, cn=Profiles
mailuser: bjensen
cn=news, cn=Profiles
newuser: bjensen
```

If inheritance were not used in this example, you would need to do one of the following:

- Explicitly set read and search access for user `bjensen` on the `cn=Profiles`, `cn=mail`, and `cn=news` entries in the directory.
- Add the owner attribute and the following ACI to the `cn=mail`, `cn=Profiles` and `cn=news,cn=Profiles` entries:

  ```
  aci: (targetattr="*") (version 3.0; acl "profiles access"; allow
  (read,search) userattr="owner#USERDN";)
  ```

#### 8.4.4.9 Add Permissions

If you use the `userattr` keyword in conjunction with all or add permissions, you might find that the behavior of the directory server is not what you expect. Typically, when a new entry is created in the directory, the directory server evaluates access rights on the entry being created, and not on the parent entry. However, in the case of

ACIs using the `userattr` keyword, this behavior could create a security hole, so the directory server's normal behavior is modified to avoid it.

Consider the following example ACI:

```
aci: (target="ldap:///dc=example,dc=com")(targetattr="*")
(version 3.0; acl "manager-write"; allow (all)
userattr = "manager#USERDN";)
```

This ACI grants managers all rights on the entries of employees that report to them. However, because access rights are evaluated on the entry being created, this type of ACI would also allow any employee to create an entry in which the manager attribute is set to their own DN. For example, disgruntled employee Joe, `cn=Joe,ou=eng,dc=example,dc=com`, might want to create an entry in the Human Resources branch of the tree to use (or misuse) the privileges granted to Human Resources employees.

He could do this by creating the following entry:

```
dn: cn= Trojan Horse,ou=Human Resources,dc=example,dc=com
objectclass: top
...
cn: Trojan Horse
manager: cn=Joe,ou=eng,dc=example,dc=com
```

To avoid this type of security threat, the ACI evaluation process does not grant add permission at *level 0*, that is, to the entry itself. You can, however, use the `parent` keyword to grant add rights below existing entries. You must specify the number of levels below the parent for add rights. For example, the following ACI allows child entries to be added to any entry in the `dc=example,dc=com` that has a manager attribute that matches the bind DN:

```
aci: (target="ldap:///dc=example,dc=com")(targetattr="*")
(version 3.0; acl "parent-access"; allow (add)
userattr = "parent[1].manager#USERDN";)
```

This ACI ensures that add permission is granted only to users whose bind DN matches the `manager` attribute of the parent entry.

## 8.4.5 Defining Access From a Specific IP Address (`ip` Keyword)

Using bind rules, you can indicate that the bind operation must originate from a specific IP address. This is often used to force all directory updates to occur from a given machine or network domain.

The LDIF syntax for setting a bind rule based on an IP address is shown in the following examples:

```
ip = "IPaddressList"
ip != "IPaddressList"
```

The *IPaddressList* is a list of one or more comma-separated elements from among any of the following:

- A specific IPv4 address, such as `123.45.6.7`
- An IPv4/CIDR-compliant address, such as `192.168.0.0/16`
- An IPv4 address with wildcards to specify a subnetwork, such as `12.3.45.*`
- An IPv4 address or subnetwork with a subnetwork mask, such as `123.45.6.*+255.255.255.192`

- An IPv6 address in any of its legal forms and contained in square brackets [and], as defined by RFC 2373 (http://www.ietf.org/rfc/rfc2373.txt) and RFC 2732 (http://www.ietf.org/rfc/rfc2732.txt). The following addresses are equivalent:
  - ldap://[12AB:0000:0000:CD30:0000:0000:0000:0000]
  - ldap://[12AB::CD30:0:0:0:0]
  - ldap://[12AB:0:0:CD30::]
- An IPv6 address with a subnet prefix length, such as ldap://[12AB::CD30:0:0:0:0]/60

The bind rule is evaluated to be true if the client accessing the directory is located at the named IP address. This can be useful for allowing certain kinds of directory access only from a specific subnet or machine. Note that the IP address from which a user authenticates can be spoofed, and can therefore not be trusted. Do not base ACIs on this information alone.

### 8.4.6 Defining Access From a Specific Domain (dns Keyword)

A bind rule can specify that the bind operation must originate from a particular domain or host machine. This is often used to force all directory updates to occur from a given machine or network domain.

The LDIF syntax for setting a bind rule based on the DNS host name is as shown here:

```
dns = "DNShostname"
dns != "DNShostname"
```

> **Caution:** The dns keyword requires that the naming service used on your machine is DNS. If the naming service is not DNS, use the ip keyword instead.

The dns keyword requires a fully qualified DNS domain name. Granting access to a host without specifying the domain creates a potential security threat. For example, the following expression is allowed but not recommended:

```
dns = "legend.eng";
```

You should use a fully qualified name such as:

```
dns = "legend.eng.example.com";
```

The dns keyword allows wildcards. For example:

```
dns = "*.example.com";
```

The bind rule is evaluated to be true if the client accessing the directory is located in the named domain. This can be useful for allowing access only from a specific domain. Note that wildcards do not work if your system uses a naming service other than DNS. In such a case, if you want to restrict access to a particular domain, use the ip keyword, as described in Section 8.4.5, "Defining Access From a Specific IP Address (ip Keyword)."

### 8.4.7 Defining Access at a Specific Time of Day or Day of Week (`timeofday` and `dayofweek` Keywords)

You can use bind rules to specify that binding can only occur at a certain time of day or on a certain day of the week. For example, you can set a rule that allows access only if the time is between the hours of 8 a.m. and 5 p.m. Monday through Friday. The time used to evaluate access rights is the time on the directory server, not the time on the client.

The LDIF syntax for setting a bind rule based on the time of day is as shown here:

```
timeofday operator "time"
```

where *operator* can be one of the following symbols:

- = (equal to)
- != {not equal to}
- > (greater than)
- >= (greater than or equal to)
- < (less than)
- <= (less than or equal to)

The time is expressed as four digits representing hours and minutes in the 24-hour clock (*hhmm* where *hh* is in the range `00-24` and *mm* is in the range `00-60`). For example:

- `timeofday = "1200";` is true if the client is accessing the directory during the minute that the system clock shows noon.
- `timeofday!= "0100";` is true for access at any other time than 1 a.m.
- `timeofday> "0800";` is true for access from 8:01 a.m. through 11:59 p.m.
- `timeofday>= "0800";` is true for access from 8:00 a.m. through 11:59 p.m.
- `timeofday< "1800";` is true for access from 12:00 midnight through 5:59 p.m.

The time and date on the directory server are used for the evaluation of the `timeofday` and `dayofweek` bind rules and not the time on the client.

The LDIF syntax for setting a bind rule based on the day in the week is as shown here:

```
dayofweek = "day1, day2 ..."
```

The possible values for the `dayofweek` keyword are the English three-letter abbreviations for the days of the week: `sun`, `mon`, `tue`, `wed`, `thu`, `fri`, `sat`. Specify all days you want to grant access, for example:

```
dayofweek = "mon, tue, wed, thu, fri";
```

The bind rule is true if the directory is being accessed on one of the days listed.

### 8.4.8 Defining Access Based on Authentication Method (`authmethod` Keyword)

You can set bind rules that state that a client must bind to the directory using a specific authentication method. The following authentication methods are available:

**`None`**
Authentication is not required. This is the default. It represents anonymous access.

**`Simple`**
The client must provide a user name and password to bind to the directory.

**`SSL`**
The client must bind to the directory over a Secure Sockets Layer (SSL) or Transport Layer Security (TLS) connection.

In the case of SSL, the connection is established to the LDAPS second port. In the case of TLS, the connection is established through a Start TLS operation. In both cases, a certificate must be provided. For information on setting up SSL, see Section 20.6, "Using SASL Authentication."

**`SASL`**
The client must bind to the directory using a Simple Authentication and Security Layer (SASL) mechanism, such as DIGEST-MD5 or GSSAPI.

The LDIF syntax for setting a bind rule based on an authentication method is as shown here:

```
authmethod = "authentication_method"
```

where `authentication_method` is `none`, `simple`, `ssl`, or `sasl sasl_mechanism`.

#### 8.4.8.1 Authentication Method Examples

The following examples show typical specifications of the `authmethod` keyword:

**`authmethod = "none"`**
Authentication is not checked during bind rule evaluation.

**`authmethod = "simple"`**
The bind rule is evaluated to be true if the client is accessing the directory using a user name and password.

**`authmethod = "ssl"`**
The bind rule is evaluated to be true if the client authenticates to the directory using a certificate over LDAPS. It is not true if the client authenticates using simple authentication (bind DN and password) over LDAPS.

**`authmethod = "sasl DIGEST-MD5"`**
The bind rule is evaluated to be true if the client is accessing the directory using the SASL DIGEST-MD5 mechanism. Other supported SASL mechanisms are EXTERNAL and GSSAPI.

### 8.4.9 Defining Access Based on a Connection's Security Strength Factor (`ssf` Keyword)

You can use bind rules to specify that binding can only occur based on a specific level of Security Strength Factor (SSF) enforced on the established connection. A connection's SSF is based on the key strength of the cipher enforced on the connection and pertains only to TLS/SSL or DIGEST-MD5/GSSAPI confidentiality or integrity connections.

The LDIF syntax for setting a bind rule based on the Security Strength Factor is shown here:

`ssf` *operator* `"`*strength*`"`

where *operator* can be one of the following symbols:

- = (equal to)
- != (not equal to)
- > (greater than)
- >= (greater than or equal to)
- < (less than)
- <= less than or equal to

The strength is a value representing the cipher key strength required on the connection and is a value (`0` to `256`). DIGEST-MD5/GSSAPI connections with integrity enforced have an SSF of `1`. TLS/SSL and DIGEST-MD5/GSSAPI confidentiality connections can have variable values of SSF based on the cipher negotiation performed between the directory server and client. The higher a connection's negotiated SSF is, the stronger the encryption is on the connection, as shown in these examples:

- `ssf = "1";` is true for access if integrity `ssf = 1` only is enforced on the connection.
- `ssf!= "40";` is true for access if `ssf not equal 40` is enforced on the connection.
- `ssf> "128";` is true for access if `ssf greater than 128` is enforced on the connection.
- `ssf>= "128";` is true for access if `ssf greater than or equal 128` is enforced on the connection.
- `ssf< "56";` is true for access if `ssf less than 56` is enforced on the connection.

Clear connections have an SSF of `0`.

The following sections describe how to define based on a connection's security strength factor keyword

#### 8.4.9.1 DIGEST-MD5 QOP Key Size Mapping

The following table illustrates the Quality of Protection (QOP) to cipher key size mapping.

| Cipher | QOP | Description |
|---|---|---|
| RC4 (40) | Low | RC4 cipher with 40-bit key (obsolete) |
| RC4 (56) | Medium | RC4 cipher with 56-bit key |
| DES | Medium | Data Encryption Standard (DES) cipher in cipher block chaining (CBC) mode with a 56-bit key |
| RC4 (128) | High | RC4 cipher with 128-bit key |
| Triple DES | High | Triple DES cipher in CBC mode with EDE with the same key for each E stage (also called "two keys mode") for a total key length of 112 bits |

#### 8.4.9.2 TLS Cipher Key Size Mapping

| Cipher | TLS RFC | Key Size | Description |
|---|---|---|---|
| RC2_CBC_40 | 4346 | 40 | RC2 cipher in cipher block chaining (CBC) mode (obsolete) |

| Cipher | TLS RFC | Key Size | Description |
|---|---|---|---|
| RC4_40 | 4346 | 40 | RC4 cipher (obsolete) |
| DES40_CBC | 4346 | 40 | DES 40-bit cipher in cipher block chaining (CBC) mode (obsolete) |
| DES_CBC | 4346 | 56 | DES 56-bit in cipher block chaining (CBC) mode cipher |
| 3DES_EDE_CBC | 4346 | 112 | TDES |
| RC4_128 | 4346 | 128 | RC4 cipher |
| IDEA_CBC | 4346 | 128 | International Data Encryption Algorithm (IDEA) cipher in cipher block chaining (CBC) mode |
| SEED_CBC | 4162 | 128 | SEED cipher in cipher block chaining (CBC) mode |
| CAMELLIA_128_CBC | 4132 | 128 | Camellia cipher in cipher block chaining (CBC) mode |
| AES_128_CBC | 3268 | 128 | Advanced Encryption Standard (AES) in cipher block chaining (CBC) mode |
| AES_256_CBC | 3268 | 256 | Advanced Encryption Standard (AES) in cipher block chaining (CBC) mode |
| CAMELLIA_256_CBC | 4132 | 256 | Camellia cipher in cipher block chaining (CBC) mode |
| AES_256_GCM | 5288 | 256 | AES in Galois Counter Mode (GCM) |

#### 8.4.9.3 Example

The following ACI allows users to change their own passwords only over a connection with an SSF strength equal to or greater than 128:

```
(targetattr="userPassword||authPassword")(version 3.0; acl "User change pwd";
(allow (write) userdn="ldap:///self" and ssf >= "128");)
```

## 8.5 Compatibility With the Oracle Directory Server Enterprise Edition Access Control Model

The following sections describe how the Oracle Unified Directory access control model differs from the access control model provided with Oracle Directory Server Enterprise Edition.

- Section 8.5.1, "Global ACI"
- Section 8.5.2, "Distinguished Name (DN) Wildcard Matching"
- Section 8.5.3, "Privilege Subsystem Impact"
- Section 8.5.4, "The `targetscope` Keyword"
- Section 8.5.5, "LDAP Modify Increment"
- Section 8.5.6, "Macro Support,"
- Section 8.5.7, "The `roledn` Keyword"

### 8.5.1 Global ACI

Global ACI configuration differs from the Oracle Directory Server Enterprise Edition global ACI implementation in two ways:

- The `ds-config-global-aci` attribute specifies a global ACI in the `cn=Access Control Handler,cn=config` entry (see Section 8.1, "Access Control Principles") rather than placing the ACI in the root DSE entry.
- The scope of the global ACI can be narrowed by specifying a target keyword in the ACI. For example, the following global ACI restricts anonymous read access to entries under the suffix `dc=example,dc=com`:

```
ds-cfg-global-aci: (target="dc=example,dc=com")
(targetattr!="userPassword||authPassword")
(version 3.0; acl "Anonymous read access only under dc=example,dc=com suffix";
allow (read,search,compare) userdn="ldap:///anyone";)
```

Removing the `(target="dc=example,dc=com")` expression would make the ACI global to all entries in Oracle Unified Directory.

### 8.5.2 Distinguished Name (DN) Wildcard Matching

The ACI DN wildcard matching implementation supports the following usage:

- Any number of wildcards can appear in Relative Distinguished Name (RDN) attribute values, where they match zero or more characters (similar to substring filters). For example, the bind rule matches the following DNs: `uid=bob jensen,dc=example,dc=com` and `uid=bjensen,dc=example,dc=com`:

  ```
  userdn="ldap:///uid=b*jensen*,dc=example,dc=com"
  ```

  It does not match the DN `cn=bill jensen,dc=example,dc=com` because the attribute type of the first RDN does not match.

- A single wildcard can also be used to match any RDN attribute type. (The wildcard in this case can be omitted as a shorthand). For example, these two bind rules behave exactly the same:

  ```
  userdn="ldap:///*=bjensen, dc=example, dc=com"
  userdn="ldap:///bjensen, dc=example, dc=com"
  ```

  They both match the following DNs: `uid=bjensen,dc=example,dc=com` and `cn=bjensen,dc=example,dc=com`.

- A single wildcard can be used to match exactly one RDN component, which can be single or multivalued). For example, the following bind rule matches the DNs `uid=jensen,dc=example,dc=com` and `cn=smith,dc=example,dc=com`:

  ```
  userdn="ldap:///*,dc=example,dc=com"
  ```

- A double wildcard can be used to match one or more RDN components. For example, the following bind rule matches the DNs `uid=jensen,ou=people,dc=example,dc=com` and `uid=jensen,ou=sales,ou=people,dc=example,dc=com`:

  ```
  userdn="ldap:///uid=bjensen,**,dc=example,dc=com"
  ```

### 8.5.3 Privilege Subsystem Impact

Oracle Directory Server Enterprise Edition has no support for privileges. The privilege subsystem (discussed in Section 23.2, "Root Users and the Privilege Subsystem") impacts ACIs in two ways:

- Users with `ds-privilege-name: bypass-acl` privileges can bypass access control evaluation.
- Users needing to modify access control rules need the `ds-privilege-name: modify-acl` privilege.

---

**Note:** Use of the Lightweight Directory Access Protocol (LDAP) Proxied Authorization Control (https://opends.dev.java.net/public/standards/rfc4370.txt) requires the bind user to have the `ds-privilege-name: proxied-auth` privilege. When the proxied authorization control is used, evaluation of the `ds-privilege-name: bypass-acl` privilege is performed using the bind user, not the proxied user.

In general, a user should not have both the `ds-privilege-name: proxied-auth` and `ds-privilege-name: bypass-acl` privileges simultaneously since this allows a proxied user to bypass ACI access evaluation.

---

### 8.5.4 The `targetscope` Keyword

The `targetscope` keyword differs from Oracle Directory Server Enterprise Edition by including a new scope:

**`subordinate`**
Restricts the ACI to the subtree below the target resource only.

### 8.5.5 LDAP Modify Increment

Oracle Unified Directory supports the LDAP Modify-Increment Extension (https://opends.dev.java.net/public/standards/rfc4525.txt). This extension is not supported in Oracle Directory Server Enterprise Edition. Attributes that are to be incremented must have write permissions.

### 8.5.6 Macro Support

Oracle Unified Directory supports macros in ACIs.

### 8.5.7 The `roledn` Keyword

Roles are not supported in Oracle Unified Directory, so the `roledn` keyword should not be used. Equivalent functionality can be achieved by using groups.

## 8.6 Using Macro ACIs for Advanced Access Control

Organizations that use repeating directory tree structures can enhance the performance and ACI memory usage by using macros to optimize the number of ACIs in the directory tree. When you reduce the number of ACIs in your directory tree, it is easier to manage your access control policy.

This section describes macro ACIs and its usage, and contains the following topics:

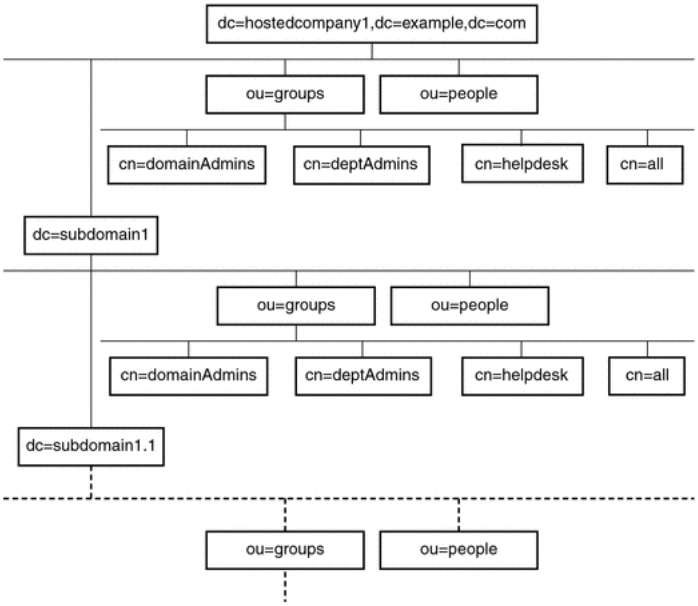
## 8.6.1 What are Macros?

Macros are placeholders used to represent a DN or a part of a DN in an ACI. You can use a macro to represent a DN in the target section of the ACI, in the bind rule section, or in both. In practice, when Directory Server receives an incoming LDAP operation, the ACI macros are matched against the resource targeted by the LDAP operation. The matching occurs in order to determine a matching substring, if it exists. If a match exists, the bind rule-side macro is expanded using the matched substring, and access to the resource is determined by evaluating that expanded bind rule.

## 8.6.2 Macro ACI Example

The advantage of using macro ACIs and how they work are best explained through an example. Figure 8–1 shows a directory tree that uses macro ACIs to effectively reduce the total number of ACIs.

***Figure 8–1 Example Directory Tree for Macro ACIs***




This illustration uses repeating pattern of subdomains with the same tree structure (`ou=groups, ou=people`). This pattern is also repeated across the tree because the `example.com` directory tree stores the suffixes `dc=hostedCompany2, dc=example,dc=com` and `dc=hostedCompany3,dc=example,dc=com` not shown in the preceding graphic.

The ACIs that apply in the directory tree also have a repeating pattern. For example, the following ACI is located on the `dc=hostedCompany1,dc=example,dc=com` node:

```
aci: (targetattr="*")(targetfilter=(objectClass=nsManagedDomain))
 (version 3.0; acl "Domain access"; allow (read,search)
 groupdn="ldap:///cn=DomainAdmins,ou=Groups,dc=hostedCompany1,dc=example,dc=com";)
```

This ACI grants read and search rights to the DomainAdmins group to any entry in the dc=hostedCompany1,dc=example,dc=com tree.

The following ACI is located on the dc=hostedCompany1,dc=example,dc=com node:

```
aci: (targetattr="*")
 (targetfilter=(objectClass=nsManagedDomain))
 (version 3.0; acl "Domain access"; allow (read,search)
 groupdn="ldap:///cn=DomainAdmins,ou=Groups,dc=hostedCompany1,dc=example,dc=com";)
```

The following ACI is located on the dc=subdomain1, dc=hostedCompany1, dc=example, dc=com node:

```
aci: (targetattr="*")
 (targetfilter=(objectClass=nsManagedDomain))
 (version 3.0; acl "Domain access"; allow (read,search)
 groupdn="ldap:///cn=DomainAdmins,ou=Groups,dc=subdomain1,dc=hostedCompany1,
 dc=example,dc=com";)
```

The following ACI is located on the dc=hostedCompany2, dc=example, dc=com node:

```
aci: (targetattr="*")
 (targetfilter=(objectClass=nsManagedDomain))
 (version 3.0; acl "Domain access"; allow (read,search)
 groupdn="ldap:///cn=DomainAdmins,ou=Groups,dc=hostedCompany2,
dc=example,dc=com";)
```

The following ACI is located on the dc=subdomain1, dc=hostedCompany2, dc=example, dc=com node:

```
aci: (targetattr="*")
 (targetfilter=(objectClass=nsManagedDomain))
 (version 3.0; acl "Domain access"; allow (read,search)
 groupdn="ldap:///cn=DomainAdmins,ou=Groups,dc=subdomain1,dc=hostedCompany2,
 dc=example,dc=com";)
```

In the preceding four ACIs, the only difference is the DN that is specified in the groupdn keyword. By using a macro for the DN, it is possible to replace these ACIs with a single ACI at the root of the tree on the dc=example,dc=com node. This macro ACI reads as follows:

```
aci: (target="ldap:///ou=Groups,($dn),dc=example,dc=com")
 (targetattr="*")(targetfilter=(objectClass=nsManagedDomain))
 (version 3.0; acl "Domain access"; allow (read,search)
 groupdn="ldap:///cn=DomainAdmins,ou=Groups,[$dn],dc=example,dc=com";)
```

The target keyword, which was not previously used, is utilized in the new ACI.

In this example, the number of ACIs is reduced from four to one. The real determining factor is the number of repeating patterns you have down and across your directory tree.

## 8.6.3 Macro ACI Syntax

Macro ACIs include the following types of expressions to replace a DN or part of a DN:

- ($dn)
- [$dn]
- ($attr.*attrName*), where *attrName* represents an attribute contained in the target entry

In this section, the ACI keywords used to provide bind credentials, such as userdn, roledn, groupdn, and userattr are collectively called the subject of the ACI. The subject determines to whom the ACI applies.

Table 8–2 lists the macros that can be used to replace specific ACI keywords.

***Table 8–2 Macro ACI Keywords***

| Macro | Description | ACI Keywords |
|---|---|---|
| `($dn)` | For matching in the target, and direct substitution in the subject. For example, it will match either target or targetfilter and substitute the matched value into userdn, groupdn, or userattr. | (target, targetfilter) and (userdn, groupdn, userattr) |
| `[$dn]` | For substituting multiple RDNs that work in subtrees of the subject. | (targetfilter) and (userdn, groupdn, userattr) |
| `($attr.`*`attrName`*`)` | For substituting the value of the *`attributeName`* attribute from the target entry into the subject. | userdn, groupdn, userattr |

The following restrictions apply to macro ACI keywords:

- If you use `($dn)` macro in a subject, then you must define a target that contains `($dn)`.
- If you use `[$dn]` macro in a subject, then you must define a target that contains `($dn)`.
- You can combine both the `($dn)` macro and the `[$dn]` macro with the `($attr.`*`attrName`*`)` macro in a subject.

The following sections describe the evaluation mechanism for macro ACIs, and contains the following topics:

- Section 8.6.3.1, "Matching for `($dn)` in the Target"
- Section 8.6.3.2, "Macro Matching for ($attr.attrName)"

### 8.6.3.1 Matching for `($dn)` in the Target

The `($dn)` macro in the target of an ACI determines the substitution value by comparing it to the entry targeted by the LDAP request. For example, you have an LDAP request targeted at this entry:

```
cn=all, ou=groups, dc=subdomain1, dc=hostedCompany1, dc=example, dc=com
```

In addition, you have an ACI that defines the target as follows:

```
(target="ldap:///ou=Groups,($dn),dc=example,dc=com")
```

The `($dn)` macro matches with `"dc=subdomain1, dc=hostedCompany1"`. This substring is then used for substitutions in the subject of the ACI.

#### Substituting ($dn) in the Subject

In the subject of the ACI, the `($dn)` macro is replaced by the entire substring that matches in the target. For example:

```
groupdn="ldap:///cn=DomainAdmins,ou=Groups,($dn),dc=example,dc=com"
```

In this scenario, if the string matching `($dn)` in the target is `dc=subdomain1, dc=hostedCompany1,` then the same string is used in the subject. The subject is then expanded as follows:

```
groupdn="ldap:///cn=DomainAdmins,ou=Groups,dc=subdomain1,dc=hostedCompany1,dc=exam
ple,dc=com"
```

In the targetfilter of the ACI, the `($dn)` macro is replaced with the entire substring that matches in the target. For example:

```
(targetattr="*")(targetfilter=(&(objectClass=nsManagedPerson)(!(memberOf=cn=Servic
eAdministrators,ou=Groups,($dn),o=ace industry,c=us))(!(memberOf=cn=Service Help
Desk Administrators,ou=Groups,($dn),o=ace industry,c=us))))
```

The targetfilter becomes:

```
(targetattr="*")(targetfilter=(&(objectClass=nsManagedPerson)
(!(memberOf=cn=ServiceAdministrators,ou=Groups,dc=subdomain1,
dc=hostedCompany1,o=ace industry,c=us))
(!(memberOf=cn=Service Help Desk
Administrators,ou=Groups,dc=subdomain1,dc=hostedCompany1,o=ace industry,c=us))))
```

After the macro has been expanded, Directory Server evaluates the ACI following the normal process to determine whether access is granted.

> **Note:** Unlike a standard ACI, an ACI that uses macro substitution does not necessarily grant access to the child of the targeted entry. This is because when the child DN is the target, the substitution might not create a valid DN in the subject string.

### Substituting [$dn] in the Subject

The substitution mechanism for `[$dn]` is slightly different than for `($dn)`. The DN of the targeted resource is examined several times, each time dropping the left-most RDN component, until a match is found.

Consider a scenario in which you have an LDAP request targeted at the `cn=all,ou=groups, dc=subdomain1,dc=hostedCompany1,dc=example,dc=com` subtree, and the following ACI:

```
aci: (targetattr="*") (target="ldap:///ou=Groups,($dn),dc=example,dc=com")
(version 3.0; acl "Domain access"; allow (read,search)
groupdn="ldap:/cn=DomainAdmins,ou=Groups,[$dn], dc=example,dc=com";)
```

The server proceeds as follows to expand this ACI:

1. The server verifies that the `($dn)` in target matches `dc=subdomain1, dc=hostedCompany1.`
2. The server replaces `[$dn]` in the subject with `dc=subdomain1, dc=hostedCompany1.`

   The resulting subject is `groupdn="ldap:///cn=DomainAdmins,ou=Groups, dc=subdomain1,dc=hostedCompany1,dc=example,dc=com".` If access is granted, because bind DN is a member of that group, macro expansion stops, and the ACI is evaluated. If bind DN is not a member, the process continues.
3. The server replaces `[$dn]` in the subject with `dc=hostedCompany1.`

   The resulting subject is `groupdn="ldap:///cn=DomainAdmins,ou=Groups, dc=hostedCompany1,dc=example,dc=com".` Bind DN is again tested for being a member of this group and if it is, the ACI is evaluated fully. However, if bind DN is not a member, macro expansion stops with the last RDN of the matched value, and ACI evaluation is finished for this ACI.

The advantage of the [$dn] macro is that it provides a flexible mechanism to grant domain-level administrators access to all the subdomains in the directory tree. Therefore, the [$dn] macro is useful for expressing a hierarchical relationship between domains.

For example, consider the following ACI:

```
aci: (target="ldap:///ou=*,($dn),dc=example,dc=com") (targetattr="*")
(targetfilter=(objectClass=nsManagedDomain)) (version 3.0; acl "Domain access";
allow (read,search) groupdn=
"ldap:/cn=DomainAdmins,ou=Groups,[$dn],dc=example,dc=com";)
```

The ACI grants access to the members of cn=DomainAdmins,ou=Groups, dc=hostedCompany1,dc=example,dc=com to all of the subdomains under dc=hostedCompany1. Thus, an administrator who belongs to that group could access, for example, the subtree ou=people,dc=subdomain1.1,dc=subdomain1.

However, at the same time, members of cn=DomainAdmins,ou=Groups, dc=subdomain1.1 would be denied access to the ou=people,dc=subdomain1, dc=hostedCompany1 and ou=people,dc=hostedCompany1 nodes.

#### 8.6.3.2 Macro Matching for ($attr.attrName)

The ($attr.attrname) macro is always used in the subject part of an ACI. For example, you could define the following groupdn:

```
groupdn =
"ldap:/cn=DomainAdmins,ou=($attr.ou),dc=HostedCompany1,dc=example,dc=com"
```

Now, assume that the server receives an LDAP operation that is targeted at the following entry:

```
dn: cn=Babs Jensen,ou=People,dc=HostedCompany1,dc=example,dc=com
cn: Babs Jensen
sn: Jensen
ou: Sales ...
```

To evaluate the groupdn part of the ACI, the server reads the value of the ou attribute stored in the targeted entry. The server then substitutes this value in the subject to expand the macro. In this example, the groupdn is expanded as follows:

```
groupdn= "ldap:///cn=DomainAdmins,ou=Sales,dc=HostedCompany1,dc=example,dc=com"
```

Directory Server then evaluates the ACI according to the normal ACI evaluation algorithm.

When the attribute that is named in the macro is multivalued, each value is used in turn to expand the macro. The first value that provides a successful match is used.

# 9

# Understanding the Oracle Unified Directory Schema Model

This chapter describes schema elements in general and illustrates the ways that these schema elements are used in Oracle Unified Directory.

The chapter covers the following topics:

- Section 9.1, "Understanding Matching Rules"
- Section 9.2, "Understanding Attribute Syntaxes"
- Section 9.3, "Understanding Attribute Types"
- Section 9.4, "Understanding Object Classes"
- Section 9.5, "Understanding Name Forms"
- Section 9.6, "Understanding DIT Content Rules"
- Section 9.7, "Understanding DIT Structure Rules"
- Section 9.8, "Understanding Matching Rule Uses"

For instructions on viewing the schema using the `ldapsearch` command, see Section 27.4.1, "Managing Attribute Types" and Section 27.4.2, "Managing Object Classes."

## 9.1 Understanding Matching Rules

Matching rules are used by Oracle Unified Directory to compare two values for the same attribute, that is, to perform matching operations on them. There are several different types of matching rules, including:

**Equality matching rules**
These matching rules are used to determine whether two values are logically equal to each other. Different implementations of equality matching rules can use different criteria for making this determination (for example, whether to ignore differences in capitalization or deciding which spaces are significant).

**Ordering matching rules**
These matching rules are used to determine the relative order for two values, for example, when evaluating greater-or-equal or less-or-equal searches, or when the results need to be sorted.

**Substring matching rules**
These matching rules are used to determine whether a given substring assertion matches a particular value. A substring assertion is composed of at least one element

from the following sets: at most one `subInitial` element, zero or more `subAny` elements, and at most one `subFinal` element.

**Approximate matching rules**
These matching rules are used to determine whether two values are approximately equal to each other. This is frequently based on "sounds like" or some other kind of fuzzy algorithm. Approximate matching rules are not part of the official LDAP specification, but they are included in Oracle Unified Directory for added flexibility.

The following sections describe matching rules:

- Section 9.1.1, "Matching Rule Description Format"
- Section 9.1.2, "Commonly Used Matching Rules"
- Section 9.1.3, "Relative Time Matching Rules"
- Section 9.1.4, "Partial Date Or Time Matching Rules"
- Section 9.1.5, "Value Normalization"

## 9.1.1 Matching Rule Description Format

The matching rule description format is described in RFC 4512 (`http://www.ietf.org/rfc/rfc4512.txt`), section 4.1.3, in Augmented Backus-Naur Form (ABNF). For more information about ABNF, see RFC 4234 (`http://www.ietf.org/rfc/rfc4234.txt`) and RFC 5234 (`http://www.ietf.org/rfc/rfc5234.txt`). This is the format that is used to display matching rules in the `matchingRules` attribute of the schema subentry, and it shows the properties that can be associated with a matching rule. The following example shows the definition of the matching rule description format:

```
MatchingRuleDescription = LPAREN WSP
numericoid                 ; object identifier
[ SP "NAME" SP qdescrs ]   ; short names (descriptors)
[ SP "DESC" SP qdstring ]  ; description
[ SP "OBSOLETE" ]          ; not active
SP "SYNTAX" SP numericoid  ; assertion syntax
extensions WSP RPAREN      ; extensions
```

The matching rule description includes these elements:

`numericoid`
The numeric OID is used to uniquely identify the matching rule in Oracle Unified Directory. Every matching rule must have a unique OID.

`NAME`
The name elements are human-readable names assigned to the matching rule that can be used to refer to it in place of the OID. A matching rule is not required to have any human-readable names. If it has only a single name, then it is enclosed in single quotes. If there are multiple names for a matching rule, each is enclosed in single quotes with spaces between the names, and parentheses around the entire set of names.

`DESC`
The description element is a human-readable description for the matching rule. There can be at most one description, and if it is present, it should be enclosed in single quotation marks.

OBSOLETE
The OBSOLETE flag indicates whether this matching rule should be considered available for use. If a matching rule is marked OBSOLETE, then it should not be possible to create any new attribute types or matching rule uses that reference this matching rule.

SYNTAX
The syntax element identifies the attribute syntax with which the matching rule is associated. This is used to indicate the acceptable format for values on which the matching rule operates. More information about attribute syntaxes can be found in Section 9.2, "Understanding Attribute Syntaxes." The syntax OID must be included in all matching rule descriptions.

***extensions***
The extensions for a matching rule can be used to identify other properties for that matching rule that might not be included in the standard definition. Oracle Unified Directory does not currently support any extensions for use in matching rules.

For example, the following is the matching rule description for the standard caseIgnoreMatch matching rule:

```
( 2.5.13.2 NAME 'caseIgnoreMatch' SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

In this case, the OID is 2.5.13.2. There is one name, which is caseIgnoreMatch. There is no description. The OID of the associated syntax is 1.3.6.1.4.1.1466.115.121.1.15 (which is the Directory String syntax). There are no extensions.

## 9.1.2 Commonly Used Matching Rules

There are a number of matching rules defined in LDAP, both in the core protocol specification as well as in other related RFCs and Internet Drafts. Many of these matching rules are defined in RFC 4517 (http://www.ietf.org/rfc/rfc4517.txt) (LDAP Syntaxes and Matching Rules), in section 4.2. Some of the most commonly used matching rules include:

caseIgnoreMatch, caseIgnoreOrderingMatch, caseIgnoreSubstringsMatch
These are equality, ordering, and substring matching rules, respectively, that ignore differences in capitalization and also treat multiple consecutive spaces as a single space.

caseExactMatch, caseExactOrderingMatch, caseExactSubstringsMatch
These are equality, ordering, and substring matching rules, respectively, that treat values in a case-sensitive manner but do treat multiple consecutive spaces as a single space.

octetStringMatch, octetStringOrderingMatch, octetStringSubstringsMatch
These are equality, ordering, and substring matching rules, respectively, that perform byte-for-byte comparisons of the values, treating them as binary data rather than strings.

numericStringMatch, numericStringOrderingMatch, numericStringSubstringsMatch
These are equality, ordering, and substring matching rules, respectively, that operate on values that start with a numeric digit, and contain only numeric digits and spaces. Spaces are ignored when performing matching with these matching rules.

`distinguishedNameMatch`
This is an equality matching rule that operates on distinguished name (DN) values. It ignores spaces around the commas or semicolons that separate DN components, spaces around plus signs that separate RDN components, and spaces around equal signs that separate RDN attribute type names from their corresponding values. Differences in capitalization are ignored for attribute type names. Equality matching for attribute values is performed using the equality matching rule for the corresponding attribute type.

`doubleMetaphoneApproximateMatch`
This is an approximate matching rule that uses the double metaphone algorithm to perform a "sounds like" comparison. Note that this matching rule is not part of any official LDAP specification, but it is included in Oracle Unified Directory for added flexibility.

### 9.1.3 Relative Time Matching Rules

Oracle Unified Directory provides two matching rules for performing a match on relative dates in Appendix D.7.1, "generalized time" attributes, `relativeTimeLTOrderingMatch` and `relativeTimeGTOrderingMatch`, as defined here:

```
( 1.3.6.1.4.1.26027.1.4.6
NAME ( 'relativeTimeLTOrderingMatch''relativeTimeOrderingMatch.lt' )
SYNTAX 1.3.6.1.4.1.1466.115.121.1.24 )

( 1.3.6.1.4.1.26027.1.4.5
NAME ( 'relativeTimeGTOrderingMatch' 'relativeTimeOrderingMatch.gt' )
SYNTAX 1.3.6.1.4.1.1466.115.121.1.24 )
```

The syntax applies to attributes with a `GeneralizedTime` syntax, but it does not take a generalized time string. Instead it takes an offset in the format of [+ | –]*number*[*unit*] where:

+|–
Specifies a time in the past or future. A positive offset (+) computes a time in the future compared to the current time, and a negative offset (–) computes a time in the past compared to the current time. The default value is positive (+).

***number***
Specifies the number of time units as a positive integer

***unit***
Specifies the time unit as a single letter, `s`, `m`, `h`, `d`, or `w`, for seconds, minutes, hours, days, or weeks

When processing the filter, the server computes the current GMT time, adds the offset and compares the attribute value with the new computed value.

The following example represents `pwdExpirationTime >= (Now + 5 days).`

```
(pwdExpirationTime:1.3.6.1.4.1.26027.1.4.5:=5d)
```

Similarly, the following example represents `pwdExpirationTime <= (Now + 5 days).`

```
(pwdExpirationTime:1.3.6.1.4.1.26027.1.4.6:=5d)
```

### 9.1.4 Partial Date Or Time Matching Rules

Oracle Unified Directory provides the `partialDateAndTimeMatchingRule` matching rule for performing a substring match on dates in Appendix D.7.1, "generalized time" attributes:

```
( 1.3.6.1.4.1.26027.1.4.7
NAME 'partialDateAndTimeMatchingRule'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.24 )
```

This matching rule applies to attributes with a `GeneralizedTime` syntax, but the value is not a generalized time. Instead, it specifies a pattern for the date, composed of one or more sequences of an integer followed by a tag. The currently supported tags are `Y`, `M`, `D`, `h`, `m`, and `s`.

The following examples use the attribute `birthDate` (described in `http://tools.ietf.org/html/draft-gryphon-ldap-schema-vcard4-00`) with the following definition:

```
attributeTypes: ( 1.3.6.1.4.1.33592.1.3.2 NAME 'birthDate'
DESC 'birthday'
EQUALITY generalizedTimeMatch
ORDERING generalizedTimeOrderingMatch
SYNTAX .3.6.1.4.1.1466.115.121.1.24
USAGE userApplications SINGLE-VALUE )
```

For example, the following filter matches all users born on September 21st.

```
(birthDate:1.3.6.1.4.1.26027.1.4.7:=09M21D)
```

As another example, the following filter matches all users born in 1965:

```
(birthDate:1.3.6.1.4.1.26027.1.4.7:=1965Y)
```

The following search operation returns all entries with a birthday the fourteenth day of any month:

```
$ ./ldapsearch -p 1389 -h localhost -D "cn=Directory Manager" -j pwd-file \
  -b "dc=example,dc=com" \
  "(birthDate:1.3.6.1.4.1.26027.1.4.7:=14D)" birthDate
```

### 9.1.5 Value Normalization

One of the tasks that most matching rules need to perform is value normalization. This is the process of transforming a given value to a form that can be used to compare values efficiently. In most cases, the normalization process should reduce all logically equivalent values to the same string so that a very simple string comparison can be performed to determine whether the strings are equal. For example, the `caseIgnoreMatch` matching rule typically normalizes values by converting all characters to lowercase and replacing occurrences of multiple consecutive spaces with a single space. A more complicated example is the `distinguishedNameMatch` matching rule, which removes all unnecessary spaces (for example, around commas, equal signs, and plus signs), converts all attribute types to lowercase, and then uses the appropriate matching rules to normalize the attribute values for each RDN component.

Note that in some cases, normalization alone is not sufficient for determining whether two values are logically equivalent. This is particularly true for cases in which the value is transformed, and there can be multiple different transformations for the same value.

## 9.2 Understanding Attribute Syntaxes

Attribute syntaxes are essentially data type definitions. The syntax for an attribute type indicates the type of data meant to be held by the corresponding values. This can be used to determine whether a particular value is acceptable for a given attribute, as well as to provide information about how Oracle Unified Directory should interact with existing values.

Oracle Unified Directory supports the ability to reject values that violate the associated attribute syntax, and this is the default behavior for the purposes of standards compliance. It is possible to disable this attribute syntax checking completely if necessary, but it is also possible to accept values that violate the associated syntax but log a warning message to Oracle Unified Directory's error log every time this occurs. However, if attributes are allowed to have values that violate their associated syntax, matching operations might not behave as expected with such values. For information about disabling schema checking, see Section 27.2, "Configuring Schema Checking."

The following sections discuss attribute syntax:

- Section 9.2.1, "The Attribute Syntax Description Format"
- Section 9.2.2, "Commonly Used Attribute Syntaxes"
- Section 9.2.3, "The Pattern-Matching Syntax Extension"
- Section 9.2.4, "The Enumeration Syntax Extension"
- Section 9.2.5, "Substitution Syntax Extension"

### 9.2.1 The Attribute Syntax Description Format

The attribute syntax description format is described in RFC 4512 (`http://www.ietf.org/rfc/rfc4512.txt`), section 4.1.5, as shown in this example:

```
SyntaxDescription = LPAREN WSP
numericoid                 ; object identifier
[ SP "DESC" SP qdstring ]  ; description
extensions WSP RPAREN      ; extensions
```

The attribute syntax description includes these elements:

`numericoid`
The numeric OID used to uniquely identify the attribute syntax in Oracle Unified Directory.

`DESC`
An optional description for the syntax. If it is provided, then it must be enclosed in single quotation marks.

***extensions***
An optional set of extensions for the attribute syntax. Oracle Unified Directory supports the following extensions:

- `X_PATTERN`: Specifies that the attribute uses the regular expression syntax. See Section 9.2.3, "The Pattern-Matching Syntax Extension" for more information.
- `X-ENUM`: Specifies that the attribute uses the enumerated syntax. See Section 9.2.4, "The Enumeration Syntax Extension" for more information.
- `X-SUBST`: Specifies that the attribute uses the substitution syntax. See Section 9.2.5, "Substitution Syntax Extension" for more information.

The following example shows the attribute syntax description for the standard directory string syntax:

```
( 1.3.6.1.4.1.1466.115.121.1.15 DESC 'Directory String' )
```

In this case, the OID is `1.3.6.1.4.1.1466.115.121.1.15`, and the description is `Directory String`. This example specifies no extensions.

## 9.2.2 Commonly Used Attribute Syntaxes

There are a number of attribute syntaxes defined in LDAP, both in the core protocol specification and in other related RFCs and Internet Drafts. Many of these attribute syntaxes are defined in RFC 4517 (http://www.ietf.org/rfc/rfc4517.txt) (LDAP Syntaxes and Matching Rules) in section 3.3. Some of the most commonly used attribute syntaxes include:

**Directory String**
The Directory String syntax is used to hold general-purpose string values containing one or more UTF-8 characters. Technically, empty values (that is, those with zero characters) are not allowed. Because Oracle Directory Server Enterprise Edition has historically allowed empty values, Oracle Unified Directory offers a configuration option that can be used to allow it as well although it is disabled by default for standards compliance.

**IA5 String**
The IA5 String syntax is used to hold string values based on the IA5 character set, which is also known as the ASCII character set.

**Printable String**
The Printable String syntax is used to hold string values that contain one or more characters from the set of uppercase and lowercase letters, numeric digits, single quotes, left and right parentheses, plus sign, comma, hyphen, period, and equal sign.

**Boolean**
The Boolean syntax is used to hold values of either `TRUE` or `FALSE`. No other values are allowed for attributes with this syntax.

**Integer**
The Integer syntax is used to hold integer values, which must contain at least one digit. It can start with a hyphen to indicate a negative value. Zero can be used as the first digit only when the value is zero.

**Octet String**
The Octet String syntax is used to hold a set of zero or more bytes. It has been used to replace the former Binary syntax.

**DN**
The DN syntax is used to hold distinguished name values, comprised of zero or more RDN components. Values should be in the format specified in RFC 4514 (http://www.ietf.org/rfc/rfc4514.txt) (LDAP String Representation of Distinguished Names).

## 9.2.3 The Pattern-Matching Syntax Extension

The `X-PATTERN` attribute syntax extension can be used to define new string syntaxes with values restricted by one or more regular expressions. The following example adds an `X-PATTERN` attribute syntax to the schema.

```
$ ldapmodify -p 1389 -h localhost -D "cn=Directory Manager" -j pwd-file
dn: cn=schema
changetype: modify
add: ldapsyntaxes
ldapSyntaxes: ( 1.3.6.1.4.1.32473.1 DESC 'Host and Port in the format of
HOST:PORT'
  X-PATTERN '^[a-zA-Z][a-zA-Z0-9-]+:[0-9]+$' )
```

This new syntax can be used to define attributes and object classes, as shown in the following example.

```
$ ldapmodify -p 1389 -h localhost -D "cn=Directory Manager" -j pwd-file
dn: cn=schema
changetype: modify
add: attributetypes
attributetypes: ( 1.3.6.1.4.1.32473.2 NAME 'example-attr-regex' SYNTAX
1.3.6.1.4.1.32473.1 )
-
add: objectclasses
objectclasses: ( 1.3.6.1.4.1.32473.3 NAME 'exampleOCregex' SUP top AUXILIARY MUST
example-attr-regex)
-
```

Values for `example-attr-regex` attributes must match the defined pattern, or the server rejects them. The following attribute fits pattern defined in the example syntax, so the server accepts it:

```
example-attr-regex: localhost:389
```

The following attribute is rejected because it does not include the required colon and numeric string:

```
localhost
```

The following attribute is rejected because it contains periods (.), which are not specified as part of the `HOST` component:

```
host.domain.com:389
```

### 9.2.4 The Enumeration Syntax Extension

The `X-ENUM` attribute syntax extension can be used to define new string syntaxes with values restricted to a set of defined, ordered values. The following example defines an `X-ENUM` attribute to the schema.

```
$ ldapmodify -p 1389 -h localhost -D "cn=Directory Manager" -j pwd-file
dn: cn=schema
changetype: modify
add: ldapsyntaxes
ldapSyntaxes: ( 1.3.6.1.4.1.32473.4 DESC 'Day Of The Week'
  X-ENUM ( 'monday' 'tuesday' 'wednesday' 'thursday'
  'friday' 'saturday' 'sunday' ) )
```

This new syntax can be used to define attributes and object classes, as shown in the following example.

```
$ ldapmodify -p 1389 -h localhost -D "cn=Directory Manager" -j pwd-file
dn: cn=schema
changetype: modify
add: attributetypes
```

```
attributetypes: ( 1.3.6.1.4.1.32473.5 NAME 'example-attr-enum' SYNTAX
1.3.6.1.4.1.32473.4 )
-
add: objectclasses
objectclasses: ( 1.3.6.1.4.1.32473.6 NAME 'exampleOCenum' SUP top AUXILIARY
MUST example-attr-enum)
```

Values for `example-attr-enum` attributes must match the defined pattern, or the server rejects them.

Enumerated values are not case-sensitive, so both of the following examples are accepted:

```
example-attr-enum: thursday
example-attr-enum: Thursday
```

Enumerated attribute values are literal (and not internationalized), so the following example does not match the pattern and is rejected, regardless of any semantic equivalence:

```
example-attr-enum: jeudi
```

The defined values specify an order, so enumerated attributes can be used in relative comparison filters, as shown in the following example:

```
(example-attr-enum>=wednesday)
```

The preceding comparison filter matches, for example, a value of `thursday`. The comparison is based on the order of the enumerated values, and ASCII values are not applicable in this case.

### 9.2.5 Substitution Syntax Extension

The `X-SUBST` attribute syntax extension can be used to define new string syntaxes with values in terms of existing syntaxes. It is provided for use when extending the native directory server schema with a non-standard schema (or an external schema) that uses syntaxes not supported by Oracle Unified Directory. Instead of altering the imported schema, extend it with the `X-SUBST` extension to instruct Oracle Unified Directory to treat values in terms of a supported syntax.

The following example defines a new syntax, `AttCertPath`, in terms of an existing syntax, `1.3.6.1.4.1.1466.115.121.1.15`, directory string. This change must be made under `cn=schema`.

```
$ ldapmodify -p 1389 -h localhost -D "cn=Directory Manager" -j pwd-file
dn: cn=schema
objectClass: top
objectClass: ldapSubentry
objectClass: subschema
ldapSyntaxes: ( 1.3.6.1.4.1.4203.666.11.10.2.4
  DESC 'AttCertPath'
  X-SUBST '1.3.6.1.4.1.1466.115.121.1.15' )
```

This feature can be useful during migration and can lessen the impact on the schema. For example, during migration to Oracle Unified Directory, an incoming schema can contain attribute definitions that use an undefined syntax. The `X-SUBST` attribute syntax extension provides a means to define those missing syntaxes in terms of other, more general syntaxes. With this capability, the schema and data can be migrated without the need to modify the schema or data or to implement new syntaxes.

## 9.3 Understanding Attribute Types

Attribute types define the set of attributes that can be used in Oracle Unified Directory and how operations involving those attributes should be conducted. Among other things, it combines an attribute syntax and set of matching rules with a unique OID and human-readable names.

The following sections describe attribute types:

- Section 9.3.1, "Attribute Type Description Format"
- Section 9.3.2, "Attribute Type Inheritance"
- Section 9.3.3, "Attribute Type Implementation"

### 9.3.1 Attribute Type Description Format

The attribute type description format is described in RFC 4512 (http://www.ietf.org/rfc/rfc4512.txt), section 4.1.2 as shown here:

```
AttributeTypeDescription = LPAREN WSP
    numericoid                    ; object identifier
    [ SP "NAME" SP qdescrs ]      ; short names (descriptors)
    [ SP "DESC" SP qdstring ]     ; description
    [ SP "OBSOLETE" ]             ; not active
    [ SP "SUP" SP oid ]           ; supertype
    [ SP "EQUALITY" SP oid ]      ; equality matching rule
    [ SP "ORDERING" SP oid ]      ; ordering matching rule
    [ SP "SUBSTR" SP oid ]        ; substrings matching rule
    [ SP "SYNTAX" SP noidlen ]    ; value syntax
    [ SP "SINGLE-VALUE" ]         ; single-value
    [ SP "COLLECTIVE" ]           ; collective
    [ SP "NO-USER-MODIFICATION" ] ; not user modifiable
    [ SP "USAGE" SP usage ]       ; usage
    extensions WSP RPAREN         ; extensions

usage = "userApplications"     /  ; user
        "directoryOperation"   /  ; directory operational
        "distributedOperation" /  ; DSA-shared operational
        "dSAOperation"            ; DSA-specific operational
```

The attribute type description includes these elements:

`numericoid`
The numeric OID used to uniquely identify the attribute type in Oracle Unified Directory. Although the specification requires a numeric OID, Oracle Unified Directory also allows a non-numeric OID for the purpose of convenience and better compatibility with Oracle Directory Server Enterprise Edition. In this case, the non-numeric OID should be the same as the name of the attribute type followed by the string `-oid`.

`NAME`
An optional set of human-readable names that can also be used to refer to the attribute type. If there is a single name, then it should be enclosed in single quotes. If there are multiple names, then they should each be enclosed in single quotes separated by spaces, and the entire set of names should be enclosed in parentheses.

`DESC`
An optional human-readable description. If there is a description, then it should be enclosed in single quotation marks.

OBSOLETE
An optional OBSOLETE flag that can be used to indicate whether the attribute type is active. If an attribute type is marked as OBSOLETE, then it means that it should not be referenced by any new elements created in Oracle Unified Directory.

SUP
An optional reference to the superior attribute type. If there is a superior type, then it may be referenced by either its OID or any of its human-readable names.

EQUALITY
An optional equality matching rule definition. If a specific equality matching rule is provided, then it can be referenced by either its OID or any of its human-readable names. If no equality matching rule is given, then the attribute type uses the default equality matching rule for the associated attribute syntax. If the attribute syntax does not have a default equality matching rule, then equality matching operations are not allowed for attributes of that type.

ORDERING
An optional ordering matching rule definition. If a specific ordering matching rule is provided, then it can be referenced by either its OID or any of its human-readable names. If no ordering matching rule is given, then the attribute type uses the default ordering matching rule for the associated attribute syntax. If the attribute syntax does not have a default ordering matching rule, then ordering matching operations are not allowed for attributes of that type.

SUBSTR
An optional substring matching rule definition. If a specific substring matching rule is provided, then it can be referenced by either its OID or any of its human-readable names. If no substring matching rule is given, then the attribute type uses the default substring matching rule for the associated attribute syntax. If the attribute syntax does not have a default substring matching rule, then substring matching operations are not allowed for attributes of that type.

SYNTAX
An optional attribute syntax for use with the attribute type. If it is provided, then it should be given as a numeric OID. The syntax identifier can also optionally contain an integer value enclosed in curly braces directly following the OID (without any spaces between the last digit of the OID and the opening curly brace), which may be used to suggest a minimum upper bound on the length of values for attributes of that type. Oracle Unified Directory does not enforce any maximum length restrictions for attribute values, so if a length is given, then it is ignored.

SINGLE-VALUE
An optional SINGLE-VALUE flag that indicates that attributes of that type are allowed to have only a single value in any entry in which they appear. If this flag is not present in the attribute type description, then attributes of that type are allowed to have multiple distinct values in the same entry.

COLLECTIVE
An optional COLLECTIVE flag that indicates that the attributes of that type are assigned their values by virtue in their membership in some collection. Collective attributes are described in RFC 3671 (http://www.ietf.org/rfc/rfc3671.txt) (Collective Attributes in LDAP) and are one of the types of virtual attributes that are supported in Oracle Unified Directory.

NO-USER-MODIFICATION
An optional NO-USER-MODIFICATION flag that indicates that values of attributes of that type cannot be modified by external clients (that is, the values can be modified only by internal processing within Oracle Unified Directory).

USAGE
An optional usage specification that indicates how the attribute type is to be used. The following attribute usages are allowed:

- userApplications — Used to store user data.
- directoryOperation — Used to store data required for internal processing within Oracle Unified Directory.
- distributedOperation — Used to store operational data that must be synchronized across servers in the topology.
- dSAOperation — Used to store operational data that is specific to a particular directory server and should not be synchronized across the topology.

***extensions***
An optional set of extensions for the attribute type. Oracle Unified Directory currently uses the following extensions for attribute types:

- X-ORIGIN — Provides information about where the attribute type is defined (for example, whether it is defined by a particular RFC or Internet Draft or whether it is defined within the project).
- X-SCHEMA-FILE — Indicates which schema file contains the attribute type definition.
- X-APPROX — Indicates which approximate matching rule should be used for the attribute type. If this is specified, then its value should be the name or OID of a registered approximate matching rule.

For example, the following is the attribute type description for the standard uid attribute type:

```
( 0.9.2342.19200300.100.1.1 NAME 'uid' EQUALITY caseIgnoreMatch
SUBSTR caseIgnoreSubstringsMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{256}
X-ORIGIN 'RFC 4519' )
```

In this case, the OID is 0.9.2342.19200300.100.1.1. There is a single human-readable name of uid. The caseIgnoreMatch rule should be used for equality matching, and the caseIgnoreSubstringsMatch rule should be used for substring matching. The attribute type uses the directory string syntax with a suggested minimum upper bound of 256 characters, and the attribute type definition was taken from RFC 4519 (http://www.ietf.org/rfc/rfc4519.txt). There is no description or superior type specified. The attribute type is not marked OBSOLETE, SINGLE-VALUE, COLLECTIVE, or NO-USER-MODIFICATION. There is no ordering matching rule specified, which means that Oracle Unified Directory falls back on the default ordering rule used by the directory string syntax. There is no X-APPROX extension to specify the approximate matching rule so the default approximate rule for the directory string syntax is used there as well.

## 9.3.2 Attribute Type Inheritance

One attribute type can reference another as its superior type. This has two primary effects:

- The matching rule and attribute syntax specifications from the superior attribute type can be inherited by the subordinate type if the subordinate does not override the superior definition. For example, if the superior attribute type uses the `IA5 String` syntax, then the subordinate attribute type also uses the `IA5 String` syntax unless its definition overrides that by specifying an alternate syntax. According to the specification in RFC 4512 (`http://www.ietf.org/rfc/rfc4512.txt`), section 2.5.1, an attribute type can have a different syntax than its superior type only if the syntax for the subordinate type is a refinement of (that is, allows a subset of the values of) the syntax for the superior attribute type.
- The OID, any of the human-readable names associated with the superior attribute type, or both can be used to collectively reference all of the subordinate types. For example, the name attribute type is referenced as the superior type for the `cn`, `sn`, `c`, `l`, `st`, `o`, `ou`, `title`, `givenName`, `initials`, `generationQualifier`, and `dmdName` attribute types. Therefore, a filter of `(name=test)` should match an entry if any attribute with one of those types has a value of `test`.

A subordinate attribute type cannot have a different usage than its superior type. That is, if the superior type is `userApplications`, then the subordinate type must also be `userApplications`. Similarly, if a superior type is declared `COLLECTIVE`, then the subtype must also be `COLLECTIVE`, but if the superior type is not `COLLECTIVE`, then the subordinate type must also not be `COLLECTIVE`.

### 9.3.3 Attribute Type Implementation

At the present time, the mechanism used to handle attribute types varies from the LDAPv3 specification in the following ways:

- The LDAPv3 specification states that a subordinate attribute type must have the same syntax as the superior type, or a refinement of that syntax. Oracle Unified Directory does not enforce this constraint because it does not have any way to determine whether one attribute syntax is a refinement of the syntax of the supertype.
- The synchronization subsystem does not take attribute usage into account (for example, so that attribute types with a usage of `dSAOperation` are not synchronized).

## 9.4 Understanding Object Classes

Object classes are essentially named sets of attribute types that can be used to control the type of data that can be stored in entries. Note that the terms "object class" and "objectclass" (that is, with and without a space between the words) are generally used interchangeably.

The following sections describe object classes:


- Section 9.4.1, "Object Class Description Format"
- Section 9.4.2, "Object Class Kinds"
- Section 9.4.3, "Object Class Inheritance"
- Section 9.4.4, "Directory Server Object Class Implementation"

### 9.4.1 Object Class Description Format

The object class description format is described in RFC 4512 (http://www.ietf.org/rfc/rfc4512.txt), section 4.1.1.

```
ObjectClassDescription = LPAREN WSP
numericoid                 ; object identifier
[ SP "NAME" SP qdescrs ]   ; short names (descriptors)
[ SP "DESC" SP qdstring ]  ; description
[ SP "OBSOLETE" ]          ; not active
[ SP "SUP" SP oids ]       ; superior object classes
[ SP kind ]                ; kind of class
[ SP "MUST" SP oids ]      ; attribute types
[ SP "MAY" SP oids ]       ; attribute types
extensions WSP RPAREN
kind = "ABSTRACT" / "STRUCTURAL" / "AUXILIARY"
```

The object class description includes these elements:

`numericoid`
The numeric OID used to uniquely identify the object class in Oracle Unified Directory. Although the specification requires a numeric OID, Oracle Unified Directory also allows a non-numeric OID for the purpose of convenience and better compatibility with the Oracle Directory Server Enterprise Edition. In this case, the non-numeric OID should be the same as the name of the object class followed by the string `-oid`.

`NAME`
An optional set of human-readable names that can be used to refer to the object class. If there is a single name, then it should be enclosed in single quotes. If there are multiple names, then they should each be enclosed in single quotes separated by spaces, and the entire set of names should be enclosed in parentheses.

`DESC`
An optional human-readable description. If there is a description, then it should be enclosed in single quotation marks.

`OBSOLETE`
An optional `OBSOLETE` flag that can be used to indicate whether the object class is active. If an object class is marked as `OBSOLETE`, then it should not be referenced by any new elements created in Oracle Unified Directory.

`SUP`
An optional set of one or more superior classes for the object class. Note that although technically the specification allows an object class to have multiple superior classes, Oracle Unified Directory currently only supports a single superior class. In this case, the `SUP` keyword should be followed by a space and the name or OID of the superior class. If there are multiple superior classes, then they should be separated by dollar signs and the entire set of superior classes should be enclosed in parentheses.

`kind`
An optional keyword that specifies the kind of object class that is being defined. If this is specified, then it must be one of `ABSTRACT`, `STRUCTURAL`, or `AUXILIARY`. If no value is specified, then the object class is considered `STRUCTURAL`.

`MUST`
An optional set of attribute types for attributes that are required to be present (that is, have at least one value) in entries with that object class. If there is only a single

required attribute, then the MUSTkeyword should be followed by the name or OID of that attribute type. If there are multiple required attribute types, then they should be separated by dollar signs and the entire set of required attribute types should be enclosed in parentheses.

MAY
An optional set of optional attribute types for attributes that are allowed (but not required) to be present in entries with that object class. If there is only a single optional attribute, then the MAYkeyword should be followed by the name or OID of that attribute type. If there are multiple optional attribute types, then they should be separated by dollar signs and the entire set of optional attribute types should be enclosed in parentheses.

***extensions***
An optional set of extensions for the object class. Oracle Unified Directory currently uses the following extensions for object classes:

- X-ORIGIN — Provides information about where the object class is defined (for example, whether it came from a particular RFC or Internet Draft or if it is defined within the project).
- X-SCHEMA-FILE — Indicates which schema file contains the object class definition (This extension is generally used for internal purposes only and is exposed to clients.)

For example, the following is the object class description for the standard person object class:

```
( 2.5.6.6 NAME 'person' SUP top STRUCTURAL MUST ( sn $ cn )
MAY ( userPassword $ telephoneNumber $ seeAlso $ description )
X-ORIGIN 'RFC 4519' )
```

In this case, the OID is 2.5.6.6. There is a single human-readable name of person. The superior class is top. The kind is STRUCTURAL. Any entry containing the person object class is required to include the sn and cn attributes and is allowed to include the userPassword, telephoneNumber, seeAlso, and description attributes. The object class definition is taken from RFC 4519 (http://www.ietf.org/rfc/rfc4519.txt). There is no description, and the object class is not considered OBSOLETE.

## 9.4.2 Object Class Kinds

As described in Section 9.4.1, "Object Class Description Format," all object classes must have a kind of either ABSTRACT, STRUCTURAL, or AUXILIARY:

- ABSTRACT object classes are intended only to be extended by other object classes. An entry must not contain any abstract class unless it also contains a structural or auxiliary class that derives from that abstract class (that is, includes a non-abstract object class which has the abstract class in its inheritance chain). All entries must contain at least the top abstract object class in the inheritance chain for their structural class. They may or may not contain other abstract classes in the inheritance chains for their structural class or any of their auxiliary classes.
- STRUCTURAL object classes are intended to define the crux of what an entry represents. Every entry must include exactly one structural object class chain, and the root of that chain must ultimately be the top abstract object class. The structural object class for an entry cannot be changed.

- `AUXILIARY` object classes are intended to define additional qualities of entries. An entry can contain zero or more auxiliary classes, and the set of auxiliary classes associated with an entry can change over time.

The model represented by object class kinds translates very neatly to the model used by the Java programming language. Abstract LDAP object classes map directly to Java abstract classes, auxiliary LDAP object classes map directly to Java interfaces, and structural LDAP object classes map directly to Java concrete (non-abstract) classes. Just as Java classes must extend exactly one superclass but can implement any number of interfaces, so must LDAP entries contain exactly one structural class chain but can include any number of auxiliary class chains. Similarly, just as it is not possible to directly instantiate an abstract Java class, it is also not possible to create an LDAP entry containing only abstract object classes.

Oracle Directory Server Enterprise Edition has never enforced many of the restrictions noted here around object class kinds. In particular, it would allow the creation of entries that did not contain any structural object class chain and would also allow the creation of entries containing multiple structural object class chains. This means that deployments using Oracle Directory Server Enterprise Edition can contain entries that violate this constraint. Oracle Unified Directory does not allow this behavior by default, but for the sake of compatibility with existing Oracle Directory Server Enterprise Edition deployments, it is possible to configure Oracle Unified Directory to allow entries to violate this constraint, optionally writing a message to Oracle Unified Directory's error log each time this condition is detected. However, if there are entries that do not contain exactly one structural object class, then some schema elements like DIT content rules that depend on this constraint might not work as expected in all cases. To configure Oracle Unified Directory to accept these kinds of schema violations, set the `single-structural-objectclass-behavior` property of the global configuration. For more information, see "Global Configuration" in the *Oracle Unified Directory Configuration Reference*.

### 9.4.3 Object Class Inheritance

As specified in Section 9.4.1, "Object Class Description Format," object classes can have zero or more superior classes (although at the present time, Oracle Unified Directory supports at most one superior class). If an object class references a superior class, then all of the required and optional attributes associated with that superior class are also associated with the subordinate class.

The following restrictions exist for object class inheritance:

- `ABSTRACT` object classes can inherit only from other abstract classes. They cannot be subordinate to structural or auxiliary classes.
- `STRUCTURAL` object classes can inherit only from abstract classes or other structural classes. They cannot be subordinate to auxiliary object classes.
- `AUXILIARY` object classes can inherit only from abstract classes or other auxiliary classes. They cannot be subordinate to structural object classes.
- All `STRUCTURAL` object classes must ultimately inherit from the top abstract object class. The net effect of this is that every entry in Oracle Unified Directory must include the `top` object class and so must also include the `objectClass` attribute type, which is required by `the` top object class).

### 9.4.4 Directory Server Object Class Implementation

The mechanism used to handle object classes varies from the LDAPv3 specification in that object classes are allowed to have at most one superior class, whereas the specification allows multiple superior classes in some cases.

## 9.5 Understanding Name Forms

Name forms can be used to define a mechanism for naming entries in Oracle Unified Directory. In particular, a name form specifies one or more attribute types that must be present in the RDN of an entry with a given structural object class. A name form can also specify zero or more attribute types, which can optionally be present in the RDN.

Each structural object class defined in Oracle Unified Directory schema can be associated with one or more name forms. If a name form is defined for a given structural object class, then the associated name form is enforced for any add or modify DN operations for entries containing that object class. If a structural object class is not associated with a name form, then any attribute type that is allowed to exist in the target entry can be used as a naming attribute type.

### 9.5.1 Name Form Description Format

The name form description format is described in RFC 4512 (http://www.ietf.org/rfc/rfc4512.txt), section 4.1.7.2, as shown here:

```
NameFormDescription = LPAREN WSP
numericoid                ; object identifier
[ SP "NAME" SP qdescrs ]  ; short names (descriptors)
[ SP "DESC" SP qdstring ] ; description
[ SP "OBSOLETE" ]         ; not active
SP "OC" SP oids            ; structural object classes
SP "MUST" SP oids         ; attribute types
[ SP "MAY" SP oids ]      ; attribute types
extensions WSP RPAREN     ; extensions
```

The name form description includes these elements:

`numericoid`
The numeric OID used to uniquely identify the name form in Oracle Unified Directory. Although the specification requires a numeric OID, Oracle Unified Directory also allows a non-numeric OID for the purpose of convenience. In this case, the non-numeric OID should be the same as the name of the name form followed by the string `-oid`.

`NAME`
An optional set of human-readable names that can be used to refer to the name form. If there is a single name, then it should be enclosed in single quotes. If there are multiple names, then they should each be enclosed in single quotes separated by spaces, and the entire set of names should be enclosed in parentheses.

`DESC`
An optional human-readable description. If a description is present, then it should be enclosed in single quotation marks.

`OBSOLETE`
An optional `OBSOLETE` flag that can be used to indicate whether the name form is active. If a name form is marked as `OBSOLETE`, then it should not be in effect within

Oracle Unified Directory, nor should it be possible to create any other elements that depend on it.

`OC`
The names or OIDs of the structural object classes with which the name form is associated.

`MUST`
The names or OIDs of one or more attribute types that must be present in the RDN for any entry with the specified structural class. If there is only a single required attribute type, then only its name or OID needs to be given. If there are multiple required attribute types, then they should be separated by spaces and dollar signs, and the entire set of required attribute types should be enclosed in parentheses.

`MAY`
The names or OIDs of zero or more attribute types that can optionally be present in the RDN for any entry with the specified structural class. If there is only a single optional attribute type, then only its name or OID needs to be given. If there are multiple optional attribute types, then they should be separated by spaces and dollar signs, and the entire set of optional attribute types should be enclosed in parentheses.

***extensions***
An optional set of extensions for the name form. Oracle Unified Directory currently uses the following extensions for name forms:

- `X-ORIGIN` — Provides information about where the name form is defined (for example, whether it came from a particular RFC or Internet Draft or whether it is defined within the project.).
- `X-SCHEMA-FILE` — Indicates which schema file contains the name form definition (This extension is generally used for internal purposes only and is exposed to clients.)

For example, the following is the name form description for the `uddiBusinessEntityNameForm` name form defined in RFC 4403 (`http://www.ietf.org/rfc/rfc4403.txt`):

```
( 1.3.6.1.1.10.15.1 NAME 'uddiBusinessEntityNameForm'
OC uddiBusinessEntity MUST ( uddiBusinessKey ) X-ORIGIN 'RFC 4403' )
```

In this case, the numeric OID is `1.3.6.1.1.10.15.1` and the human-readable name is `uddiBusinessEntityNameForm`. Entries with the `uddiBusinessEntity` structural object class are required to use `uddiBusinessKey` as their only RDN attribute type. There is no description, nor are there any other attribute types that can optionally be included in the associated entries. The name form is not marked `OBSOLETE`.

## 9.6 Understanding DIT Content Rules

DIT content rules provide a mechanism for defining the content that can appear in an entry. At most one DIT content rule can be associated with an entry based on its structural object class. If such a rule exists for an entry, then it works in conjunction with the object classes contained in that entry to define which attribute types must, may, and must not be present in the entry, as well as which auxiliary classes that it may include.

The following sections describe DIT content rules:

- Section 9.6.1, "DIT Content Rule Description Format"

- Section 9.6.2, "DIT Content Rule Implementation"

## 9.6.1 DIT Content Rule Description Format

The DIT content rule description format is described in RFC 4512 (http://www.ietf.org/rfc/rfc4512.txt), section 4.1.6, as shown here:

```
DITContentRuleDescription = LPAREN WSP
numericoid                ; object identifier
[ SP "NAME" SP qdescrs ]  ; short names (descriptors)
[ SP "DESC" SP qdstring ] ; description
[ SP "OBSOLETE" ]         ; not active
[ SP "AUX" SP oids ]      ; auxiliary object classes
[ SP "MUST" SP oids ]     ; attribute types
[ SP "MAY" SP oids ]      ; attribute types
[ SP "NOT" SP oids ]      ; attribute types
extensions WSP RPAREN     ; extensions
```

The DIT content rule description includes these elements:

`numericoid`
The numeric OID of the structural object class with which the DIT content rule is associated. Although the specification requires a numeric OID, this `numericoid` should match the OID specified for the associated object class, so if the object class OID was non-numeric, then this OID should be as well.

`NAME`
An optional set of human-readable names used to refer to the DIT content rule. If there is a single name, then it should be enclosed in single quotes. If there are multiple names, then they should each be enclosed in single quotes separated by spaces, and the entire set of names should be enclosed in parentheses.

`DESC`
An optional human-readable description. If a description is provided, then it should be enclosed in single quotation marks.

`OBSOLETE`
An optional `OBSOLETE` flag that can be used to indicate whether the DIT content rule is active. If a DIT content rule is marked as `OBSOLETE`, then it should not be in effect within Oracle Unified Directory.

`AUX`
An optional list of auxiliary object classes that can be present in entries with the associated structural class. If no values are provided, then such entries are not allowed to have any auxiliary object classes. Values should be specified as one or more of the names or OIDs of the allowed auxiliary classes. If multiple auxiliary classes are allowed, then separate them by spaces and dollar signs, and enclose the entire set of names in parentheses.

`MUST`
An optional list of attribute types that are required to be present in entries with the associated structural class. This is in addition to the attribute types required by the object classes included in the entry, and these additional attribute types do not need to be allowed by any of those object classes. Values should be specified as one or more of the names or OIDs of the required attribute types. If multiple attribute types are required, then separate them by spaces and dollar signs, and enclose the entire set of required attribute types in parentheses.

MAY
An optional list of attribute types that can optionally be present in entries with the associated structural class. This is in addition to the attribute types allowed by the object classes included in the entry. Values should be specified as one or more of the names or OIDs of the optional attribute types. If there are multiple optional attribute types, separate them by spaces and dollar signs and enclose the entire set of optional attribute types in parentheses.

NOT
An optional list of attribute types that are prohibited from being present in entries with the associated structural class. This list cannot include any attribute types that are required by the structural class or any of the allowed auxiliary classes, but it can be used to prevent the inclusion of attribute types that would otherwise be allowed by one of those object classes. Values should be specified as one or more of the names or OIDs of the prohibited attribute types. If multiple types are prohibited, then separate them by spaces and dollar signs, and enclose the entire set of prohibited attribute types in parentheses.

***extensions***
An optional set of extensions for the DIT content rule. Oracle Unified Directory currently uses the following extensions for DIT content rules:

- X-ORIGIN — Provides information about where the DIT content rule is defined (for example, whether it came from a particular RFC or Internet Draft, or whether it is defined within the project)
- X-SCHEMA-FILE — Indicates which schema file contains the DIT content rule definition (This extension is generally used for internal purposes only and is exposed to clients.)

The following provides an example of a DIT content rule description:

```
( 2.16.840.1.113730.3.2.2 NAME 'inetOrgPersonContentRule'
AUX ( posixAccount $ shadowAccount $ authPasswordObject )
MUST uid )
```

In this case, the numeric OID is 2.16.840.1.113730.3.2.2, which is the OID for the inetOrgPerson structural object class. It has a human-readable name of inetOrgPersonContentRule and no description. It allows entries containing the inetOrgPerson object class to also contain the posixAccount, shadowAccount, and authPasswordObject auxiliary classes, and those entries must contain the uid attribute type. It is not marked OBSOLETE, and it does not define any additional optional or prohibited attribute types, nor does it include any extensions.

## 9.6.2 DIT Content Rule Implementation

At the present time, the mechanism used to handle DIT content rules varies from the LDAPv3 specification. The LDAPv3 specification states that if the structural object class used in an entry does not have a corresponding DIT content rule, then that entry is not allowed to contain any auxiliary object classes. Because Oracle Directory Server Enterprise Edition does not support DIT content rules, Oracle Unified Directory does not prevent the use of auxiliary object classes in entries for which there is no corresponding DIT content rule. If it is desirable to prevent the inclusion of auxiliary classes in a given type of entry, then a DIT content rule should be created with no allowed auxiliary classes to cover entries with the appropriate structural object class.

## 9.7 Understanding DIT Structure Rules

DIT structure rules can be used to define the allowed hierarchical structure of the directory data. In particular, they make it possible to specify what types of entries are allowed to exist as immediate children of entries with a specified structural object class. For example, only entries with the `inetOrgPerson` structural class can be immediate children of entries with an `organizationalUnit` structural object class.

DIT structure rules are themselves hierarchical. Each DIT structure rule is assigned a rule ID, which is an integer value, and is also associated with a name form (which in turn links it to one or more structural object classes). DIT structure rules can also reference one or more superior DIT structure rules, and this provides the mechanism for controlling the data hierarchy. If a DIT structure rule does not specify any superior rules, then entries containing its associated structural object class are allowed to exist at the root of the associated schema. If a DIT structure does specify one or more superior rules, then entries with an associated structural object class are allowed to exist only below entries containing the structural object class of one of those superior rules.

The following sections describe DIT structure rules:

- Section 9.7.1, "DIT Structure Rule Description Format"
- Section 9.7.2, "DIT Structure Rules and Multiple Schemas"

### 9.7.1 DIT Structure Rule Description Format

The DIT structure rule description format is described in RFC 4512 (`http://www.ietf.org/rfc/rfc4512.txt`), section 4.1.7.1, as shown here:

```
DITStructureRuleDescription = LPAREN WSP
ruleid                      ; rule identifier
[ SP "NAME" SP qdescrs ]    ; short names (descriptors)
[ SP "DESC" SP qdstring ]   ; description
[ SP "OBSOLETE" ]           ; not active
SP "FORM" SP oid            ; NameForm
[ SP "SUP" ruleids ]        ; superior rules
extensions WSP RPAREN       ; extensions
ruleids = ruleid / ( LPAREN WSP ruleidlist WSP RPAREN )
ruleidlist = ruleid *( SP ruleid )
ruleid = number
```

The DIT structure rule description includes these elements:

***ruleid***
The integer rule ID assigned to the DIT structure rule. It must be unique among all other DIT structure rules in the schema.

`NAME`
An optional set of human-readable names that can be used to refer to the DIT structure rule. If there is a single name, then it should be enclosed in single quotes. If there are multiple names, then they should each be enclosed in single quotes separated by spaces, and the entire set of names should be enclosed in parentheses.

`DESC`
An optional human-readable description. If a description is provided, then it should be enclosed in single quotes.

OBSOLETE
An optional OBSOLETE flag that can be used to indicate whether the DIT structure rule is active. If it is marked OBSOLETE, then it should not be taken into account when entries are created or moved.

FORM
The name or OID of the name form with which the DIT structure rule is associated. As mentioned in Section 9.7, "Understanding DIT Structure Rules," the name form associates the DIT structure rule with a structural object class.

SUP
An optional set of superior rule IDs for the DIT structure rule. If there are multiple superior rule IDs, then separate them by spaces, and enclose the entire set of superior rule IDs in parentheses. It is permissible for multiple DIT structure rules to use overlapping sets of superior rule IDs.

***extensions***
An optional set of extensions for the DIT structure rule. Oracle Unified Directory currently uses the following extensions for DIT structure rules:

- X-ORIGIN — Provides information about where the DIT structure rule is defined (for example, whether it came from a particular RFC or Internet Draft, or whether it is defined within the project)
- X-SCHEMA-FILE — Indicates which schema file contains the DIT structure rule definition (This extension is generally used for internal purposes only and is exposed to clients.)

The following example is the DIT structure rule definition for the uddiContactStructureRule DIT structure rule:

```
dITStructureRule:
( 2 NAME 'uddiContactStructureRule' FORM uddiContactNameForm SUP ( 1 )
X-ORIGIN 'RFC 4403' )
```

In this case, the rule ID is 2, and the human-readable name is uddiContactStructureRule. It is associated with the uddiContactNameForm name form (which in turn links it to the uddiContact object class), and it has a superior rule ID of 1. It was defined in RFC 4403 (http://www.ietf.org/rfc/rfc4403.txt). It does not have a description, nor is it marked OBSOLETE.

## 9.7.2 DIT Structure Rules and Multiple Schemas

DIT structure rules can provide a mechanism for placing constraints on Oracle Unified Directory hierarchy, but in order to maximize their utility, it may be necessary to use them in conjunction with support for multiple schemas. For example, consider a directory with a naming context of dc=example,dc=com, below which are two branches: ou=People,dc=example,dc=com and ou=Groups,dc=example,dc=com. If you want to allow only inetOrgPerson entries below the ou=People branch and only groupOfNames entries below the ou=Groups branch, then that can be fully accomplished only if there are different schemas that govern the ou=People and ou=Groups branches.

If there were a single schema governing the entire directory server, then you can imagine that it would have four DIT structure rules:

- dITStructureRule: (11 NAME 'domainStructureRule' FORM domainNameForm)

- dITStructureRule: (12 NAME 'organizationalUnitStructureRule' FORM organizationalUnitNameForm SUP 11)
- dITStructureRule: (13 NAME 'inetOrgPersonStructureRule' FORM inetOrgPersonNameForm SUP 12)
- dITStructureRule: (14 NAME 'groupOfNamesStructureRule' FORM groupOfNamesNameForm SUP 12)

This set of DIT structure rules would allow the structure described above, but it would also allow the creation of group entries below the ou=People branch and the creation of user entries below the ou=Groups branch. The only way to prevent that using DIT structure rules would be to define separate schemas for the ou=People and ou=Groups branches and define only the inetOrgPersonStructureRule rule in the schema for the ou=People branch, and only define the groupOfNamesStructureRule rule in the schema for the ou=Groups branch.

## 9.8 Understanding Matching Rule Uses

Matching rule uses can be used to specify which attribute types can be used in conjunction with a given matching rule when processing a search request with an extensible match filter component. If that extensible match component includes both an attribute type and a matching rule ID, then Oracle Unified Directory checks to see if there is a matching rule use for the associated matching rule, and if there is, it ensures that it allows the specified attribute type to be used with that matching rule.

The matching rule use description format is described in RFC 4512 (http://www.ietf.org/rfc/rfc4512.txt), section 4.1.4, as shown here:

```
MatchingRuleUseDescription = LPAREN WSP
numericoid                 ; object identifier
[ SP "NAME" SP qdescrs ]   ; short names (descriptors)
[ SP "DESC" SP qdstring ]  ; description
[ SP "OBSOLETE" ]          ; not active
SP "APPLIES" SP oids       ; attribute types
extensions WSP RPAREN      ; extensions
```

The matching rule use description includes these elements:

***numericoid***
The numeric OID of the matching rule with which the matching rule use is associated. There can be only one matching rule use associated with a given matching rule.

NAME
An optional set of human-readable names that may be used to refer to the matching rule use. If there is a single name, then it should be enclosed in single quotes. If there are multiple names, then they should each be enclosed in single quotes and separated by spaces, and the entire set of names should be enclosed in parentheses.

DESC
An optional human-readable description. If there is a description, then it should be enclosed in single quotes.

OBSOLETE
An optional OBSOLETE flag that can be used to indicate whether the matching rule use is active. If it is marked OBSOLETE, then it should not be taken into account when determining whether to allow an extensible match filter.

APPLIES
A set of one or more attribute types that can be used in conjunction with the associated matching rule. If there is an associated attribute type, then its name or OID can be used. If there are multiple attribute types, then separate them by spaces and dollar signs, and enclose the entire set of associated attribute types in parentheses.

***extensions***
An optional set of extensions for the matching rule use. Oracle Unified Directory currently uses the following extensions for matching rule uses:

- X-ORIGIN — Provides information about where the matching rule is defined (for example, whether it came from a particular RFC or Internet Draft, or whether it is defined within the project)
- X-SCHEMA-FILE — Indicates which schema file contains the matching rule definition (This extension is generally used for internal purposes only and is exposed to clients.)

The following example shows a matching rule use description:

```
( 1.3.6.1.4.1.26027.1.999.10 NAME 'testAddMRUSuccessful' APPLIES cn )
```

In this case, the numeric OID is 1.3.6.1.4.1.26027.1.999.10, the single human-readable name is testAddMRUSuccessful, and it can be used in conjunction with the cn attribute. It does not have a description, it is not marked OBSOLETE, and it does not have any extensions.

# 10

# Understanding Root Users and the Privilege Subsystem

Most LDAP directory servers typically have a single superuser, which is much like the root account in traditional UNIX systems. This account can bypass access controls and other restrictions that might be enforced for regular users. In Oracle Unified Directory you can define multiple root users, and a privilege subsystem that makes it possible to control capabilities at a more fine-grained level.

The following sections describe root user accounts and the privilege subsystem:



## 10.1 Root User Accounts

Root user accounts are defined below the `cn=Root DNs,cn=config` branch in the server configuration. Each root account is defined as a regular user entry, with the exception that it includes the `ds-cfg-root-dn-user` auxiliary object class. A root user entry can also have one or more values for the `ds-cfg-alternate-bind-dn` attribute. This attribute specifies alternate DNs that can be used to authenticate as that user (for example, so you can bind as `cn=Directory Manager` instead of having to use `cn=Directory Manager,cn=Root DNs,cn=config`, which is the actual entry DN).

The ability to define multiple root users, each in its own entry, has a number of advantages:

- Each administrator that needs root access to the directory server can have their own account with their own credentials. This makes it easier to keep an audit trail of who does what in the directory server than if all of the administrators shared a single root account.
- Because each root user account has its own set of credentials, the credentials for one root user can be changed without impacting any of the other root users. It is not necessary to coordinate root password changes among all of the administrators because each of them has their own account. If an administrator leaves, that account can simply be deactivated or removed.
- Because each root user has its own entry, and you can put whatever attributes and object classes you want into that entry (as long as it also has the

`ds-cfg-root-dn-user` auxiliary object class), root users are capable of using strong authentication like the EXTERNAL or GSSAPI SASL mechanisms.

- Root users are subject to password policy enforcement. This means that you can force root users to change their passwords on a regular basis, ensure that they are only allowed to authenticate or change their passwords using secure mechanisms, and ensure that they choose strong passwords. You can also use custom password policies for root users, so that they are subject to different sets of password policy requirements than other users in the directory.
- You can define different resource limits for root users than for regular users. Because each root account has its own entry, operational attributes like `ds-rlim-size-limit`, `ds-rlim-time-limit`, and `ds-rlim-lookthrough-limit` work for root users just as they do with regular user accounts.

## 10.2 Privilege Subsystem

As mentioned above, root user accounts in traditional directories are special because they can bypass access controls and other restrictions, and there are some kinds of operations that only root users can perform. This is much like the concept of root users in traditional UNIX operating systems. However, there might be cases in which a regular user needs to do something that only a root user can do. If users are given root access, they are given far more power than they actually need to do their job, and system administrators have to hope that they use this power responsibly and do not intentionally or unintentionally impact some other part of the system. Alternately, the user might not be given root access and either not be able to perform a vital function or have to rely on one of the system administrators to perform the task.

Solaris 10 and onward address this problem in UNIX systems by creating a privilege subsystem (also called "process rights management"). The engineers developing Solaris realized that it is dangerous and undesirable to be forced to give someone root access just to perform one specific task. For example, just because a user may need to start a process that listens on a port below 1024 does not mean that they should also be able to bypass filesystem permissions, change network interface settings, or mount and unmount file systems. With the privilege subsystem in Solaris 10, it is possible to give a user just the specific capability that they need, for example, the ability to bind to privileged ports, without giving them full root access. Similarly, it is possible to take away privileges that might otherwise be available. For example, an account that is only used to run a specific daemon does not need to be able to see processes owned by other users on the system.

**Note:** Administrators should consider Oracle Privileged Account Management system to achieve the best security level.

Oracle Unified Directory also has a privilege subsystem that defines distinct capabilities that users might need and makes it possible to give them just the level of access that they require. Regular users can be granted privileges that they would not otherwise have, certain privileges can be taken away from root users. The set of privileges currently defined in the directory server includes:

`bypass-acl`
Allows the user to bypass access control evaluation

`modify-acl`
Allows the user to make changes to the access controls defined in the server

`config-read`
Allows the user to have read access to the server configuration

`config-write`
Allows the user to have write access to the server configuration

`jmx-read`
Allows the user to read JMX attribute values

`jmx-write`
Allows the user to update JMX attribute values

`jmx-notify`
* Allows the user to subscribe to JMX notifications

`ldif-import`
Allows the user to request the LDIF import task

`ldif-export`
Allows the user to request the LDIF export task

`backend-backup`
Allows the user to request the backend backup task

`backend-restore`
Allows the user to request the backend restore task

`server-shutdown`
Allows the user to request the server shutdown task

`server-restart`
Allows the user to request the server restart task

`proxied-auth`
Allows the user to use the proxied authorization control or request an alternate SASL authorization ID

`disconnect-client`
Allows the user to terminate arbitrary client connections

`cancel-request`
* Allows the user to cancel arbitrary client requests

`unindexed-search`
Allows the user to request unindexed search operations

`password-reset`
Allows the user to reset the passwords for other users

`update-schema`
Allows the user to update the server schema

`privilege-change`
Allows the user to change the set of privileges assigned to a user, or to change the set of default root privileges

At the present time, the privileges marked with an asterisk (*) are not yet implemented in the server and therefore have no effect.

The privilege subsystem is largely independent of the access control subsystem. Unless the user also has the `bypass-acl` privilege, operations might still be subject to access control checking. For example, if a user has the `config-read` privilege, that user can see only those parts of the configuration that are allowed by access control. As a rule, whenever an operation is covered by both the privilege subsystem and access control, both mechanisms must allow that operation.

## 10.3 Assigning Privileges to Normal Users

By default, normal users are not granted any of the privileges described previously. Therefore, if a user should be allowed to perform any of the associated operations, they must be granted the appropriate privileges. This can be done by adding the `ds-privilege-name` operational attribute to the user's entry.

> **Note:** Adding a privilege with a value such as `modify-acl` is not sufficient for granting a user the right to add, replace, or delete an ACI. Appropriate access control for the user to modify the ACI for another entry is also required. See Appendix 8.2, "ACI Syntax" for more information.

`ds-privilege-name` is a multivalued attribute, and if a user is to be given multiple privileges, then a separate value should be used for each one. When the virtual attribute subsystem is in place, it should also be possible to grant privileges to groups of users automatically by making `ds-privilege-name` a virtual attribute in those user entries.

As an example, the following modification can be used to add the `proxied-auth` privilege to the user `cn=Proxy User,dc=example,dc=com`:

```
dn: cn=Proxy User,dc=example,dc=com
changetype: modify
add: ds-privilege-name
ds-privilege-name: proxied-auth
```

## 10.4 Assigning Privileges to Root Users

With the introduction of the privilege subsystem, the primary distinguishing characteristics of root users that separate them from other accounts in the server are that they exist in the configuration rather than in the user data, and that because they are root users they automatically inherit a certain set of privileges. The set of privileges automatically granted to root users is defined in the `ds-cfg-default-root-privilege-name` attribute of the `cn=Root DNs,cn=config` entry. By default, root users are automatically granted the following privileges:

- `bypass-acl`
- `modify-acl`
- `config-read`
- `config-write`
- `ldif-import`

- ldif-export
- backend-backup
- backend-restore
- server-shutdown
- server-restart
- disconnect-client
- cancel-request
- unindexed-search
- password-reset
- update-schema
- privilege-change

If you want to alter the set of privileges that are automatically assigned to root users, then you may do so by editing the ds-cfg-default-root-privilege-name attribute. Further, if you want to have a different set of privileges for a specific root user, then you can accomplish that using the ds-privilege-name attribute in that root user's entry, just like for a normal user. For example, the following modification may be used to give a specific root user (in this case cn=Test Root User,cn=Root DNs,cn=config) the ability to use proxied authorization while removing the ability to change user privileges or access the configuration. (The minus sign before the privilege indicates that it is being removed rather than granted.):

```
dn: cn=Test Root User,cn=Root DNs,cn=config
changetype: modify
add: ds-privilege-name
ds-privilege-name: proxied-auth
ds-privilege-name: -config-read
ds-privilege-name: -config-write
```

In this case, the cn=Test Root User,cn=Root DNs,cn=config user inherits all privileges automatically granted to root users with the exception of the config-read and config-write privileges and is also given the proxied-auth privilege.

# 11

# Understanding the Proxy Functionality

This chapter describes the functionality that is specific to a proxy server instance, and covers the following topics:

- Section 11.1, "Load Balancing Using the Proxy"
- Section 11.2, "Data Distribution Using the Proxy"
- Section 11.3, "Global Index Catalog"
- Section 11.4, "DN Renaming Using the Proxy"
- Section 11.5, "RDN Changing"
- Section 11.6, "Understanding the Transformation Framework"

> **Note:** Before you read this chapter, review Chapter 1, "Overview of Oracle Unified Directory" for a better understanding of the concepts described here.

## 11.1 Load Balancing Using the Proxy

You can use the proxy to load balance requests across multiple data sources or replicated LDAP servers.

In a load balancing deployment, the requests are routed to one of the data sources based on the *load balancing algorithm* set.

You can choose one of the following load balancing algorithms:

- Failover. Several remote LDAP server handle requests, based on the priority configured on a server, for a given operation type. When there is a failure, requests are sent to the server with the next highest priority for that operation type.

  For more information, see Section 11.1.1, "Failover Load Balancing."

- Optimal. There is no priority between the different remote LDAP servers. The LDAP server with the lowest saturation level is the one that handles the requests. The saturation level of the remote LDAP servers is regularly reevaluated, to ensure that the best route is chosen.

  For more information, see Section 11.1.2, "Optimal Load Balancing."

- Proportional. All the remote LDAP servers handle requests, based on the proportions (weight) set.

  For more information, see Section 11.1.3, "Proportional Load Balancing."

- Saturation. There is one main LDAP server that handles all requests, until the saturation limit is reached.

  For more information, see Section 11.1.4, "Saturation Load Balancing."

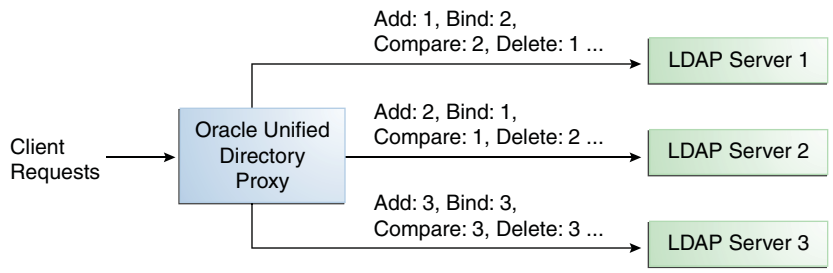
- Search Filter. Several LDAP servers are deployed, and handle requests based on certain attributes in the request search filter. For more information, see Section 11.1.5, "Search Filter Load Balancing."

### 11.1.1 Failover Load Balancing

In a load balancing with failover algorithm, the proxy routes requests to the remote LDAP server or data center with the highest priority for a given operation type, for example for Add operations. The proxy continues to send requests to the priority route until the remote LDAP server goes down. This may be caused by a network cut, a hardware failure, a software failure or some other problem. At failover, the proxy routes incoming requests to the server with the second highest priority for that specific operation type.

Figure 11–1 illustrates a failover load balancing configuration. In this example, there are three routes, each with a unique priority per operation type. All Add operations are treated by Server 1, since it has the highest priority, that is `priority=1`, while Bind operations are handled by Server 2. If Server 1 goes down, the Add requests are sent to the server with the second highest priority, that is, Server 2.

***Figure 11–1 Failover Load Balancing Example***




By default, the proxy does not immediately reroute requests to a server that has gone down, once it is running again. For example, if Server 1 goes down, the Add requests are sent to Server 2. Even when Server 1 is up again, Server 2 continues to handle incoming Add requests. However, if Server 2 goes down, and Server 1 is up again, Server 1 will now receive incoming requests. This default behavior can be changed with the `switch-back` flag. For information about configuring the `switch-back` flag, see Section 15.1.3.5.2, "Setting the `switch-back` Flag."

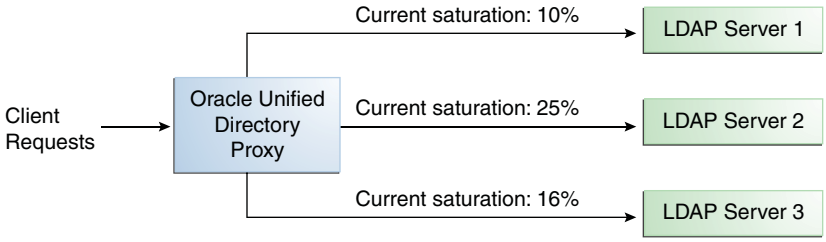
For failover to work effectively, the monitoring check interval must be set to be low enough so that the failover happens inside a time interval that suits your business needs. For details about setting the monitoring check interval, see Chapter 29, "Monitoring Oracle Unified Directory."

### 11.1.2 Optimal Load Balancing

With the optimal load balancing algorithm, the proxy sends requests to the route with the lowest saturation level. The proxy continues to send requests to this route until the saturation level of the remote LDAP server on that route passes the saturation level of the other remote LDAP servers in the deployment. The saturation level is represented as a percentage.

When the saturation level of a route changes, the load balancing algorithm re-evaluates the best route and if required, selects another route as the active one. The route with the lowest saturation level is always chosen as the optimal route. In the configuration illustrated by Figure 11–5, Server 1 has the lowest saturation level and will handle all the requests until its saturation level rises above the saturation level of the other servers. If one of the servers goes down, its saturation level is considered as 100%.

***Figure 11–2 Optimal Load Balancing Example***




You can configure the saturation precision, to set the difference of saturation between two servers before the route changes to the server with the lowest saturation level. By default, the saturation precision is set to 5. However, if you find that the algorithm is switching between servers too often, you can set the saturation precision to 10, for example. The saturation precision is set in the LDAP server extension, see Section 15.1.3.5.3, "Setting the Saturation Precision for the Optimal or Saturation Algorithm."

#### 11.1.2.1 Determining Saturation Level

The saturation level is a ratio between the number of connections in use in the connection pool and its configured maximum size. The connection pool maximum size is an advanced parameter of the LDAP server extension object.

If the number of connections in use is lower than the maximum pool size divided by 2, then the saturation is `0`. This implies that the pool is not saturated.

When more than half of the connections are in use, the saturation level is calculated as follows:

100 * (1 - available connections/(max pool size/2))

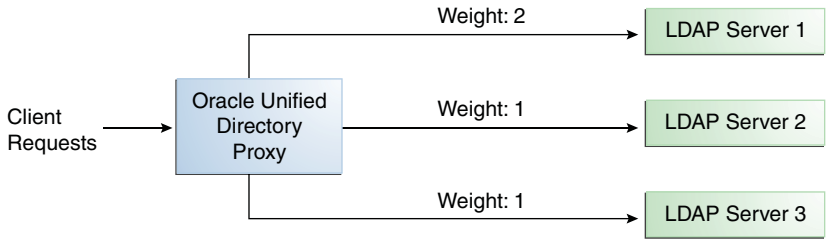
This implies that the saturation level is `100` when all the connections are in use.

### 11.1.3 Proportional Load Balancing

With the proportional load balancing algorithm, the proxy forwards requests across multiple routes to remote LDAP servers or data sources, based on the proportions set. The proportion of requests handled by a route is identified by the weight that you set for each route in your configuration. The weight is represented as an integer value.

When you configure load balancing, you must indicate the proportion of requests handled by each LDAP server. In the example in Figure 11–3, Server 1 handles twice as many connections as Server 2, since the weight is set with a proportion of 2:1. Server 2 and Server 3 handle the same amount of requests (1:1).

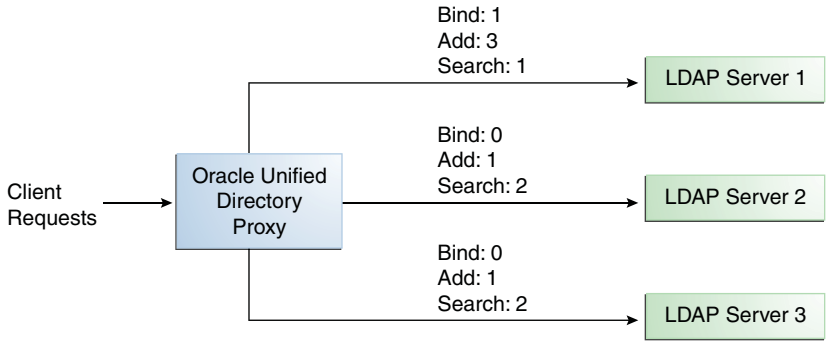
***Figure 11–3 Proportional Load Balancing Example***




You can configure a specific weight for each type of client operation, as illustrated in Figure 11–4. For example, in you want Server 1 to handle all the Bind operations, this is possible. To do so, set the weight of bind to `1` (or higher) for Server 1, and to `0` for Server 2 and Server 3.

In the example illustrated in Figure 11–4, Server 1 will handle three times as many Add requests as Server 2 and Server 3. However, Server 1 will handle only one half the Search requests handled by Server 2, and Server 3. Server 2 and Server 3 will handle the same amount of Add and Search requests, but will not handle Bind requests.

***Figure 11–4 Proportional Load Balancing with Request Specific Management***




If you do not modify the weights of operations other than Bind, Add, and Search, as illustrated in Figure 11–4, the servers will share the same load for all other operations (for example for Delete operations).

For more information on configuring the load balancing weights of routes when using proportional load balancing, see Section 15.1.3.5, "Modifying Load Balancing Properties."

## 11.1.4 Saturation Load Balancing

With the saturation load balancing algorithm, the proxy sends requests to a chosen priority route. The proxy continues to send requests to the priority route until the remote LDAP server on that route passes the saturation threshold set. The saturation threshold is represented as a percentage.

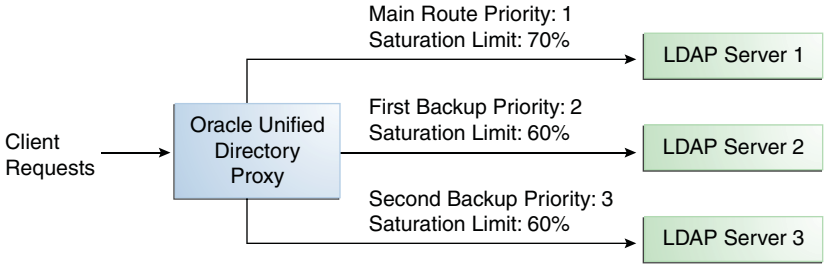
For example, if you want a remote LDAP server to manage all incoming requests, set it as priority 1. If you want that same remote LDAP server to stop handling requests when its saturation index reaches 70%, set the saturation threshold to 70%, as illustrated in Figure 11–5. In this way, the server handles all incoming requests until it becomes 70% saturated. The proxy then sends all new requests to the remote LDAP server to Server 2, since it has the next highest priority. Server 2 will continue to

handle requests until it reaches its own saturation threshold, or until Server 1 is no longer saturated.

In other words, if Server 1 reaches 70% saturation, the proxy directs the requests to Server 2. If Server 1 is still at 70%, and Server 2 reaches 60%, the proxy directs the new requests to Server 3.

However, if while Server 2 is handling requests, the saturation level of Server 1 drops to 55%, the proxy will direct all new requests to Server 1, even if Server 2 has not reached its saturation threshold.

***Figure 11–5 Saturation Load Balancing Example***




If all routes have reached their saturation threshold, the proxy chooses the route with the lowest saturation.

You can set a saturation threshold alert that warns you when a server reaches its saturation limit. For example, if you set a saturation threshold alert to 60%, you will receive a notification when the server reaches this limit, and you can act before the server becomes too degraded.

For more information about how to determine the saturation level, see Section 11.1.2.1, "Determining Saturation Level."

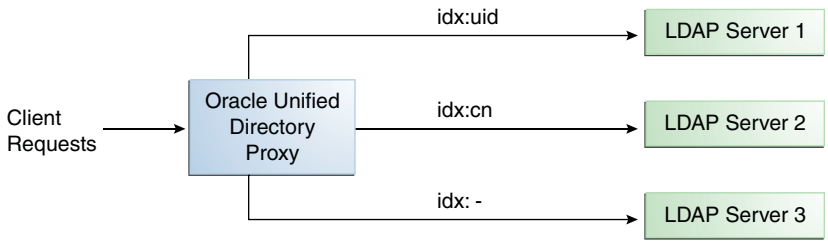
## 11.1.5 Search Filter Load Balancing

With the search filter load balancing algorithm, the proxy routes search requests to LDAP servers based on the presence of certain attributes defined in the request search filter.

The topology consists of several LDAP servers that are accessible through the proxy. All the LDAP servers contain similar data, but each server is optimized based on attributes defined in the search filter to provide better performance. You can configure each route with a list of allowed attributes and a list of prohibited attributes. A search request matches a route when the request search filter contains at least one allowed attribute, and none of the prohibited attributes.

The Figure 11–6 illustrates a search filter load balancing algorithm. In this example, there are three LDAP servers and therefore three distinct routes. LDAP server 1 indexes the `uid` attribute, LDAP server 2 indexes the `cn` attribute, and the third LDAP server is a pass-through route.

***Figure 11–6 Search Filter Load Balancing***




When the proxy receives a search request that contains the `uid` attribute in its search filter, the search request is routed to LDAP server 1 for better performance. Similarly, if the search filter contains a `cn` attribute, then the search request is routed to LDAP server 2. All other search requests are routed to the pass-through LDAP server 3.

All other requests, such as ADD, DELETE, MODIFY, and so on can be routed to any LDAP server based on the highest priority. Each search filter route is given a priority. This priority determines the order in which the route are evaluated. The highest priority route filter that matches the search filter is selected to process the request. If all the search filter routes have the same priority, then any route can process the request.

## 11.2 Data Distribution Using the Proxy

The Oracle Unified Directory distribution feature addresses the challenge of large deployments, such as horizontal scalability, where all the entries cannot be held on a single data source, or LDAP server. Using distribution can also help you scale the number of updates per second.

In a distribution deployment, you must first split your data into smaller chunks. To split the data, you can use the `split-ldif` command. These chunks of data are called *partitions*. Typically, each partition is stored on a separate server.

The split of the data is based on one of the following distribution algorithms:

- Numeric. Entries are split into partitions and distributed based on the numeric value of the naming attribute (for example uid). See Section 11.2.1, "Numeric Distribution" for more information.
- Lexico. Entries are split into partitions and distributed based on the alphabetical value of the naming attribute (for example cn). See Section 11.2.2, "Lexico Distribution" for more information.
- Capacity. Entries are added to a partition based on the capacity of each partition. This algorithm is used for Add requests only. All other requests are distributed by the global index catalog or by a broadcast. See Section 11.2.3, "Capacity Distribution" for more information.
- DN pattern. Entries are split into partitions and distributed based on the pattern (value) of the entry DN. See Section 11.2.4, "DN Pattern Distribution" for more information.

The type of data distribution you choose will depend on how the data in your directory service is organized. Numeric and lexico distribution have a very specific format for distribution. DN pattern can be adapted to match an existing data distribution model.

If a client request (except Add) cannot be linked to one of the distribution partitions, the proxy broadcasts the incoming request to all the partitions, unless a *global index catalog* has been configured.

However, if the request is clearly identified as outside the scope of the distribution, the request is returned with an error indicating that the entry does not exist. For example, if the distribution partitions includes data with uid's from 1-100 (`partition1`) and 100-200 (`partition2`) but you run a search where the base DN is `uid=222,ou=people,dc=example,dc=com`, the proxy will indicate that the entry does not exist.

Moreover, for the numeric and lexico algorithms, it is the *first RDN* after the distribution base DN that is used to treat a request. For example, the following search will return an error, as the uid is not the first RDN after the distribution base DN, for example `ou=people,dc=example,dc=com`.

```
$ ldapsearch -b "uid=1010,o=admin,ou=people,dc=example,dc=com" "objectclass=*"
```

Consider the number of partitions carefully. When you define the number of partitions you want in your deployment, you should note that you cannot split and redistribute the data into new partitions without downtime. You can, however, add a new partition with data that has entries outside the initial ones.

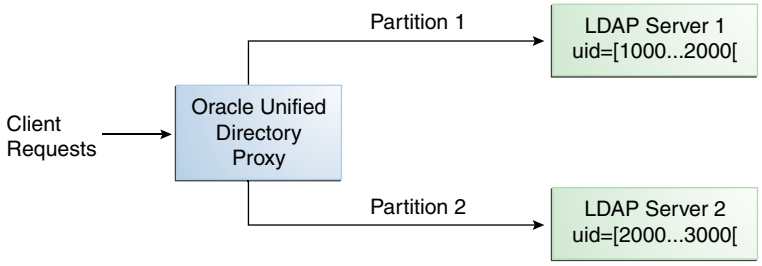
For example, if the initial partitions cover data with uids from 1-100 (`partition1`) and 100-200 (`partition2`), you can later add a `partition3` which includes uids from 200-300. However, you cannot easily split `partition1` and `partition2` so that `partition1` includes uids 1-150 and `partition2` includes uids 150-300, for example. Splitting partitions is essentially like reconfiguring a new distribution deployment.

### 11.2.1 Numeric Distribution

With a distribution using numeric algorithm, the proxy forwards requests to one of the partitions, based on the numeric value of the first RDN after the distribution base DN in the request. When you set up distribution with numeric algorithm, you split the data of your database into different partitions based on a numerical value of the attribute of your choice, as long as the attribute represents a numerical string. The proxy then forwards all client requests to the appropriate partition, using the same numeric algorithm.

For example, you could split your data into two partitions based on the uid of the entries, as illustrated in Figure 11–7.

***Figure 11–7 Numeric Distribution Example***




In this example, a search for an entry with a uid of `1111` is sent to Partition 1, while a search for an entry with a uid of `2345` is sent to Partition 2. Any request for an entry

with a uid outside the scope of the partitions defined will indicate that no such entry exists.

> **Note:** The upper boundary limit of a distribution algorithm is exclusive. This means that a search for uid `3000` in the example above returns an error indicating that the entry does not exist.

***Example 11–1 Examples of Searches Using Numeric Distribution Algorithm***

The following search will be successful:

```
$ ldapsearch -b "uid=1010,ou=people,cn=example,cn=com" "cn=Ben"
```

However, the following searches will indicate that the entry does not exist (with result code `32`):

```
$ ldapsearch -b "uid=1010,o=admin,ou=people,cn=example,cn=com" "objectclass=*"

$ ldapsearch -b "uid=99,ou=people,cn=example,cn=com" "objectclass=*"
```

The following search will be broadcast, as the proxy cannot determine the partition to which the entry belongs, using the distribution algorithm defined above:

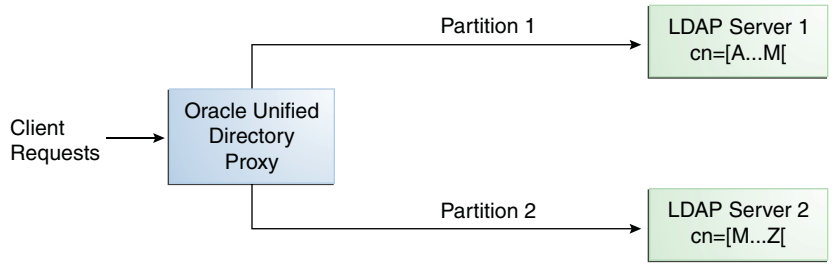
```
$ ldapsearch -b "ou=people,cn=example,cn=com" "uid=*"
```

## 11.2.2 Lexico Distribution

With a distribution using lexico algorithm, the proxy forwards requests to one of the partitions, based on the alphabetical value of the first RDN after the distribution base DN in the request. When you set up distribution with lexico algorithm, you split the data of your database into different partitions, based on an alphabetical value of the attribute of your choice. The proxy then forwards all client requests to the appropriate partition, using the same algorithm.

For example, you could split your data into two partitions based on the cn of the entries, as illustrated in Figure 11–8.

***Figure 11–8 Lexico Distribution Example***




In this example, any requests for an entry with a cn starting with `B` such as `Ben` are sent to Partition 1, while requests for an entry with a cn from `M-Y` are sent to Partition 2.

**Note:** The upper boundary limit of a distribution algorithm is exclusive. This means that a search for cn= `Zachary` in the example above will indicate that no such entry is found. In order to include entries starting with Z in the search boundaries, then you should use the `unlimited` keyword. For example, `cn=[M..unlimited[`will include all entries beyond M.

***Example 11–2 Examples of Searches Using Lexico Distribution Algorithm***

The following search will be successful:

```
$ ldapsearch -b "cn=Ben,ou=people,cn=example,cn=com" "objectclass=*"
```

The following search will also be successful, as `cn=Ben` is the *first* RDN.

```
$ ldapsearch -b "uid=1010,cn=Ben,ou=people,cn=example,cn=com" "objectclass=*"
```

However, the following searches will indicate that the entry does not exist (with result code `32`):

```
$ ldapsearch -b "cn=Ben,o=admin,ou=people,cn=example,cn=com" "objectclass=*"

$ ldapsearch -b "cn=Zach,ou=people,cn=example,cn=com" "objectclass=*"
```

The distribution cannot determine to which partition the following search belongs and will be broadcast:

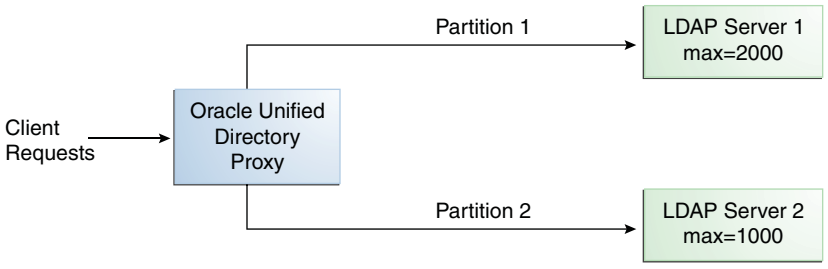
```
$ ldapsearch -b "ou=people,cn=example,cn=com" "cn=*"
```

### 11.2.3 Capacity Distribution

With a capacity-based distribution, the proxy sends Add requests based on the capacity of each partition, which is determined by the maximum number of entries the partitions can hold. All other requests are distributed by the global index catalog or by broadcast.

Because the data is distributed to the partitions in a completely random manner, the easiest way to identify on which partition a particular data entry is by using a global index. Global index is mandatory when using capacity distribution. If no global index is set up, all requests other than Add will have to be broadcast. For more information about global indexes, see Section 11.3, "Global Index Catalog" and Section 15.1.7, "Configuring Global Indexes By Using the Command Line."

***Figure 11–9 Capacity Distribution Example***




In the example illustrated in Figure 11–9, Partition 1 has twice the capacity of Partition 2, therefore Partition 1 will receive twice the add requests sent to Partition 2. This way,

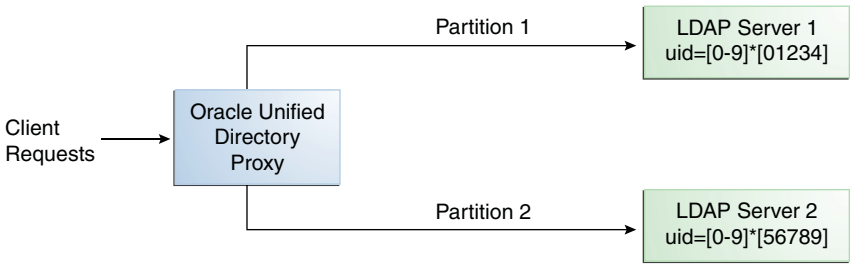
both partitions should be full at the same time. When all the partitions are full, the distribution will send one request to each partition at each cycle.

### 11.2.4 DN Pattern Distribution

With a distribution using DN pattern algorithm, the proxy forwards requests to one of the partitions, based on the match between a request base DN and a string pattern. The match is only perform on the relative part of the request base DN, that is, the part after the distribution base DN. For example, you could split your data into two partitions based on a the DN pattern in the uid of the entries, as illustrated in Figure 11–10.

Distribution using DN pattern is more onerous than distribution with numeric or lexico algorithm. If possible, use another distribution algorithm.

***Figure 11–10 DN Pattern Distribution Example***

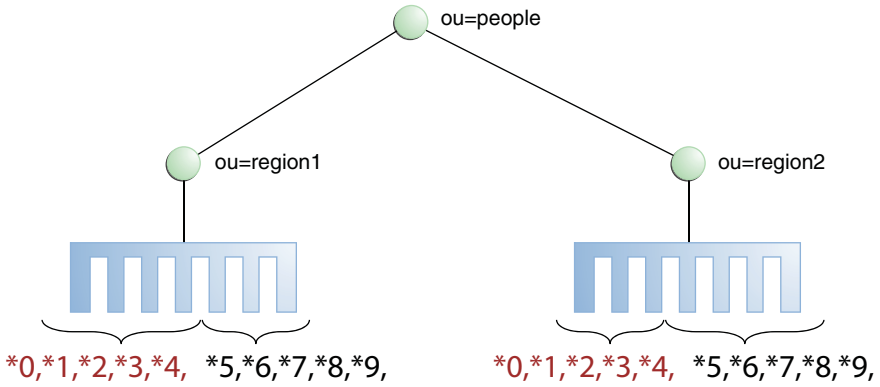



In this example, all the data entries with a uid that ends with 0, 1, 2, 3, or 4 will be sent to Partition 1. Data entries with a uid that ends with 5, 6, 7, 8, or 9 will be sent to Partition 2.

This type of distribution, although using numerical values is quite different from numeric distribution. In numerical distribution, the data is partitioned based on a numerical *range*, while DN pattern distribution is based on a *pattern* in the data string.

Distribution using a DN pattern algorithm is typically used in cases where the distribution partitions do not correspond exactly to the distribution base DN. For example, if the data is distributed as illustrated in Figure 11–11, the data for Partition 1 and Partition 2 is in both base DN `ou=people,ou=region1` and `ou=people,ou=region2`. The only way to distribute the data easily is to use the DN pattern.

*Figure 11–11 Example of Directory Information Tree*




***Example 11–3 Example of DN Pattern Algorithm Split by Region***

If the deployment of the information is based in two geographical locations, it may be easier to use the DN pattern distribution to distribute the data. For example, if employee numbers were 4 digit codes, where the first digit indicated the region, then you could have the following:

| Region 1 | Region 2 |
|---|---|
| 1000 | 2000 |
| 1001 | 2001 |
| 1002 | 2002 |
| 1003 | 2003 |
| 1004 | 2004 |
| 1005 | 2005 |
| 1006 | 2006 |
| 1007 | 2007 |
| 1008 | 2008 |
| 1009 | 2009 |
| 1010 | 2010 |
| ... | ... |

In order to spread the load of data, the entries in each location are split over two servers, where Server 1 contains all entries that end with 0, 1, 2, 3, and 4, while Server 2 contains all the entries that end with 5, 6, 7, 8, and 9, as illustrated in Figure 11–10.

Therefore, a search for DN pattern `1222` would be sent to partition 1, as would `2222`.

## 11.3 Global Index Catalog

A global index catalog can be used with a distribution deployment. If you are configuring a capacity based distribution, you must have a global index, with DN indexed. The global index catalog maps the entries to the distribution partition in which the data is held. When the proxy receives a request from the client, the distribution looks up the attribute entry in the global index catalog, and forwards the

client request to the correct partition. This diminishes the need for broadcasts. Moreover, if a modify DN request is made, the global index catalog will ensure that the entry is always found.

A global index catalog maps the entries based on specific attributes, such as employee number or telephone number. The value of the attribute to be indexed *must be unique* across all the entries. You cannot use a global index to map entries based on country, for example, as that information is not unique.

If you index an attribute whose values are not unique, the proxy server might be unable to return all the requested entries. Say, for example, that you index the `mail` attribute, whose values are not necessarily unique. You now add the following two entries in sequence:

- Entry 1, with `uid=user.1` and `mail=joe.smith@example.com` is sent to partition 1.
- Entry 2, with `uid=user.2` and `mail=joe.smith@example.com` is sent to partition 2.

In this situation, the global index `mail` keeps reference to the second entry only. A search with the filter `(mail=joe.smith@example.com)` will return only the second entry, `uid=user.2`.

A global index catalog can include several global indexes. Each global index maps a different attribute. For example, you can have one global index catalog called `GI-catalog`, which includes a global index mapping the entries based on the *telephone number* and one mapping the entries based on the *employee number*. This means that you can forward client requests to the right partition using either the telephone number or the employee number.

Global index catalogs and global indexes are created and configured using the `gicadm` command. For more information see Section 15.1.7, "Configuring Global Indexes By Using the Command Line" and Appendix A.2.7, "gicadm."

The global indexes can be populated with data from LDIF files. The data from one LDIF file can be split into partitions using the `split-ldif` command. For more information, see Appendix A.3.15, "split-ldif."

A global index catalog should be replicated to avoid a single point of failure. For information on replicating the global index catalog, see Section 15.1.7.2, "Replication of Global Index Catalogs."

#### *Example 11–4 Using a Global Index Catalog for Telephone Numbers*

A typical example of a unique attribute which can be used to create a global index is a telephone number: the value of the attribute is unique, that is, only one person (employee, for example) can have that telephone number.

In the example below, the entries in the database have been split based on the telephone number. The global index includes the following information:

| Value | Partition ID |
|---|---|
| 4011233 | 1 |
| 4011234 | 1 |
| 7054477 | 2 |

The global index does not store the name of the employees, location, and other attribute values that may be associated to the telephone number. It only maps the

attribute indexed to the partition. The data associated to the indexed value (here telephone number) is stored in the remote LDAP server.

If an employee has multiple phone numbers, these are regarded as multi-valued entries. In this case, if the global index is created based on the telephone number, there will be two global index entries that will result in finding one employee, say Ben Brown.

In the example above, employee Ben Brown could have both telephone number `4011233` and `7054477` assigned to him. In this case, a search on one of Ben Brown's telephone number would return the correct partition, and all the information associated to the telephone number, including the name Ben Brown, regardless that he has two phone numbers attributed to him.

## 11.4 DN Renaming Using the Proxy

Each entry in a directory is identified by a DN and a set of attributes and their values. Sometimes, the DN and the attributes defined on the client side do not map with the DN and the attributes defined on the server side. For instance, an organization, Example A contains `dc=parentcompany`, `dc=com` entries. It acquires another organization, Example B. Example B contains `dc=newcompany`, `dc=com` entries. Therefore, `dc=newcompany`, `dc=com` must be renamed to `dc=parentcompany`, `dc=com` for the existing client applications to work correctly.

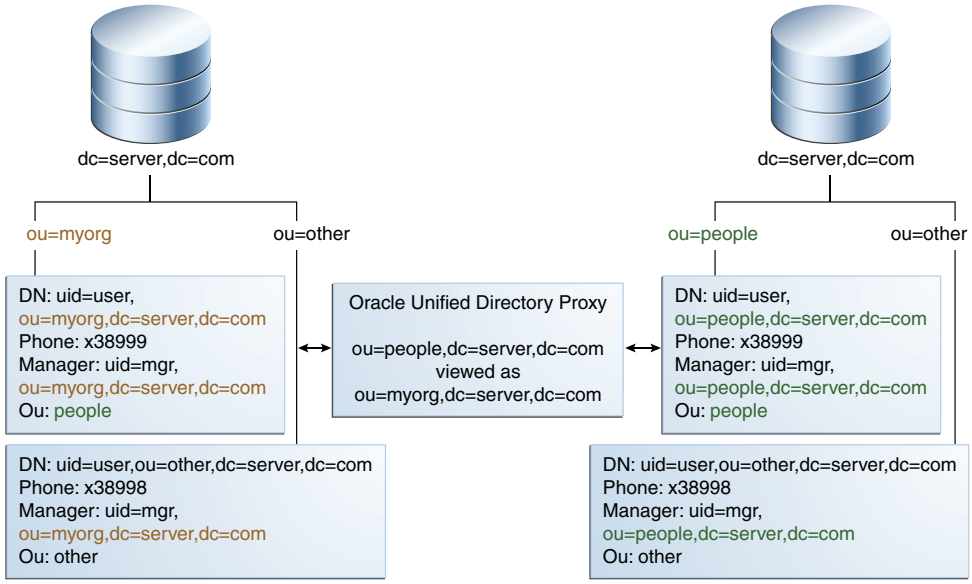
You can define a DN renaming workflow element to rename DNs to values that match the server side. When a client makes a request, the DNs and attributes are renamed to match those in the server. When the result is returned to the client, the DN and attributes are changed back to match what the client has requested.

### 11.4.1 How the DN Renaming Workflow Element Works

Oracle Unified Directory provides a DN renaming workflow element that allows you to transform the content of a Directory Information Tree (DIT) into another DIT with a different base DN. When an operation (Add, Bind, Delete, Modify, and so on) goes through a DN renaming workflow element, its parameters are transformed according to the DN renaming configuration to transform the virtual entries into real entries.

Figure 11–12 illustrates how DN renaming is performed using the proxy.

***Figure 11–12 DN Renaming***




The client expects `ou=myorg`, `dc=server`, `dc=com` entries. However, the LDAP server contains `ou=people`, `dc=server`, `dc=com` entries. The proxy renames the DNs by making use of the DN renaming workflow element.

In this example, the real entries `ou=people`, `dc=server`, `dc=com` are seen as `ou=myorg`, `dc=server`, `dc=com` entries from the client side.

The DN renaming transformation is applicable to the following objects:

- DN of the entry: For example, the real entry on the LDAP server `dn:uid=user`, `ou=people`, `dc=server`, `dc=com` is transformed into a virtual entry `dn:uid=user`, `ou=myorg`, `dc=server`, `dc=com` from the client perspective.
- Attributes of the entry that contain DNs: For example, the server-side value of the manager attribute of an entry with an objectclass `inetorgpersonmanager`: `manager:uid=mgr`, `ou=people`, `dc=server`, `dc=com` is transformed into the value `manager:uid=mgr`, `ou=myorg`, `dc=server`, `dc=com` on the client side.

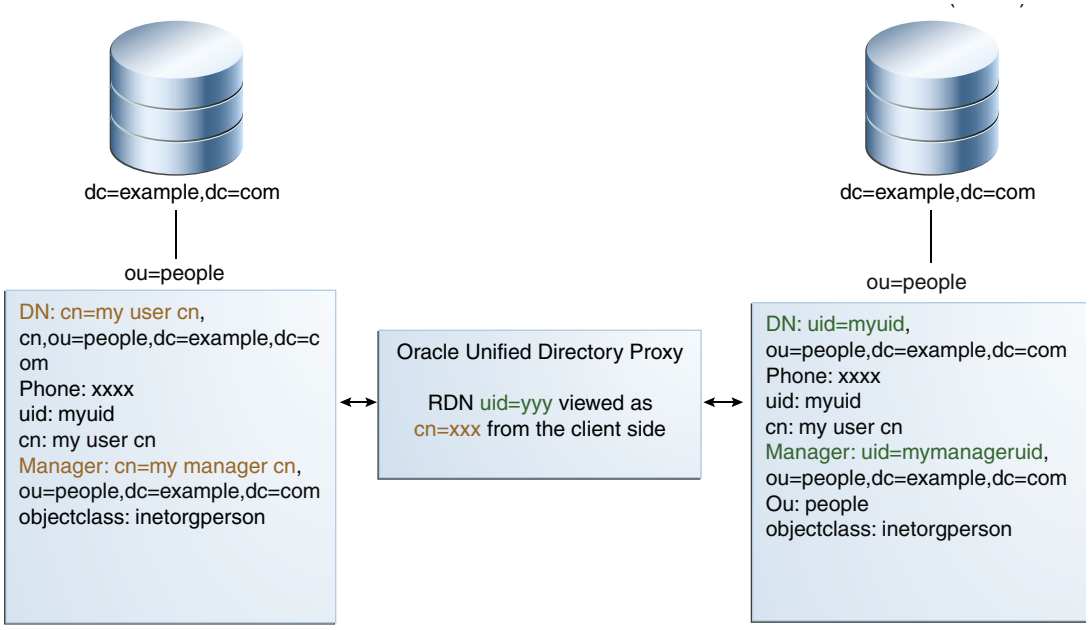
> **Note:** You can apply the transformation to all the user attributes of the entries, define a restricted list of attributes to which the operation applies, or define a restricted list of attributes to which the operation does not apply.

## 11.5 RDN Changing

Oracle Unified Directory enables you to rename or replace RDN values from the source directory to Oracle Unified Directory using the RDNChanging configuration.

Figure 11–13 illustrates how DN renaming is performed using the proxy.

***Figure 11–13 RDN Changing***




**Note:** The relative distinguished name (RDN) is the leftmost element in an entry DN. For example, the RDN for `uid=Marcia Garza,ou=People,dc=example,dc=com` is `uid=Marcia Garza`. You can only change the leftmost element in an entry DN.

The RDNChanging configuration has the following parameters:

**objectclass**
Identifies the objectclass type that RDN renaming is performed on. The default setting is `person`.

**replace-value**
True or False: Indicates whether the value of original RDN value in the client view (identified by the `source-rdn` parameter) should be replaced by the value of the new RDN value (identified by the `client-rdn` parameter). The default setting is `true`.

**Note:** When the value is set to true, and an entry has multiple values for the new RDN attribute, then Oracle Unified Directory uses the first value in RDN.

**source-rdn**
Identifies the original RDN attribute name from the source directory to be replaced or renamed in Oracle Unified Directory.

**client-rdn**
Identifies the new RDN attribute name to be used in Oracle Unified Directory and replaces the attribute name identified by the source-rdn configuration parameter.

**dn-attributes**
List of attributes with DNs to perform RDN renaming on. The default list of attributes are `member`, `manager`, and `owner`.

# 11.6 Understanding the Transformation Framework

Oracle Unified Directory supports transformation of data through the definition of workflow elements. By creating an instance of workflow element you can display physical data in a different way. This chapter describes how transformation in Oracle Unified Directory occurs and contains the following topics:

- Section 11.6.1, "Overview of Transformation"
- Section 11.6.2, "Components of Transformation"
- Section 11.6.3, "Configuring Transformation"

## 11.6.1 Overview of Transformation

The data structure of an LDAP client application may differ from the data structure of the LDAP repository. They may differ on the schema (different types of attribute in the entries) or the values (same attribute name with different semantic of values). This is where you need transformation.

A transformation performs a specific action in a certain direction. You have to specify the transformations that you need and define these on an existing workflow elements.

This section contains the following topics:

- Section 11.6.1.1, "Transformation Models"
- Section 11.6.1.2, "Implementing Transformation in Oracle Unified Directory"

### 11.6.1.1 Transformation Models

The direction of transformation that is whether the transformation is applied during the request, during the response, or both determines the transformation model.

Transformations can be categorized into the following types:

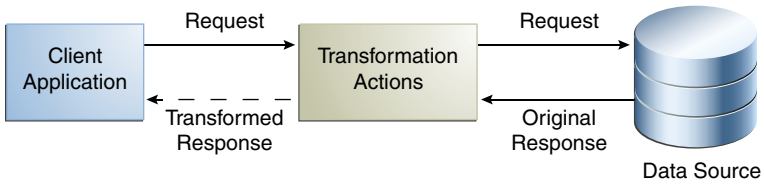
- **Read transformations** (outbound transformations): For more information, see Section 11.6.1.1.1, "Read Transformations."
- **Write transformations** (inbound transformations): For more information, see Section 11.6.1.1.2, "Write Transformations."
- **Mapping transformations** (bidirectional transformations): For more information, See Section 11.6.1.1.3, "Mapping Transformations."

#### 11.6.1.1.1 Read Transformations

The read transformation is the most common transformation. A read transformation is applied only during the response to a request. No transformation is applied during the request and the physical data is not changed.

Figure 11–14 illustrates the concept of a read transformation.

***Figure 11–14 Read Transformation***




Consider a scenario of an organization that has a legacy application whose function is to display person entries. The application does not support entries that do not contain an `email` attribute. The physical data source has been upgraded and the `email` attribute no longer exists for person entries.

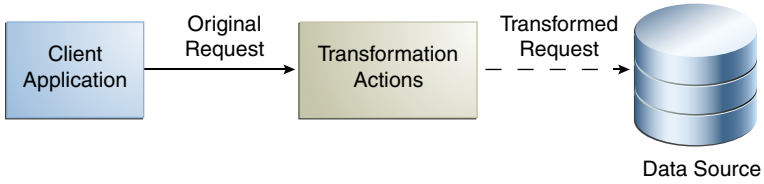
You need to apply a transformation here, which is to add the `email` attribute during the search response. This transformation changes the entry that is read from the data source and adds an `email` attribute whose value is `firstname.surname@mycompany.com`. No reverse transformation is required and the physical data is not changed.

#### 11.6.1.1.2 Write Transformations

A write transformation is applied during the request, but not during the response. A write transformation modifies data provided by the client before storing it in the backend.

Figure 11–15 illustrates the concept of a write transformation.

***Figure 11–15 Write Transformation***



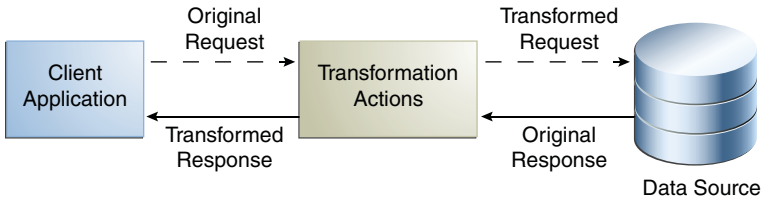

Consider a scenario of an organization that has a legacy application whose function is to add person entries to a data source. The application adds the entries without the `email` attribute. The physical data source has been upgraded and the `email` is now a mandatory attribute for person entries. You need to apply a transformation here, which is to add the `email` attribute during the add request. This transformation changes the entry that is written to the database. No reverse transformation is required.

#### 11.6.1.1.3 Mapping Transformations

The mapping transformation is the most common transformation. It is bidirectional in the sense that it is first applied during the request, and the reverse is applied during the response. These transformations are called mappings, because an attribute or entry in the physical data view maps to an attribute or entry in the virtual data view. Mapping transformations enable you to process existing values before assigning them to a DN component, an attribute type or value, or an object class.

Figure 11–16 illustrates the concept of a mapping transformation.

***Figure 11–16 Mapping Transformation***




Consider a scenario of an organization, which has a physical data source that contains entries with the attributes `surname` and `firstname`. The organization has a client application that requires entries to have a `cn` (common name) attribute of the form `firstname surname`.

The client application sends a search request for an entry of the form `cn=Joe Smith`. A transformation is defined that extracts the firstname and surname during this request and transforms the request to one of the form `surname=Smith, firstname=Joe`. The corresponding entry is located in the data source. Before returning this entry to the client application, the inverse transformation is performed. The client application receives the entry as `cn=Joe Smith`, which it understands.

This request is transformed to be of the form `surname=Smith, firstname=Joe`.

#### 11.6.1.2 Implementing Transformation in Oracle Unified Directory

Oracle Unified Directory is an LDAP server that supports transformations in a proxy server.

To implement transformations, you need to:

- Create an instance of a workflow element of type **transformations**.
- Insert the transformation workflow element in the desired workflow elements list.

For more information about the architecture of Oracle Unified Directory, see Section 4.2, "Architecture of Oracle Unified Directory."

A transformation workflow element instance is essentially a data view on which certain transformation actions are defined.

### 11.6.2 Components of Transformation

This section describes the components for configuring the workflow elements for transformation. It contains the following topics:



#### 11.6.2.1 Transformation Types

You can configure the workflow element of type transformations with the following set of transformations:

**Note:** Here:

- **Client side:** Refers to the side where the Oracle Unified Directory server interacts with the client application.
- **Source side:** Refers to the side where the Oracle Unified Directory server interacts as a data server with its local data source, or as a proxy server with a remote server.
- **Inbound direction:** Refers to the direction where transformations are applied from the client to the source.
- **Outbound direction:** Refers to the direction where transformations are applied from the source to the client.

This section contains the following topics:


- Section 11.6.2.1.1, "`addOutboundAttribute` Transformation Type"
- Section 11.6.2.1.2, "`filterOutboundAttribute` Transformation Type"
- Section 11.6.2.1.3, "`addInboundAttribute` Transformation Type"
- Section 11.6.2.1.4, "`filterInboundAttribute` Transformation Type"
- Section 11.6.2.1.5, "`mapAttribute` Transformation Type"


#### 11.6.2.1.1 `addOutboundAttribute` Transformation Type

This transformation adds a virtual attribute or value(s) to entries returned to the client during a SEARCH operation, when the list of attributes in the request is either undefined (all) or when it contains this attribute.

When you cannot determine if an entry already contains a virtual attribute, the conflict-behavior parameter decides which of the following policy will apply:

- The virtual value is not added
- The virtual value is added and merged with the existing values
- The virtual value replaces the existing one

If you are aware that the virtual attribute is searchable in the source repository, which implies some entries in the source repository contain the virtual attribute and searches are optimized on this attribute, and if the flag `virtual-in-source` is set then the transformation process forwards the virtual attribute to the source repository in the `SEARCH REQUEST` filter. Usually, the virtual attribute is not forwarded to the source repository. When it is set to `FALSE`, search requests are optimized for common cases, which implies virtual attributes not expected to be in the source repository.

**Note:** You must keep in mind that the source schema check is applied when the virtual attribute is expected to appear in ADD or MODIFY requests. Therefore, it is recommended to configure the schema of the source to accept the virtual attribute. Otherwise, disable schema checking.

Table 11–1 describes the parameters of `addOutboundAttribute` transformation type.

*Table 11–1 Parameters of addOutboundAttribute Transformation Type*

| Parameter | dsconfig CLI | Multi (M) / Single (S) Valued | Optional (O) / Mandatory (M) | Values |
|---|---|---|---|---|
| Name of the client virtual attribute and the value definitions of the client virtual attribute | `client-attribute` | **S** | **M** | string<br>For more information, see Section 11.6.2.3, "Defining Attribute Values for Transformation."<br>For example, `displayName=%cn%` publishes the attribute `displayName` with value of `cn`. |
| Conflict behavior policy | `conflict-behavior` | **S** | **O** [default=`merge-real-and-virtual`] | `merge-real-and-virtual`<br>`real-overrides-virtual`<br>`virtual-overrides-real` |
| Virtual in source policy | `virtual-in-source` | **S** | **O** [default = `FALSE`] | `TRUE, FALSE` |
| Condition based on a filter that the entry must match | `entry-match-filter` | **S** | **O** [default = apply to all entries processed by the workflow element] | LDAP filter |
| Condition based on DN that must be an ascendant | `entry-parent-suffix` | **M** | **O** [default = apply to all requests processed by the workflow element] | DN |
| Condition to exclude operations in the operation processing | `excluded-operation` | **M** | **O** [default = apply to all LDAP operations] | enumerated (ADD, MODIFY...) |

#### 11.6.2.1.2 `filterOutboundAttribute` Transformation Type

This transformation removes an attribute or value(s) from entries received from the source before sending to the client.

Table 11–2 describes the parameters of `filterOutboundAttribute` transformation type.

*Table 11–2 Parameters of FilterOutboundAttribute Transformation Type*

| Parameter | dsconfig CLI | Multi (M) / Single (S) Valued | Optional (O) / Mandatory (M) | Values |
|---|---|---|---|---|
| Name of the source virtual attribute and the value definitions of the client virtual attribute | `source-attribute` | **S** | **M** | string<br>For more information, see Section 11.6.2.3, "Defining Attribute Values for Transformation."<br>For example, `certificate=verisign` filters the `verisign` value from the `certificate` attribute. |
| Condition based on a filter that the entry must match | `entry-match-filter` | **S** | **O** [default = apply to all entries processed by the worflow element] | LDAP filter |
| Condition based on DN that must be an ascendant | `entry-parent-suffix` | **M** | **O** [default = apply to all requests processed by the workflow element] | DN |
| Condition to exclude operations in the operation processing | `excluded-operation` | **M** | **O** [default = apply to all LDAP operations] | enumerated (ADD, MODIFY...) |

#### 11.6.2.1.3 `addInboundAttribute` Transformation Type

This transformation adds a virtual attribute or value(s) to entries received from the client while performing the ADD operation before forwarding the data to the source.

When you cannot determine if an entry already contains a virtual attribute, the conflict-behavior parameter decides which of the following policy will apply:

- The virtual value is not added
- The virtual value is added and merged with the existing values
- The virtual value replaces the existing one

Table 11–3 describes the parameters of `addInboundAttribute` transformation type.

*Table 11–3 Parameters of addInboundAttribute Transformation Type*

| Parameter | dsconfig CLI | Multi (M) / Single (S) Valued | Optional (O) / Mandatory (M) | Values |
|---|---|---|---|---|
| Name of the source virtual attribute and the value definitions of the source virtual attribute | `source-attribute` | **S** | **M** | string<br>For more information, see Section 11.6.2.3, "Defining Attribute Values for Transformation."<br>For example, `email={%cn%.%sn%@mycompany.com}` writes the attribute `email` with value derived from attributes `cn` and `sn`. |
| Conflict behavior policy | `conflict-behavior` | **S** | **O** [default=`merge-real-and-virtual`] | `merge-real-and-virtual`<br>`real-overrides-virtual`<br>`virtual-overrides-real` |
| Condition based on a filter that the entry must match | `entry-match-filter` | **S** | **O** [default = apply to all entries processed by the worflow element] | LDAP filter |
| Condition based on DN that must be an ascendant | `entry-parent-suffix` | **M** | **O** [default = apply to all requests processed by the workflow element] | DN |
| Condition to exclude operations in the operation processing | `excluded-operation` | **M** | **O** [default = apply to all LDAP operations] | enumerated (ADD, MODIFY) |

#### 11.6.2.1.4 `filterInboundAttribute` Transformation Type

This transformation removes an attribute or value(s) from entries (and modifications) received from the client on a ADD (and MODIFY) before forwarding to the source.

Table 11–4 describes the parameters of `filterInboundAttribute` transformation type.

*Table 11–4 Parameters of FilterInboundAttribute Transformation Type*

| Parameter | dsconfig CLI | Multi (M) / Single (S) Valued | Optional (O) / Mandatory (M) | Values |
|---|---|---|---|---|
| Name of the client virtual attribute and the value definitions of the client virtual attribute | `client-attribute` | **S** | **M** | string<br>For more information, see Section 11.6.2.3, "Defining Attribute Values for Transformation."<br>For example, `certificate=verisign` filters the value `verisign` of the attribute `certificate`.<br>Similarly, `secondarylocation=%primarylocation%` filters the values of `secondarylocation` when it matches the values of `primarylocation`. |
| Condition based on a filter that the entry must match | `entry-match-filter` | **S** | **O** [default = apply to all entries processed by the worflow element] | LDAP filter |
| Condition based on DN that must be an ascendant | `entry-parent-suffix` | **M** | **O** [default = apply to all requests processed by the workflow element] | DN |
| Condition to exclude operations in the operation processing | `excluded-operation` | **M** | **O** [default = apply to all LDAP operations] | enumerated (ADD, MODIFY...) |

#### 11.6.2.1.5 `mapAttribute` Transformation Type

The transformation can rename or revalue a client attribute to one source attribute in both directions.

Table 11–5 describes the parameters of `mapAttribute` transformation type.

*Table 11–5 Parameters of mapAttribute Transformation Type*

| Parameter | dsconfig CLI | Multi (M) / Single (S) Valued | Optional (O) / Mandatory (M) | Values |
|---|---|---|---|---|
| Name of the client attribute and the value definitions of the mapping from the client virtual attribute to the source virtual attribute | `client-attribute` | **S** | **M** | string<br>For more information, see Section 11.6.2.3, "Defining Attribute Values for Transformation."<br>For example, `displayName=%cn%` publishes `displayName` attribute replacing it with the value of `cn` attribute, and writes `cn` attribute replacing it with the value of `displayName` attribute. |
| Virtual in source policy | `virtual-in-source` | **S** | **O** [default = `FALSE`] | `TRUE, FALSE` |
| Conflict behavior policy | `conflict-behavior` | **S** | **O** [default=`merge-real-and-virtual`] | `merge-real-and-virtual`<br>`real-overrides-virtual`<br>`virtual-overrides-real` |
| Condition based on a filter that the entry must match | `entry-match-filter` | **S** | **O** [default = apply to all entries processed by the worflow element] | LDAP filter |
| Condition based on DN that must be an ascendant | `entry-parent-suffix` | **M** | **O** [default = apply to all requests processed by the workflow element] | DN |
| Condition to exclude operations in the operation processing | `excluded-operation` | **M** | **O** [default = apply to all LDAP operations] | enumerated (ADD, MODIFY,...) |

### 11.6.2.2 Transformation Conditions

You can configure the **Transformations** workflow element with a set of conditions. Conditions are properties (attributes) that can be set either on a `transformations-workflow-element` or on an individual

`transformation`. Transformation works only when LDAP request matches all conditions and all conditions set at the level of workflow element.

The following conditions are applicable for implementing transformation:

- Conditions can be configured to rules whether transformations apply or not.
- Conditions can be set on the transformations-workflow-element. In this situation, conditions apply for all transformation set on the workflow-element and are evaluated prior to eventually processing each transformation.
- Conditions can also be set on each individual transformation and are evaluated prior to eventually processing this transformation.

In this sense, conditions can be broadly categorized as follows:

- Section 11.6.2.2.1, "Parent Suffix"
- Section 11.6.2.2.2, "Entry Match Filter"
- Section 11.6.2.2.3, "Excluded LDAP Operation"

#### 11.6.2.2.1 Parent Suffix

This condition is applicable for transformations applied only for LDAP operations that target an entry for which name is under one of the parent suffixes specified.

When no condition of this type is configured, then transformation applies to all entries processed.

#### 11.6.2.2.2 Entry Match Filter

This condition is applicable for transformations applied on LDAP operations only for entries that match the provided filter.

When no condition of this type is configured, then transformation applies to all entries processed.

#### 11.6.2.2.3 Excluded LDAP Operation

This condition specifies a list of multi-valued attributes, where each attribute is an LDAP operation that should *not* be impacted by the transformation. It allows you to disable the action of the transformation (when it has one) on each LDAP protocol message.

When no condition of this type is configured, then transformation applies to all LDAP operations normally impacted by this type of transformation.

### 11.6.2.3 Defining Attribute Values for Transformation

An attribute value allows you to define the value of a virtual attribute during transformation. This value can either be a default value, or rule that creates the value from other attribute values.

For `addInboundAttribute`, `addOutboundAttribute`, and `mapAttribute`, you must configure the values of the virtual attribute added. For `filterInboundAttribute` and `filterOutboundAttribute`, the values you intend to filter *may* be configured.

An attribute can derive its value from the following:

- Section 11.6.2.3.1, "Constant"
- Section 11.6.2.3.2, "Value of Another Attribute"

- Section 11.6.2.3.3, "Regular Expressions"
- Section 11.6.2.3.4, "Values Mapping"
- Section 11.6.2.3.5, "Multi-valued Virtual Attributes"

#### 11.6.2.3.1 Constant

It is used to generate an attribute with a static default value or to filter a static value of an attribute.

For example, the property `source-attribute:mycompany=Acme` is used to provide a default company name.

```
dsconfig --create-transformation \
--type add-inbound-attribute \
--set source-attribute:mycompany=Acme \
--transformation-name virtDeptName \
```

#### 11.6.2.3.2 Value of Another Attribute

It is used to create a new attribute from an existing attribute in the entry that is being processed or to filter a value taken from another attribute using the `%inputAttrName%` syntax.

For example, the property `source-attribute:displayName=%cn%` specifies that the value of the new attribute must be taken from the value of the `cn` attribute.

```
dsconfig --create-transformation \
--type add-inbound-attribute \
--set source-attribute:displayName=%cn% \
--transformation-name virtDeptName \
```

> **Note:** You must keep in mind that another virtual attribute generated in the same `transformations-workflow-element` should not be referenced, because the evaluation order is not guaranteed.

#### 11.6.2.3.3 Regular Expressions

It is used to create an attribute value or to filter an attribute value by manipulating the value of an existing attribute using the `{expression}` syntax.

For example, the property `client-attribute:mail={%cn%.%sn%@mycompany.com}` is a regular expression that is used for deriving an attribute by combining the values of existing attributes.

```
dsconfig create-transformation \
--type add-outbound-attribute \
--set client-attribute:mail={%cn%.%sn%@mycompany.com} \
--transformation-name virtDeptName \
```

#### 11.6.2.3.4 Values Mapping

It is used for defining virtual values as a mapping of values of another attribute using the `virtAttrName=%refAttrName%(virtValue1,refValue1)(virtValue2,refValue2)` syntax.

For the `virtAttrName` parameter, the transformation adds or filters values extracted from `refAttrName`. If `refAttrName` matches `refValue1`, then transformation processes either add or filter for `virtValue1`. In the values provided, characters `'(', ')', ','` and `'\'`must be escaped using `'\'` character.

For example, consider an organization with several departments where department name is returned for the retrieved department ID, such as Department:1–Marketing, 2–Sales, 3–Finance and so on. But, when `deptId` is `1`, the value returned for `deptName` is `Marketing`. When `deptId` is `2`, the value for `deptName` is `Sales`. Similarly, when `deptId` is `3`, the value returned for `deptName` is `Finance`.

```
dsconfig --create-transformation \
--type add-outbound-attribute \
--set client-attribute:deptName=%deptId%(Marketing,1)(Sales,2)(Finance,3) \
--transformation-name virtDeptName
```

#### 11.6.2.3.5 Multi-valued Virtual Attributes

It is used to specify a virtual multi-valued attribute using the `virtAttrName=virtAttrValue1=virtAttrValue2=` syntax.

```
dsconfig --create-transformation \
--type add-outbound-attribute \
--set client-attribute:countriesResp=France=Germany=Italy \
--transformation-name virtCountriesRep
```

## 11.6.3 Configuring Transformation

This section describes how to configure the transformation workflow element, and contains the following topics:

- Section 11.6.3.1, "Overview of the Configuration Model"
- Section 11.6.3.2, "Example of Configuring Transformation Using CLI"

### 11.6.3.1 Overview of the Configuration Model

The `transformations-workflow-element` and `transformations` are the backbone entities for configuring transformation.

The **Transformations** workflow element is a container that contains a list of references to transformations. One transformation can be reused by multiple `transformation-workflow-elements`. Conditions are properties (attributes) that can be set either on a `transformations-workflow-element` or on a `transformation`.

Table 11–6 describes the parameters that you can configure for a `transformations-workflow-element`.

*Table 11–6 Parameters to Configure for a transformations-workflow-element*

| Parameter | dsconfig CLI | Multi (M) / Single (S) Valued | Optional (O) / Mandatory (M) | Values |
|---|---|---|---|---|
| List of transformation types<br>See Section 11.6.2.1, "Transformation Types" for more information. | transformation object (association) | **M** | **M** | Reference to a transformation |
| Condition based on a filter that the entry must match<br>See Section 11.6.2.2.2, "Entry Match Filter" for more information. | `entry-match-filter` property | **S** | **O** [default = apply to all entries processed by the worflow element] | LDAP filter |
| Condition based on DN that must be an ascendant<br>See Section 11.6.2.2.1, "Parent Suffix" for more information. | `entry-parent-suffix` property | **M** | **O** [default = apply to all requests processed by the workflow element] | DN |
| Condition to exclude operations in the operation processing<br>Section 11.6.2.2.3, "Excluded LDAP Operation" for more information. | `excluded-operation` property | **M** | **O** [default = apply to all LDAP operations] | enumerated (ADD, MODIFY...) |

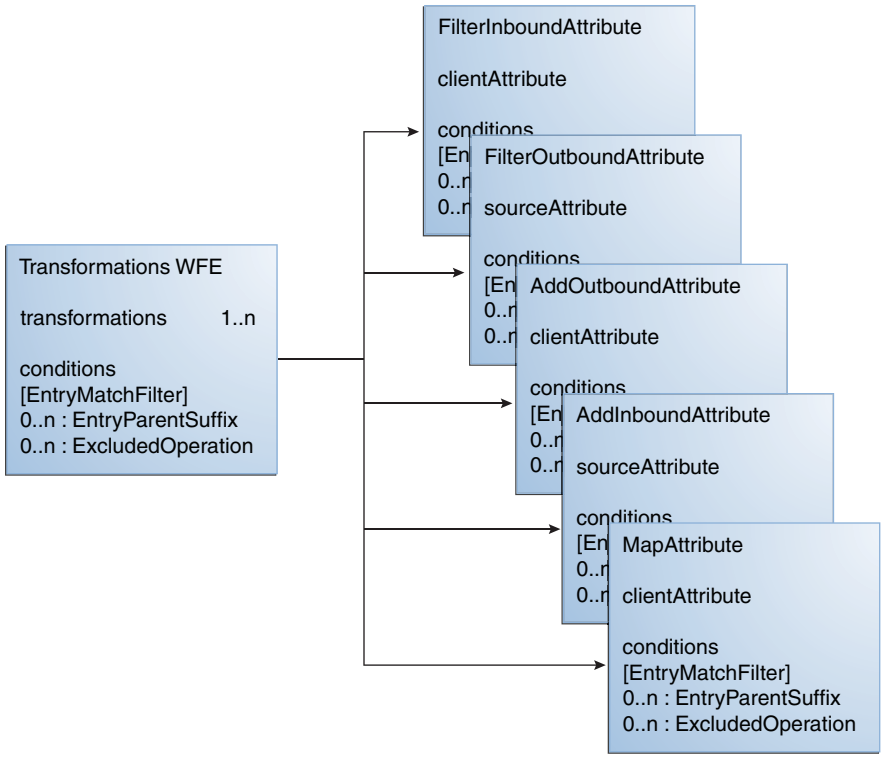
You cannot configure the order in which the transformations should work. For instance, you define a `transformation-workflow-element` that uses transformation A and transformation B. But, you cannot determine if an entry is first processed by transformation A and then by transformation B. It can be B before A.

If you need to define the order in which transformation should occur, for example transformation A should happen before transformation B, then it is recommended that

you first create a `transformation-workflow-element` that uses transformation A. Next, create another `transformation-workflow-element` that uses transformation B. Now, place the second `transformation-workflow-element` after the first `transformation-workflow-element`.

Figure 11–17 illustrates a high-level configuration model.

***Figure 11–17 Configuration Model***




### 11.6.3.2 Example of Configuring Transformation Using CLI

The example in this section describes how to create transformations with dsconfig CLI, create a transformation workflow element, add transformations, and associate conditions.

Perform the following steps to configure transformation:

1. Create a first transformation of type `filter-outbound-attribute`.

```
$ dsconfig --create-transformation -X -n -Q -p -D cn="directory manager" -j
pwd-file \
--set source-attribute:description \
--type filter-outbound-attribute\
--transformation-name fodescription
```

2. Create another transformation of type `add-outbound-attribute`.

```
$ dsconfig --create-transformation -X -n -Q -p -D cn="directory manager" -j
pwd-file  \
--set client-attribute:legacyemail=%cn%.%sn%@mycompany.com \
--type add-outbound-attribute \
--transformation-name legacyemail
```

3. Create the `transformations-workflow-element` with the first transformation, and add it to the processing flow.

```
$ dsconfig --create-workflow-element -X -n -Q -p -D cn="directory manager" -j
pwd-file \
--set transformation:legacyemail \
--set set next-workflow-element:pxywfe \
--type transformations \
--element-name trsfwfe

$ dsconfig --set-workflow-prop -X -n -Q -p -D cn="directory manager" -j
pwd-file \
--workflow-name pxywf \
--set workflow-element:trsfwfe
```

4. Add the second transformation to the workflow element.

```
$ dsconfig --set-workflow-element-prop -X -n -Q -p -D cn="directory manager" -j
pwd-file \
--element-name trsfwfe \
--add transformation:fodescription
```

5. Define the transformation criteria, which is that the transformation will occur only under `cn=users.`

```
$ dsconfig --set-workflow-element-prop -X -n -Q -p -D cn="directory manager" -j
pwd-file \
--element-name trsfwfe \
--set entry-parent-suffix:cn=users,dc=oracle
```

6. Set that transformations will happen only for users located in Paris.

```
$ dsconfig --set-workflow-element-prop -X -n -Q -p -D cn="directory manager" -j
pwd-file \
--element-name trsfwfe \
--set entry-match-filter:l=Paris
```

7. Create a new mapping transformation and add it to the workflow element.

```
$ dsconfig --create-transformation -X -n -Q -p -D cn="directory manager" -j
pwd-file  \
--set client-attribute:faxnum=%facsimileTelephoneNumber% \
--type map-attribute \
--transformation-name mapfax

$ dsconfig --set-workflow-element-prop -X -n -Q -p -D cn="directory manager" -j
pwd-file \
--element-name trsfwfe \
--add transformation:mapfax
```

8. Set that this transformation will happen only for persons.

```
$ dsconfig --set-transformation-prop -X -n -Q -p -D cn="directory manager" -j
pwd-file \
--transformation-name mapfax \
--set entry-match-filter:\(objectclass=person\)
```

# 12

# Understanding Oracle Unified Directory Mapping

This chapter describes Oracle Unified Directory mapping and includes the following topics:



## 12.1 An Overview of Identity Mappers

Identity Mappers are responsible for establishing a mapping between an identifier string provided by a client, and the entry for the user that corresponds to that identifier. Identity Mappers are used to process several SASL mechanisms to map an authentication ID (for instance, a Kerberos principal when using GSSAPI) to a directory user. They are also used when processing requests with the proxied authorization control.

Oracle Unified Directory supports multiple SASL identity mappers. For example, you can define Identity Mapper1 for a user `xyz` and Identity Mapper2 for the remaining users. This is beneficial when using GSSAPI where users with different domains, such as `@example.com` and `@oracle.com` require different identity mappers.

Oracle Unified Directory also provides support for an identifier string that is a bind ID and not a DN. However, this is applicable for simple binds only. The key idea is that a client should be able to specify any attribute in the simple bind that is allowed by the corresponding Identity Mapper. Consider the following examples:

```
ldapsearch -D "user@example.com" -w password -b "" objectclass=*
```

In this example, bind ID is the e-mail ID of the user.

## 12.2 Supported Identity Mappers

The following Identity Mappers are available in the server:

### 12.2.1 Exact Match Identity Mapper

The Exact Match Identity Mapper maps an identifier string to a user entry by searching for the entry containing a specified attribute whose value is the provided identifier. For example, the user name provided by the client for DIGEST-MD5 authentication must match the value of the uid attribute. Note that this attribute must be specified in the identity mapper configuration.

This is primarily used in simple binds and all SASL binds except GSSAPI.

### 12.2.2 Match And Replace Identity Mapper

The Match And Replace Identity Mapper provides a way to use a regular expression to translate the provided identifier when searching for the appropriate user entry.

This may be used, for example, if the provided identifier is expected to be an e-mail address or Kerberos principal, but only the user name (the part preceding the `@` symbol) should be used in the mapping process. Note that a replacement is made only if all or part of the provided ID string matches the given match pattern. If no part of the ID string matches the provided pattern, the given ID string is used without any alteration.

This is primarily used in GSSAPI binds.

## 12.3 Components of Identity Mappers

The following components have a direct aggregation relation to Identity Mappers:

- Section 12.3.1, "Global Configuration"
- Section 12.3.2, "Network Group"

### 12.3.1 Global Configuration

The Global Configuration contains properties that affect the overall operation of the Oracle Unified Directory.

### 12.3.2 Network Group

The Network Group is used to classify incoming client connections and route requests to workflows.

## 12.4 Configuring Identity Mappers

Identity Mappers are configured at the following instances:

- Network Group
- Global Configuration

To summarize, each Network Group has one or more Identity and Certificate mappers, which are used to map identities specific to that network group. If an identity or certificate mapper is not defined at the network-group level, then a global identity mapper is used as the default setting.

This section contains the following topics:

- Section 12.4.1, "Configuring Global Identity Mappers"
- Section 12.4.2, "Configuring Network Group Identity Mappers"

### 12.4.1 Configuring Global Identity Mappers

Identity mappers are configured by default at the global level. However, if you want to configure an identity mapper globally, then run the following command:

```
dsconfig set-global-configuration-prop --add "generic-identity-mapper:Exact Match"
```

The preceding command is based on the assumption that the `Exact Match` identity mapper already exists. This identity mapper is provided by default in the configuration.

### 12.4.2 Configuring Network Group Identity Mappers

For an existing default network group called `network-group` configure the generic-identity-mapper as follows:

```
dsconfig set-network-group-prop --group-name network-group --set
"generic-identity-mapper:Exact Match"
```

The preceding command is based on the assumption that the `Exact Match` identity mapper already exists. This identity mapper is provided by default in the configuration.

## 12.5 Selecting Identity Mappers

Normally, one identity mapper is defined per network group. The generic-identity-mapper defines an identity mapper that applies to all but GSSAPI binds. The gssapi-identity-mapper defines the one that applies to GSSAPI binds only.

As described earlier, the exact match and match and replace identity mappers are generally used as generic-identity-mapper and gssapi-identity-mapper respectively. However, you can select a different combination based on your requirement.

## 12.6 Ordering Identity Mappers

An identity mapper is selected based on the regex pattern; therefore there is a possibility that a conflict might arise when multiple identity mappers are defined. So, it becomes imperative to define the order in which identity mappers are evaluated in the network group.

You can define priorities for the conflicting identity mappers to resolve this conflict. If a conflict arises, the identity mapper with the lowest priority is selected and used for mapping. If identity mappers have equal priority, then the behavior is undefined.

Run the following command to define priority:

```
dsconfig -h hostname -p admin_port -D USER set-identity-mapper-prop --mapper-name
"Exact Match" --set "priority:2"
```

A lower priority value implies higher priority. Priority for network groups is also determined in a similar fashion.

# Part III

## Basic Administration

This part describes how to start and stop the server and how to configure the various server elements, depending on the required deployment scenario.

This part includes the following chapters:

# 13

# Starting and Stopping the Server

This chapter describes the basic procedures to start and stop a server instance. The procedures described in this chapter apply to an Oracle Unified Directory directory server, proxy server, and replication gateway instance.

This chapter includes the following topics:

- Section 13.1, "Starting the Server"
- Section 13.2, "Stopping the Server"
- Section 13.3, "Checking if the Server is Started or Stopped"
- Section 13.4, "Running the Server as a Non-Root User"

## 13.1 Starting the Server

To start the server, run the `start-ds` command on UNIX or Linux systems or the `start-ds.bat` command on Windows systems. By default, the `start-ds` command starts the server as a background process when no options are specified. You can use the `start-ds` command with the `--nodetach` option to run the server as a foreground process. For more information, see Appendix A.2.15, "start-ds."

The `start-ds` command automatically attempts to find the correct Java environment to use when starting the server. You can specify the path to the Java installation, and provide additional options directly to the JVM when the directory server is starting. For more information, see "Configuring the Default JVM and Java Arguments" in *Oracle Fusion Middleware Installation Guide for Oracle Unified Directory.*

### 13.1.1 To Start the Server by Using `start-ds`

1. Change to the appropriate directory.

```
(UNIX, Linux)  $ cd INSTANCE_DIR/OUD/bin
(Windows)      C:\> cd INSTANCE_DIR\OUD\bat
```

2. Type `start-ds`.

```
(UNIX, Linux)  $ start-ds
(Windows)      C:\> start-ds
```

### 13.1.2 To Start the Server as a Foreground Process

1. Change to the appropriate directory.

```
(UNIX, Linux)  $ cd INSTANCE_DIR/OUD/bin
(Windows)      C:\> cd INSTANCE_DIR\OUD\bat
```

2. Type `start-ds` with `-N` or `--nodetach`.

```
(UNIX, Linux)  $ start-ds --nodetach
(Windows)      C:\> start-ds --nodetach
```

You can stop the directory server by pressing `Control-C` in the terminal window in which the server is running or by running the `stop-ds` utility from another window.

### 13.1.3 To Restart the Server

1. Change to the installation directory.

```
(UNIX, Linux)  $ cd INSTANCE_DIR/OUD/bin
(Windows)      C:\> cd INSTANCE_DIR\OUD\bat
```

2. Type `stop-ds` with `-R` or `--restart`.

```
(UNIX, Linux)  $ stop-ds --restart
(Windows)      C:\> stop-ds --restart
```

### 13.1.4 To Start the Server by Using a Script (UNIX/Linux)

The `start-ds` command provides a "quiet" option (`-Q` or `--quiet`) that suppresses output during the startup process unless a significant error occurs. You can use this option in a startup script.

1. Create a shell script and add the following `start-ds` command.

```
INSTANCE_DIR/OUD/bin/start-ds --quiet
```

2. Run the script.

## 13.2 Stopping the Server

On any system (whether the server is running in the foreground or the background), or even from a remote system, you can stop the server using one of the following methods. For more information about the `stop-ds` command, see Appendix A.2.17, "stop-ds."

### 13.2.1 To Stop the Server by Using `stop-ds`

1. Change to the appropriate directory.

```
(UNIX, Linux)  $ cd INSTANCE_DIR/OUD/bin
```

```
(Windows)         C:\> cd INSTANCE_DIR\OUD\bat
```

2. Type `stop-ds`

```
(UNIX, Linux)  $ stop-ds
(Windows)      C:\> stop-ds
```

### 13.2.2 To Stop the Server that is Running in the Foreground

This procedure assumes that the directory server is running as a foreground process (using the `-N` or `--nodetach` option).

1. Type Control-C in a terminal window on UNIX or in the Command Prompt window on Windows systems to stop the server.

   Alternatively, run the `stop-ds` command from another window.

### 13.2.3 To Stop the Server by Using a Script (UNIX/Linux)

The `stop-ds` command provides a "quiet" option (`-Q` or `--quiet`) that suppresses output during the stopping process unless a significant error occurs. You can use this option in a shutdown script.

1. Create a shell script and add the following `stop-ds` command.

   *INSTANCE_DIR*`/OUD/bin/stop-ds --quiet`

2. Run the script.

## 13.3 Checking if the Server is Started or Stopped

At any time, you can check if the server is started or stopped by using the `status` command.

### 13.3.1 To Check the Server Status

1. Change to the appropriate directory.

```
(UNIX, Linux)  $ cd INSTANCE_DIR/OUD/bin
(Windows)      C:\> cd INSTANCE_DIR\OUD\bat
```

2. Type `status`

```
(UNIX, Linux)  $ status
(Windows)      C:\> status
```

## 13.4 Running the Server as a Non-Root User

Like many network daemons, Oracle Directory Server Enterprise Edition has a `setuid` capability that allows it to be started as a root user but then drop privileges to run as a user with fewer capabilities. Oracle Unified Directory does not currently include this capability. However, you can install, start, and run the server as a

non-root user. Note that the information in this section applies primarily to UNIX-based platforms, because Windows systems do not historically place as many restrictions on non-administrative users.

### 13.4.1 Reasons for Running the Server as a Non-Root User

In many cases, running the server as a non-root user from the start is a more attractive option and provides greater functionality than the `setuid` equivalent. Running the server as a non-root user means that administrators do not need root access to the system, which is often desirable from an operational perspective. In addition, more administrative actions can be performed with the server online, because the server can do things that might not have been available after it had dropped root privileges.

The primary reason that servers are typically started and/or run as root users is so that they can listen on a privileged port (namely, ports between 1 and 1024). The standard port for LDAP communication is port 389, and the standard port for LDAPS is 636. On most UNIX-based systems only root users are able to create processes that listen on these ports. There can be other reasons for starting as a root user (for example, the ability to use a larger number of file descriptors), but it is generally easier to configure around these other limitations.

Although the standard LDAP and LDAPS ports are 389 and 636, the server is not required to run on those ports. In some environments, it is common to run the server on ports above 1024 (such as 1389 and 1636) so that it is not necessary to be root to start it. Virtually all LDAP-enabled clients provide the ability to specify the port on which the server is listening. As long as the clients know what port the server is using, any value is allowed. For information about configuring the listen port, see Section 14.1.5.2, "Configuring the LDAP Connection Handler."

### 13.4.2 How to Run as a Non-Root User on the Standard LDAP Ports

If clients expect the server to be listening on port 389 or 636, other options are still available. The best option, available on Solaris systems from Solaris 10 onwards, is to use the process rights management subsystem (also called *least privilege*). The privileges subsystem in Solaris makes it possible to give non-root users and roles capabilities normally available only to the root user (much like the Privilege Subsystem allows within the server). In particular, the `net_privaddr` privilege controls which users can bind to privileged ports If this privilege is granted to a non-root user, that user can bind to privileged ports. To configure a user with this privilege, run the following command, as the root user:

```
# usermod -K defaultpriv=basic,net_privaddr,sys_resource,-proc_info,-file_link_any
oud
```

This command configures the `oud` user so that it starts with the `basic` privilege set (which is what non-root users have by default). The command then adds the `net_privaddr` and `sys_resource` privileges, which allow the user to increase the number of file descriptors available, among other things. The command removes the `proc_info` privilege (which allows the user to see processes owned by other users) and the `file_link_any` privilege (which allows the user to create hard links to files that they do not own). After running this command, the `oud` user is able to start the server listening on a privileged port.

Even on systems without a capability like least privilege, it is possible to expose the server on a privileged port such as 389 or 636 without requiring root privileges to be able to start it. One possibility would be to run the server on an unprivileged port and use a directory proxy server listening on the privileged port to forward communication to the server on an unprivileged port. It is also possible to use network

hardware to achieve the same purpose or to use firewall rules on the same system. For example, on Linux systems the following commands can be used to redirect traffic targeting port 389 to port 1389:

```
# iptables --append PREROUTING --table nat --protocol tcp --dport 389 \
  --jump REDIRECT --to-port 1389
# iptables -t nat -A OUTPUT -p tcp --dport 389 -j DNAT --to :1389
```

# 14

# Configuring the Server Instance

The easiest way to access the server configuration is by using the `dsconfig` command or, for certain aspects of the configuration, by using Oracle Directory Services Manager.

This chapter covers the following topics:

- Section 14.1, "Managing the Server Configuration With `dsconfig`"
- Section 14.2, "Managing the Server Configuration With Oracle Directory Services Manager"
- Section 14.3, "Managing Administration Traffic to the Server"
- Section 14.4, "Configuring Commands As Tasks"
- Section 14.5, "Deploying and Configuring the DSML Gateway"

## 14.1 Managing the Server Configuration With `dsconfig`

The topics in this section are intended for administrators or users who want to configure and manage a deployed Oracle Unified Directory instance. These topics provide an overview of the `dsconfig` command-line utility and its use in server configuration.

You can use the `dsconfig` command to configure both the Oracle Unified Directory directory server and the proxy server. For a list of the supported sub-commands for the directory server or proxy instance, and for specific information about this command, see Appendix A.2.4, "dsconfig."

You can also use `dsconfig` to configure a number of proxy-specific components. This section contains the following topics:

- Section 14.1.1, "Overview of the `dsconfig` Command"
- Section 14.1.2, "Using `dsconfig` in Interactive Mode"
- Section 14.1.3, "Getting Help With `dsconfig`"
- Section 14.1.4, "Configuring a Server Instance With `dsconfig`"
- Section 14.1.5, "Configuring Connection Handlers With `dsconfig`"
- Section 14.1.6, "Configuring Network Groups With `dsconfig`"
- Section 14.1.7, "Configuring Workflows With `dsconfig`"
- Section 14.1.8, "Configuring Workflow Elements With `dsconfig`"
- Section 14.1.9, "Configuring Plug-Ins With `dsconfig`"

- Section 14.1.10, "Configuring Suffixes with dsconfig."

## 14.1.1 Overview of the dsconfig Command

The dsconfig command provides a simple mechanism for accessing the server configuration. dsconfig presents the configuration as a set of components, each of which can be managed through one or more sub-commands.

dsconfig can also be used interactively. In interactive mode, dsconfig functions much like a wizard, walking you through the server configuration. For more information, see Section 14.1.2, "Using dsconfig in Interactive Mode."

dsconfig can only be used to configure a *running* server instance. Offline configuration is not supported by dsconfig.

Like the other administration commands, dsconfig uses the administration connector to access the server. For more information, see Section 14.3, "Managing Administration Traffic to the Server." All of the examples in this section assume that the administration connector is listening on the default port (4444) and that the command is accessing the server running on the local host. If this is not the case, the --port and --hostname options must be specified.

This section contains the following topics:

- Section 14.1.1.1, "dsconfig and Certificate Checking"
- Section 14.1.1.2, "dsconfig Sub-Commands"
- Section 14.1.1.3, "dsconfig Advanced Properties"

### 14.1.1.1 dsconfig and Certificate Checking

dsconfig accesses the server over a secured connection with certificate authentication. If you run dsconfig in interactive mode, you are prompted as to how you want to trust the certificate.

If you run dsconfig in non-interactive mode (that is, with the -n option), specification of the trust store parameters depends on whether you run the command locally or remotely.

- **Running dsconfig locally.** (The command is launched on the server that you are administering.) If you do not specify the trust store parameters, the server uses the local instance trust store by default. Unless you specify otherwise, the local instance trust is INSTANCE_DIR/OUD/config/admin-truststore.
- **Running dsconfig remotely.** (The command is launched on a different server to the one you are administering.) You *must* specify the trust store parameters or the -X (--trustAll) option. The easiest way to specify the trust store parameters is to run the command once in interactive mode and to save the certificate that is presented by the server in your trust store.

```
$ dsconfig

>>>> >>>> Specify Oracle Unified Directory LDAP connection parameters

Directory server hostname or IP address [host1.example.com]:
Directory server administration port number [4444]:
Administrator user bind DN [cn=Directory Manager]:
Password for user 'cn=Directory Manager':

How do you want to trust the server certificate?
```

```
    1)  Automatically trust
    2)  Use a truststore
    3)  Manually validate

Enter choice [3]: 3

Administrator user bind DN [cn=Directory Manager]:

Password for user 'cn=Directory Manager':

Server Certificate:

User DN  : CN=host1.example.com, O=Administration Connector Self-Signed
Certificate
Validity : From 'Wed Apr 29 11:13:21 MEST 2009'
             To 'Fri Apr 29 11:13:21 MEST 2011'
Issuer   : CN=host1.example.com, O=Administration Connector Self-Signed
Certificate

Do you trust this server certificate?

    1)  No
    2)  Yes, for this session only
    3)  Yes, also add it to a truststore
    4)  View certificate details

Enter choice [2]: 3

Truststore path: /local/instances/certificates/jctruststore

Password for keystore '/local/instances/certificates/jctruststore':

...
```

When you have saved the certificate in the trust store, you can specify those trust store parameters in non-interactive mode.

```
$ dsconfig -h localhost -p 4444 list-connection-handlers -n \
--trustStorePath /local/instances/certificates/jctruststore \
--trustStorePasswordFile /local/instances/certificates/jctruststore.pin -j
pwd-file
Connection Handler       : Type : enabled : listen-port : use-ssl
-------------------------:------:---------:-------------:--------
JMX Connection Handler   : jmx  : false   : 1689        : false
LDAP Connection Handler  : ldap : true    : 1389        : false
LDAPS Connection Handler : ldap : false   : 636         : true
LDIF Connection Handler  : ldif : false   : -
```

### 14.1.1.2 dsconfig Sub-Commands

dsconfig provides an intuitive list of sub-commands to manage various elements of the configuration.

You can use these sub-commands to add, delete, list, view, and modify different components:

| Subcommand | Function |
|---|---|
| dsconfig create-component options | Creates a new component |
| dsconfig delete-component options | Deletes an existing component |

| Subcommand | Function |
|---|---|
| `dsconfig get-component-prop options` | Displays the properties of a component |
| `dsconfig list-components options` | Lists the existing defined components |
| `dsconfig set-component-prop options` | Modifies the properties of a component |

For example, the following five sub-commands are used to manage connection handlers:

| Subcommand | Function |
|---|---|
| `dsconfig create-connection-handler options` | Creates connection handlers |
| `dsconfig delete-connection-handler options` | Deletes connection handlers |
| `dsconfig get-connection-handler-prop options` | Displays the properties of a connection handler |
| `dsconfig list-connection-handlers options` | Lists the existing defined connection handlers |
| `dsconfig set-connection-handler-prop options` | Modifies the properties of a connection handler |

Not all types of components can be created and deleted. For example, a directory server has only a single global configuration. For this reason, the global configuration is managed with only two sub-commands:

| Subcommand | Function |
|---|---|
| `dsconfig get-global-configuration-prop options` | Displays the global configuration properties |
| `dsconfig set-global-configuration-prop options` | Modifies the global configuration properties |

The configurable properties of all components can be queried and modified to change the behavior of the component. For example, an LDAP connection has properties that determine its IP listener address, its port, and its SSL configuration.

#### 14.1.1.3 `dsconfig` Advanced Properties

There are a number of component properties that are considered *advanced* properties. The advanced properties are not displayed by default, and have default values that apply in most cases. If you want to modify the values or the advanced properties, use `--advanced` before the subcommand. For example:

```
$ dsconfig --advanced get-extension-prop
```

## 14.1.2 Using dsconfig in Interactive Mode

Unless you specify all configuration parameters and the -n (--no-prompt) option, dsconfig runs in interactive mode. Interactive mode functions like a wizard, walking you through the server configuration. Interactive mode is a good approach to start using dsconfig.

When you run dsconfig in interactive mode, you can specify that you want the equivalent command (including all your selections) to be displayed, or to be written to a file. The following example shows how to use the --displayCommand option to display the equivalent non-interactive command when configuring the trust manager. Note that the equivalent command is displayed at the point at which the command has been applied and validated by the directory server.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file
--displayCommand
...
The TrustStore Manager Provider was modified successfully

The equivalent non-interactive command-line is:
dsconfig --hostname "localhost" --port "4444" --bindDN "cn=directory
manager" --bindPasswordFile pwd-file --trustAll
set-trust-manager-provider-prop --provider-name "PKCS12" --set
"enabled:true"
```

To copy the equivalent command to a file, use the --commandFilePath option, as shown in the following example:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file
--commandFilePath /tmp/filename
```

## 14.1.3 Getting Help With dsconfig

The dsconfig command has extensive online help that is accessed using the --help option. This section provides an overview, and contains the following topics:

- Section 14.1.3.1, "Global Usage"
- Section 14.1.3.2, "Finding the Correct Subcommand"
- Section 14.1.3.3, "Getting Help for an Individual Subcommand"
- Section 14.1.3.4, "Displaying a Summary of a Component's Properties"
- Section 14.1.3.5, "Displaying Detailed Help on a Property"

### 14.1.3.1 Global Usage

Use the following command to display dsconfig's global usage:

```
$ dsconfig --help
```

### 14.1.3.2 Finding the Correct Subcommand

The global usage information does not include the list of available sub-commands. To retrieve the list of sub-commands, use one of the --help-xxx options, where *xxx* determines the group of sub-commands to be displayed.

---

**Note:** Use the --help-all option used to display all of the available sub-commands.

---

For example, to find all the sub-commands relating to distribution, use the following command:

```
$ dsconfig --help-distribution
```

#### 14.1.3.3 Getting Help for an Individual Subcommand

When you have determined which subcommand you want, you can get more detailed help on that subcommand by using the subcommand `--help` option as follows:

```
$ dsconfig create-monitor-provider --help
```

#### 14.1.3.4 Displaying a Summary of a Component's Properties

The `dsconfig` command has built-in documentation for all of the components and their properties. This documentation can be accessed by using the `list-properties` subcommand. For example, a summary of the properties associated with a work queue can be displayed by using the following command:

```
$ dsconfig list-properties -c work-queue
```

If the `-c` option is not specified, a summary of the properties for all components is displayed.

#### 14.1.3.5 Displaying Detailed Help on a Property

The summary table displays only brief usage information for each property. More detailed information are available using the verbose mode of the `list-properties` subcommand:

```
$ dsconfig list-properties -c work-queue --property num-worker-threads -v
```

If the `--property` option is not specified, verbose help is provided for all the work-queue properties.

### 14.1.4 Configuring a Server Instance With `dsconfig`

The `dsconfig` command is the recommended utility for accessing the server configuration. Accessing the configuration directly over LDAP, using the `ldap*` utilities is discouraged. This section describes the utility to access the server components and contains the following topics:

- Section 14.1.4.1, "To Display the Properties of a Component"
- Section 14.1.4.2, "To List Components"
- Section 14.1.4.3, "To Create a Component"
- Section 14.1.4.4, "To Modify the Properties of a Component"
- Section 14.1.4.5, "To Modify the Values of a Multi-Valued Property"
- Section 14.1.4.6, "To Delete a Component"
- Section 14.1.4.7, "To Use `dsconfig` in Batch Mode"

#### 14.1.4.1 To Display the Properties of a Component

Each component has one or more properties that can be displayed by using the component's `get-xxx-prop` subcommand. Each component is associated with a single LDAP entry in the server configuration, and each property is associated with a single LDAP attribute.

To display the properties of the default LDAP connection handler, run the following command:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  get-connection-handler-prop --handler-name "LDAP Connection Handler"

Property : Value(s)
------------------------:-----------------------------------------------------
allow-ldap-v2 : true
allow-start-tls : false
allowed-client : -
denied-client : -
enabled : true
keep-stats : true
key-manager-provider : -
listen-address : 0.0.0.0
listen-port : 1389
ssl-cert-nickname : server-cert
ssl-cipher-suite : -
ssl-client-auth-policy : optional
ssl-protocol : -
trust-manager-provider : -
use-ssl : false
```

The dsconfig command displays the default values or behavior for properties that have not been customized.

#### 14.1.4.2 To List Components

You can view a list and summary of the instances of one component by using the component's list-*xxx*s subcommand. This can be particularly useful if you have more than one instance of the same component.

For example, to list the configured connection handlers, run this command:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  list-connection-handlers
```

Depending on your installation, the output will be similar to the following.

```
Connection Handler       : Type : enabled : listen-port : use-ssl
-------------------------:------:---------:-------------:--------
JMX Connection Handler   : jmx  : false   : 1689        : false
LDAP Connection Handler  : ldap : true    : 1389        : false
LDAPS Connection Handler : ldap : false   : 636         : true
LDIF Connection Handler  : ldif : false   : -           : -
SNMP Connection Handler  : snmp : false   : 161         : -
```

#### 14.1.4.3 To Create a Component

New instances of a component can be created by using the component's create-xxx subcommand. Often there are several *subtypes* of the component. For example, there are four types of connection handler: LDAP, LDIF, JMX, and SNMP. Because all of these are created by using the same subcommand, you must specify the type of component that you want to create. Do this by using the subcommand -t or --type.

When you create a new component, you must specify the component's mandatory properties. The mandatory properties depend on the type of component that is being created. For example, an LDAP connection handler might have different mandatory properties to a JMX connection handler. If a mandatory property is left undefined, dsconfig enters interactive mode and prompts you for the undefined properties. If

you include the -n (non-interactive) option, dsconfig fails to create the component and displays an error message indicating which properties need to be defined.

1. Display the types of connection handler that can be created by accessing the help for the connection handler component.

```
$ dsconfig create-connection-handler --help

Usage: dsconfig create-connection-handler {options}
Creates Connection Handlers

Global Options:
See "dsconfig --help"

SubCommand Options:
--handler-name {NAME}
The name of the new Connection Handler
--set {PROP:VALUE}
Assigns a value to a property where PROP is the name of the property and
VAL is the single value to be assigned. Specify the same property multiple
times in order to assign more than one value to it
-t, --type {TYPE}
The type of Connection Handler which should be created. The value for TYPE
can be one of: custom | jmx | ldap | ldif | snmp
```

2. Create a new LDAP connection handler, specifying values for the mandatory enabled and the listen-port properties.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  create-connection-handler -t ldap --handler-name "My LDAP Connection Handler"
```

An error message similar to the following will be displayed.

```
The LDAP Connection Handler could not be created because the following
mandatory properties were not defined:

Property     Syntax
---------------------------------
enabled      false | true
listen-port  1 <= INTEGER <= 65535
```

### 14.1.4.4 To Modify the Properties of a Component

The properties of a component can be modified by using the component's set-xxx-prop subcommand. Multiple properties can be modified at the same time by using multiple occurrences of the --set option. The following example uses the set-connection-handler-prop subcommand to modify the properties of a connection handler.

> **Note:** Many components have a Java class property that specifies the name of a Java class to be used as the implementation of the component. Do not modify this property, as doing so could prevent your server from operating correctly. These properties are treated as *advanced* properties and hidden from view unless you run dsconfig with the --advanced option.

For example, to configure the LDAP connection handler to accept LDAPv2 connections, run this command:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  set-connection-handler-prop --handler-name="LDAP Connection Handler" \
  --set allow-ldap-v2:true
```

#### 14.1.4.5 To Modify the Values of a Multi-Valued Property

You can set multiple values for a property by using the `--add` option multiple times in the same `dsconfig` command.

This example sets multiple values for the `allowed-client` property.

To restrict connections through the LDAP connection handler to specific clients, run these commands:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -n \
  set-connection-handler-prop --handler-name "LDAP Connection Handler" \
  --add allowed-client:myhost --add allowed-client:myhost.example \
  --add allowed-client:myhost.example.com
```

#### 14.1.4.6 To Delete a Component

Existing instances of a component can be removed using the `dsconfig delete-xxx` subcommand

The following example deletes the LDAP connection handler that was created in the previous example:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  delete-connection-handler --handler-name "My LDAP Connection Handler"
```

#### 14.1.4.7 To Use `dsconfig` in Batch Mode

The `-F` or `--batchFilePath` option of the `dsconfig` command enables you to specify a number of operations that are completed in a single command by consolidating those operations in a file. This can significantly improve performance when several `dsconfig` commands are required.

To use `dsconfig` in batch mode, complete the following steps:

1. Create a script that contains the required commands for creating a new backend that is used to store a new suffix.

   For example, the following file (named `new-backend.txt`) achieves three separate tasks:

   - creates the db-local-backend workflow element
   - adds a set of index entry limit for the uniquemember attribute (for example, how to set properties, but this step is not mandatory)
   - creates the workflow for the new suffix
   - registers the new suffix in the default network group

   ```
   create-workflow-element --element-name myBackend --type db-local-backend \
   --set enabled:true --set base-dn:cn=myexample,cn=com
   set-local-db-index-prop --element-name myBackend --index-name uniqueMember \
   --set index-entry-limit:5000
   create-workflow --workflow-name myWorkflow --set base-dn:cn=myexample,cn=com \
   --set enabled:true --set workflow-element:myBackend
   set-network-group-prop --group-name network-group --add workflow:myWorkflow
   ```

2. Run the `dsconfig` command with that file as a parameter.

   ```
   $ dsconfig -h localhost -p 4444 -D cn="directory manager" -j pwd-file \
   ```

```
-F new-backend.txt -X -n
```

## 14.1.5 Configuring Connection Handlers With dsconfig

*Connection handlers* are responsible for handling all interaction with client applications, including accepting connections, reading requests, and sending responses.

For information about configuring secure connections, see Section 20.5, "Configuring SSL and StartTLS for LDAP and JMX."

The section describes how to configure the connection handlers by using the dsconfig command, and contains the following topics:

- Section 14.1.5.1, "To Display All Connection Handlers"
- Section 14.1.5.2, "Configuring the LDAP Connection Handler"
- Section 14.1.5.3, "Configuring the LDIF Connection Handler"
- Section 14.1.5.4, "Configuring the JMX Connection Handler"

These sections provide examples on only a few aspects of the configuration. For details about all the configuration properties, use the following command:

```
$ dsconfig list-properties -c connection-handler
```

### 14.1.5.1 To Display All Connection Handlers

Oracle Unified Directory supports the following types of connection handler:

- **LDAP connection handler.** This connection handler is used to interact with clients using LDAP. It provides full support for LDAPv3 and limited support for LDAPv2.
- **LDAPS connection handler.** This connection handler is used to interact with clients using LDAP over SSL.
- **LDIF connection handler.** This connection handler is used to process changes in the server using internal operations.
- **JMX connection handler.** This connection handler allows interactions with clients using the Java Management Extensions (JMX) framework and the Remote Method Invocation (RMI) protocol.
- **SNMP.** This connection handler is used to process SNMP requests to retrieve monitoring information described by MIB 2605. The supported SNMP protocols are SNMP V1, V2c and V3.

To display all configured connection handlers, along with their basic properties, use the dsconfig list-connection-handlers command.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  list-connection-handlers

Connection Handler       : Type : enabled : listen-port : use-ssl
-------------------------:------:---------:-------------:--------
JMX Connection Handler   : jmx  : false   : 1689        : false
LDAP Connection Handler  : ldap : true    : 1389        : false
LDAPS Connection Handler : ldap : false   : 636         : true
LDIF Connection Handler  : ldif : false   : -           : -
SNMP Connection Handler  : snmp : false   : 161         : -
```

### 14.1.5.2 Configuring the LDAP Connection Handler

The following command displays the properties of the LDAP connection handler:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  get-connection-handler-prop --handler-name "LDAP Connection Handler"
```

Depending on your configuration, the output will be similar to the following.

```
Property               : Value(s)
-----------------------:------------
allow-ldap-v2          : true
allow-start-tls        : false
allowed-client         : -
denied-client          : -
enabled                : true
keep-stats             : true
key-manager-provider   : -
listen-address         : 0.0.0.0
listen-port            : 1389
ssl-cert-nickname      : server-cert
ssl-cipher-suite       : -
ssl-client-auth-policy : optional
ssl-protocol           : -
trust-manager-provider : -
use-ssl                : false
```

#### 14.1.5.2.1 To Control Which Clients Have LDAP Access to the Directory Server

You can specify a list of clients that may or may not access the directory server over LDAP. To do this, set the `allowed-client` or `denied-client` property of the LDAP connection handler. These properties take an IP address or subnetwork with subnetwork mask as values.

By default, these properties are not set and all clients are allowed access. Changes to these properties take effect immediately but do not interfere with connections that are already established.

This example permits access only to clients in the subnet mask `255.255.255.10`.

Run the `dsconfig` command as follows:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  set-connection-handler-prop --handler-name "LDAP Connection Handler" \
  --set allowed-client:255.255.255.10
```

### 14.1.5.3 Configuring the LDIF Connection Handler

The LDIF connection handler is disabled by default. This connection handler can be used to process changes in the server using internal operations. The changes to be processed are read from an LDIF file.

The following command displays the default properties of the LDIF connection handler:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -n \
  get-connection-handler-prop --handler-name "LDIF Connection Handler"
```

Depending on your installation, the output will be similar to the following.

```
Property       : Value(s)
---------------:-------------------------
allowed-client : -
denied-client  : -
enabled        : false
ldif-directory : config/auto-process-ldif
poll-interval  : 5 s
```

The ldif-directory property specifies the directory in which the LDIF files are located. The connection handler checks if there are any files in this directory, at an interval specified by the poll-interval property. The connection handler then processes the changes contained in those files as internal operations and writes the result to an output file with comments indicating the result of the processing.

#### 14.1.5.3.1 To Enable the JMX Alert Handler Through the LDIF Connection Handler

This example demonstrates how to enable the JMX alert handler through the LDIF connection handler.

1. Check the status of the JMX alert handler (disabled by default).

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  get-alert-handler-prop --handler-name "JMX Alert Handler"
```

   Depending on your installation, the output will be similar to the following.

```
Property            : Value(s)
--------------------:---------
disabled-alert-type : -
enabled             : false
enabled-alert-type  : -
```

2. Create an LDIF file in the default LDIF directory that enables the JMX alert handler.

```
$ cd ../config/
$ mkdir auto-process-ldif
$ cd auto-process-ldif/
$ cat > disable-jmx.ldif << EOM
> dn: cn=JMX Alert Handler,cn=Alert Handlers,cn=config
> changetype: modify
> replace: ds-cfg-enabled
> ds-cfg-enabled: true
> EOM
$
```

3. After a period of time longer than poll-interval, recheck the status of the JMX alert handler.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  get-alert-handler-prop --handler-name "JMX Alert Handler"

Property            : Value(s)
--------------------:---------
disabled-alert-type : -
enabled             : true
enabled-alert-type  : -
```

### 14.1.5.4 Configuring the JMX Connection Handler

The following command displays the default properties of the JMX connection handler:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  get-connection-handler-prop --handler-name "JMX Connection Handler"
```

Depending on your installation, the output will be similar to the following.

```
Property             : Value(s)
```

```
--------------------:------------
allowed-client       : -
denied-client        : -
enabled              : false
key-manager-provider : -
listen-port          : 1689
ssl-cert-nickname    : server-cert
use-ssl              : false
```

#### 14.1.5.4.1 To Change the Port on Which the Server Listens for JMX Connections

This example changes the port on which the server listens for JMX connections to 1789.

Use the dsconfig command as follows:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -n \
  set-connection-handler-prop \
  --handler-name "JMX Connection Handler" --set listen-port:1789
```

### 14.1.6 Configuring Network Groups With dsconfig

Network groups are the single entry point of all client requests to the Oracle Unified Directory. The network group handles all client interactions, dispatching them and delegating the treatment of the request to workflows. A client connection is associated to the network group with the highest priority and for which all the criteria are met. During installation, a default network group with a priority of 1 is created. To set request filtering policies or resource limits, you must create a network group quality of service policy.

Each network group is associated with one or more workflows. The workflows provide access to a naming context (or suffix). By associating a workflow with a network group, you indicate to the network group which naming contexts are available. Typically to create a network group, you would already have a workflow created. For information about workflows, see Section 14.1.7, "Configuring Workflows With dsconfig."

This section describes how to configure network groups using the dsconfig command, and covers the following topics:

- Section 14.1.6.1, "Creating a Network Group"
- Section 14.1.6.2, "Modifying Network Group Properties"
- Section 14.1.6.3, "Creating a Network Group Quality of Service Policy"
- Section 14.1.6.4, "Modifying a Network Group Quality of Service Policy"
- Section 14.1.6.5, "Relocating the Root DSE Entry for a Network Group"

All the commands in the following procedures specify the hostname (-h), the admin port (-p), the bind DN (-D), and the bind password file (-j). The examples use the -X option to trust all certificates.

#### 14.1.6.1 Creating a Network Group

You can create many network groups, in which case client requests will be handled by the network group with the highest priority, for which the criteria are met. Therefore, when you create a network group, you must consider all the network groups you plan to create, and the priority of each. The priority can be 0 or above, where 0 is the highest priority.

It is possible to create two network groups with the same priority. However, if two or more network groups have the same priority and match the client request, the network group that will handle the request is random, among those matching the client request. You should therefore specify a different priority for each network group.

The default properties of a new network group are as follows.

```
Property                Value(s)
-------------------------------------------------------------------------
allowed-auth-method     All authorization methods are allowed.
allowed-bind-dn         All bind DNs are allowed.
allowed-bind-id         All bind IDs are allowed.
allowed-client          All clients with addresses that do not match
                        an address on the deny list are allowed. If
                        there is no deny list, then all clients are
                        allowed.
allowed-protocol        All supported protocols are allowed.
certificate-mapper      The global certificate mapper will be used.
denied-client           If an allow list is specified, then only
                        clients with addresses on the allow list are
                        allowed. Otherwise, all clients are allowed.
enabled                 true
generic-identity-mapper The global generic identity mapper will be
                        used.
gssapi-identity-mapper  The global GSSAPI identity mapper will be
                        used.
is-security-mandatory   false
priority                1
workflow                userroot0
```

To create a network group, use the `dsconfig create-network-group` command, as follows:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  create-network-group --group-name network-group1 --set enabled:true\
  --set workflow:workflow1 --set priority:1
```

After you have created a network group, you can associate a network group quality of service policy to it. For information about creating a quality of service policy, see Section 14.1.6.3, "Creating a Network Group Quality of Service Policy."

### 14.1.6.2 Modifying Network Group Properties

The network group properties filter the traffic and indicate how a request is directed.

You can modify network group properties, by using the `dsconfig set-network-group-prop` command. For example, to modify the *priority* of the network group:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  set-network-group-prop --group-name network-group1 --set priority:3
```

You can configure the network group properties to set the following criteria:

- the authentication method allowed between the client and the network group (`allowed-auth-method`).
- the bind DN allowed to connect to the network group (`allowed-bind-dn`).
- the list of clients authorized to access the Oracle Unified Directory (`allowed-client`), expressed by the IP address or name of the client. If no

allowed client list is provided, all clients are allowed, assuming they are not listed in the denied client list.

- the protocol allowed to connect to the Oracle Unified Directory (allowed-protocol). If none is specified, all protocols are allowed.
- the name of the certificate mapper that should be used to match client certificates to user entries (certificate-mapper). If none is specified, the global certificate mapper is used.
- the list of clients not authorized to access the Oracle Unified Directory (denied-client). If no denied client list is provided, all clients are authorized, assuming there is no limitation set by an allowed client list.
- the set of identity mappers that will be used by the network group to map an identity while performing SIMPLE, non-GSSAPI SASL bind requests and proxy authorization controls (generic-identity-mapper).
- the set of identity mappers that will be used by the network group to map an identity while performing GSSAPI/SASL bind requests (gssapi-identity-mapper).
- whether security between the client and the Oracle Unified Directory is always required (is-security-mandatory).
- the priority of the network group (priority). A client connection is first compared against the network group with the highest priority. If the client connection does not match its connection criteria, the client connection is compared against the network group with the next highest priority, and so on. If no network group is selected, the client connection is rejected.

For example, you can ensure that no connections are accepted from the IP address 208.77.188.166, by network-group1 as follows:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  set-network-group-prop --group-name network-group1 \
  --set denied-client:208.77.188.166
```

#### 14.1.6.2.1 Setting an Allowed or Denied Client List

For allowed-client and denied-client lists, you must be aware of the name service configuration on the server. For example, if the name service knows the host as myclienthost.example.com, you must specify myclienthost.example.com as the value, and not just myclienthost. Similarly, if the name service knows the host as myclienthost, you must specify the value as myclienthost. If you do not know how the name service is configured, you should specify both the fully qualified domain name (for example myclienthost.sun.com) and the short name (myclienthost) of the machine. Specifying multiple values will ensure that the name is resolved correctly. For example:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  set-network-group-prop \
  --group-name network-group1 \
  --add denied-client:myhost \
  --add denied-client:myhost.example \
  --add denied-client:myhost.example.com
```

To avoid any issues, use the IP address for clarity.

If you use localhost or the name of the local machine when connecting to Oracle Unified Directory, the IP addresses of the client will be different. To prevent

connections from the localhost, specify both `localhost` and the name of the local machine in the list of denied clients.

#### 14.1.6.3 Creating a Network Group Quality of Service Policy

You can, optionally, associate a quality of service (QoS) policy with a network group. A QoS policy applies additional filtering criteria to client connections to determine how the network group handles the request.

Oracle Unified Directory supports four types of QoS policy:

- request filtering policy
- resource limits
- affinity
- referral

> **Note:** ODSM accesses an Oracle Unified Directory instance over the administration connector. The administration connector is not subject to the QoS policies defined for a network group. ODSM therefore bypasses the QoS policies defined for a network group. For more information, see Section 14.3, "Managing Administration Traffic to the Server".

To create a network group quality of service policy, use the `dsconfig create-network-group-qos-policy` command. You must specify the name of the network group to which the quality of service policy applies, and the type of quality of service policy.

##### 14.1.6.3.1 Creating a Request Filtering Quality of Service Policy

A request filtering policy applies the following criteria to an incoming client request:

- `allowed-attributes`: list of attributes that can be specified in the filter of a search request
- `allowed-operations`: type of operation accepted by the network group. For example, you can specify that a network group should accept only read requests.
- `allowed-search-scopes`: scope of a search accepted, for example one-level only.
- `allowed-subtrees`: list of specific subtrees that can be specified as a base DN in a search request
- `prohibited-attributes`: list of attributes which, if specified in the filter of a search request, will be rejected
- `prohibited-subtrees`: list of specific subtrees that, if specified as base DNs in a search request, may not be specified will not manage a request

The following example defines a request filtering policy that ensures that users can only search and not modify data:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  create-network-group-qos-policy --group-name network-group1 \
  --type request-filtering --set allowed-operations:search
```

##### 14.1.6.3.2 Creating a Resource Limit Quality of Service Policy

A resource limit policy sets specific limits on the client connections that can access the server via that network group. The following limits can be defined:

- max-concurrent-ops-per-connection: the maximum number of simultaneous operations per established connection. To run the server in synchronous mode, set the maximum to 1.
- max-ops-per-connection: the maximum number of operations per connection.
- max-connections: the maximum number of concurrent client connections to the server. If you do not set a maximum number of connections, the server limit is used.
- max-connections-from-same-ip: the maximum number of connections from the same IP address. Set this parameter if you want to avoid Denial of Service attacks. This parameter should not be set if you know that most requests typically come from the same client.
- max-ops-per-interval: the maximum number of operations per specified interval. For example, a setting of 1,000 will limit the number of operations to 1,000 per the interval set using max-ops-interval.
- max-ops-interval: the interval during which the number of operations is counted for the max-ops-per-interval parameter. For example, an interval set to one second results in operations being counted per second. The limit (max-ops-per-interval) is checked and enforced during each interval.
- min-substring-length: the minimum search string length. The shorter the search string, the more results that need to be found and displayed. It is therefore useful to set a minimum search string length in the substring search filter to limit the resources that are used.
- size-limit: the maximum number of entries that can be returned to the client during a single search operation. It is recommended that you keep the default setting for this property.
- time-limit: the maximum length of time that should be spent processing a search operation. It is recommended that you keep the default setting for this property.

The following example defines a resource limit policy that ensures that a user enters a search string of at least five characters, to limit the number of return values:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  create-network-group-qos-policy --group-name network-group1 \
  --type resource-limits --set min-substring-length:5
```

#### 14.1.6.3.3 Creating an Affinity Quality of Service Policy

In a load balancing deployment, you can use *affinity* to override the regular routing process. The properties of the affinity policy determine the routing process that should be followed.

The following properties can be configured:

- affinity-policy: specifies the routing policy that should be used.

  The affinity policy can take one of the following values:

  - all-requests-after-first-request
  - all-requests-after-first-write-request

- all-write-requests-after-first-write-request
- first-read-request-after-write-request

Specific operations will set affinity, depending on the affinity policy. For the first policy in the previous list (all-requests-after-first-request) all operations will set affinity. For the remaining policies (all-requests-after-first-write-request, all-write-requests-after-first-write-request, and first-read-request-after-write-request) only an ADD, DELETE, MOD or MODDN operation will set affinity.

- affinity-timeout defines the duration during which the affinity applies.

Even when affinity has been set by a previous operation, the load balancing algorithm is only bypassed in specific situations, depending on the affinity policy and the current operation type. If the affinity policy is all-requests-after-first-request or all-requests-after-first-write-request, the affinity route will be used for every operation type, unless the affinity timeout has expired. If the affinity policy is all-write-requests-after-first-write, the affinity route will be used for any ADD, DELETE, MOD or MODDN operation, unless the timeout has expired. The affinity route will not be used for other operations. If the affinity policy is first-read-request-after-write-request, the affinity route will be used for all operations *except* ADD, DELETE, MOD or MODDN operations, unless the timeout has expired.

The following example sets an affinity policy that can be set by any operation and used for all operations, for a maximum of sixty seconds.

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  create-network-group-qos-policy --group-name network-group1 \
  --type affinity --set affinity-timeout:60s \
  --set affinity-policy:all-requests-after-first-request
```

**Note:** The affinity feature can be used with all load balancing algorithms except for the failover algorithm. With the failover algorithm, only one route is active at a time. The active route changes when the remote server goes down, so all connections to the remote server are broken. Affinity can therefore not apply in a failover scenario.

#### 14.1.6.3.4 Creating a Referral Quality of Service Policy

You can configure the behavior of a proxy server when a referral is received from the remote LDAP server by defining a referral quality of service policy. The referral itself must be defined on the remote LDAP server.

When you create a network group quality of service, you can set the following referral properties:

- the maximum number of hops supported (referral-hop-limit) when the referral policy is set to follow. The default is set to 5.
- define the type of referral policy (referral-policy), such as discard, forward, or follow. This defines how a referral will be treated by the network group.

For example, the referral-policy is set by default to forward. You can change it to discard or to follow, as follows:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  create-network-group-qos-policy --group-name network-group1 \
  --type referral --set referral-policy:follow
```

#### 14.1.6.4 Modifying a Network Group Quality of Service Policy

To modify a QoS policy, use the `dsconfig set-network-group-qos-policy-prop` command, specifying the network group name and the policy type.

The following example sets the minimum search string limit of a resource limits quality of service policy.

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  set-network-group-qos-policy-prop --group-name network-group1 \
  --policy-type resource-limits --set min-substring-length:5
```

#### 14.1.6.5 Relocating the Root DSE Entry for a Network Group

The Root DSE is a special entry that provides information about the server's name, version, naming contexts, and supported features. The Root DSE entry of a network group can be in a local server or a remote server.

To relocate the Root DSE, use the `dsconfig set-network-group-prop` command, as shown in the following example:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  set-network-group-prop --group-name network-group1 \
  --set relocated-rootdse-workflow-element:<new rootDSE workflow element>  \
```

The value of the `relocated-rootdse-workflow-element` property is the workflow element where a Root DSE can be found (This is the entry returned by a search on the null DN).

### 14.1.7 Configuring Workflows With `dsconfig`

A workflow is the link between the network group and the naming context (suffixes). It defines the naming context that will be accessible for a given network group, when handling a request to a load balancing or distribution configuration. To create a workflow, you must already have a load balancing or distribution workflow element created. For information on workflow elements, see Section 14.1.8, "Configuring Workflow Elements With dsconfig."

The proxy automatically creates a number of private workflows. These workflows should not be modified or deleted. Privacy settings of the remote LDAP servers must be considered when configuring workflows. Privacy settings are as follows:

**LDIFBackend**
Privacy defined by the property `ds-cfg-is-private-backend`. This flag is set by default to private, but can be changed.

**JEB backend**
Always public, and contains user data.

**Config File Handler backend**
Always private

**Backup backend**
Always private

**Schema backend**
Always private

**Tasks backend**
Always private

**Monitor backend**
Always private

**Truststore backend**
Always private

This section describes examples to configure workflows using the dsconfig command, and contains the following topics:

- Section 14.1.7.1, "Listing Existing Workflows"
- Section 14.1.7.2, "Viewing Workflow Properties"
- Section 14.1.7.3, "Creating a Workflow"

All the commands in the following procedures specify the proxy hostname (-h), the proxy admin port (-p), the bind DN (-D), and the bind password file (-j). The examples use the -X option to trust all certificates.

### 14.1.7.1 Listing Existing Workflows

To display all the workflows configured on a server instance, use the dsconfig list-workflows command. The following example shows the default workflow configured on a proxy server instance:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  list-workflows

Workflow       : Type    : enabled
---------------:---------:--------
workflow1      : generic : true
```

### 14.1.7.2 Viewing Workflow Properties

To view the properties of a specific workflow, use the dsconfig get-workflow-prop command. For example:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  get-workflow-prop --workflow-name workflow1

Property         : Value(s)
-----------------:-------------------
base-dn          : "ou=people,o=test"
enabled          : true
workflow-element : load-bal-we1
```

The base-dn indicates the base DN used for the workflow, and therefore for the deployment using that workflow. The workflow-element property indicates the workflow element that will process the requests.

---

**Note:** The base-dn property is read-only and cannot be modified.

---

#### 14.1.7.3 Creating a Workflow

Each workflow is associated with a workflow element. When you create a workflow, you must specify the associated workflow element name (`--set workflow-element`). In other words, you must already have created the workflow element. See Section 14.1.8, "Configuring Workflow Elements With dsconfig."

To create a workflow, use the `dsconfig create-workflow` command. For example:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  create-workflow \
  --workflow-name workflow1 \
  --set base-dn:ou=people,o=test \
  --set enabled:true \
  --set workflow-element:load-bal-we1
```

### 14.1.8 Configuring Workflow Elements With dsconfig

Workflow elements are part of a routing structure, and are linked to workflows. In the case of a directory server instance, DB local workflow elements are associated with a physical database.

For information about all the types of workflow elements that can be configured, and what they are used for, see Section 4.1.3, "Workflow Elements".

A proxy deployment must include LDAP proxy workflow elements and either a load balancing or distribution workflow element.

This section describes how to configure workflow elements by using the `dsconfig` command, and covers the following topics:

- Section 14.1.8.1, "Listing Workflow Elements"
- Section 14.1.8.2, "Creating Workflow Elements"
- Section 14.1.8.3, "Modifying Workflow Elements"

All the commands in the following procedures specify the hostname (`-h`), the administration port (`-p`), the bind DN (`-D`), and the bind password file (`-j`). The examples use the `-X` option to trust all certificates.

#### 14.1.8.1 Listing Workflow Elements

To display all the configured workflow elements, use the `dsconfig list-workflow-elements` command.

The following example shows the default workflow elements for a directory server instance.

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  list-workflow-elements

Workflow Element : Type               : enabled
-----------------:--------------------:--------
adminRoot        : ldif-local-backend : true
userRoot         : db-local-backend   : true
virtualAcis      : db-local-backend   : true
```

The following example shows the default workflow elements for a proxy server instance, deployed for load balancing:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  list-workflow-elements
```

```
Workflow Element : Type                : enabled
-----------------:--------------------:--------
adminRoot        : ldif-local-backend : true
load-bal-we1     : load-balancing     : true
proxy-we1        : proxy-ldap         : true
proxy-we2        : proxy-ldap         : true
```

### 14.1.8.2 Creating Workflow Elements

To create workflow elements in interactive mode, use the `dsconfig create-workflow-element` command. If you configured a proxy instance during the setup, the required workflow elements will already have been created.

You can create the following types of workflow elements:

- DB Local Backend. For more information, see Section 14.1.8.2.1, "To Create a DB Local Backend Workflow Element.".
- Proxy LDAP. For more information, see Section 15.1.1.3.3, "To Create a Proxy LDAP Workflow Element."
- Load balancing. For more information, see Section 15.1.3.2, "Creating a Load Balancing Workflow Element."
- Distribution. For more information, see Section 15.1.4.2, "Creating a Distribution Workflow Element."
- DN renaming. For more information, see Section 15.1.5, "Configuring DN Renaming With dsconfig."
- Kerberos Authentication. For more information, see Section 20.8.4, "Creating a Kerberos Workflow Element Using dsconfig"

#### 14.1.8.2.1 To Create a DB Local Backend Workflow Element

A local backend workflow element provides access to a backend in a directory server instance. To create a new local backend workflow element, use the `dsconfig create-workflow-element` command, specifying one or more base DNs that will be accessed through the workflow element.

A single backend can be responsible for one or more base DNs. No two backends may have the same base DN, but one backend can have a base DN that is below a base DN provided by another backend. If any of the base DNs is subordinate to a base DN for another backend, then all base DNs for that backend must be subordinate to that same base DN.

The following example creates and enables a local backend workflow element to access the base DN `ou=admins,dc=example,dc=com`.

```
$ dsconfig create-workflow-element -h localhost -p 4444 -D "cn=directory manager"\
  -j pwd-file -X -n --element-name admins --type db-local-backend \
  --set base-dn:ou=admins,dc=example,dc=com --set enabled:true
```

### 14.1.8.3 Modifying Workflow Elements

Once you have created a workflow element, you can modify its properties using the `dsconfig set-workflow-element-prop` command.

## 14.1.9 Configuring Plug-Ins With dsconfig

Plug-ins are responsible for providing custom logic in the course of processing an operation or at other well-defined points within the directory server. The dsconfig command is used to manage the configuration of the directory server. For information about using dsconfig, see Section 14.1, "Managing the Server Configuration With dsconfig". This section covers the following topics:

- Section 14.1.9.1, "Overview of Plug-In Types"
- Section 14.1.9.2, "Modifying the Plug-In Configuration"

### 14.1.9.1 Overview of Plug-In Types

The dsconfig plugin-type property can be used to configure a plug-in to use one or more of the numerous plug-in types supported by the server. You cannot add a new default plug-in type to the configuration of an existing plug-in. Although, you can remove one or more of the default plug-in type values from a plug-in's configuration, you must take care when doing this. Usually a plug-in has been engineered to support its default plug-in types for a reason. Removing one or more plug-in types might endanger the safe operation of the directory server.

Most of the plug-ins support more than one type, and multiple plug-ins are sometimes defined with the same plug-in type. The order in which these plug-ins are invoked during processing is undefined. If a specific order is required (for example, if the processing performed by one plug-in depends on the result of another), you can specify the order in which the plug-ins are invoked. For more information, see Section 14.1.9.2.5, "To Configure Plug-In Invocation Order."

### 14.1.9.2 Modifying the Plug-In Configuration

The following sections show various examples of managing plug-in configuration using dsconfig. dsconfig uses the administration connector to access the server. All of the examples in this section assume that the administration connector is listening on the default port (4444) and that the command is accessing the server running on the local host. If this is not the case, the --port and --hostname options must be specified.

The dsconfig command always accesses the server over a secured connection with certificate authentication. If you run dsconfig in interactive mode, you are prompted as to how you want to trust the certificate. If you run dsconfig in non-interactive mode (that is, with the -n option) you must specify the -X or --trustAll option, otherwise the command will fail.

This section describes examples to manage plug-in configuration, and covers the following topics:

- Section 14.1.9.2.1, "To Display the List of Plug-Ins"
- Section 14.1.9.2.2, "To Create a New Plug-In"
- Section 14.1.9.2.3, "To Enable or Disable a Plug-In"
- Section 14.1.9.2.4, "To Display and Configure Plug-In Properties"
- Section 14.1.9.2.5, "To Configure Plug-In Invocation Order"

#### 14.1.9.2.1 To Display the List of Plug-Ins

This example shows a directory server configured with the current supported plug-ins. For a description of these plug-ins and their purpose, see "The Plug-In Configuration" in the Oracle Unified Directory Configuration Reference.

Use dsconfig to display the list of plug-ins that are currently configured.

```
$ dsconfig -h localhost -p 4444 -D cn="Directory Manager" -j pwd-file -X -n \
  list-plugins
```

Depending on your installation, the output will be similar to the following.

```
Plugin                          : Type                            : enabled
--------------------------------:---------------------------------:--------
7-Bit Clean                     : seven-bit-clean                 : false
Change Number Control           : change-number-control           : true
Entry UUID                      : entry-uuid                      : true
LastMod                         : last-mod                        : true
LDAP Attribute Description List : ldap-attribute-description-list : true
Password Policy Import          : password-policy-import          : true
Profiler                        : profiler                        : true
Referential Integrity           : referential-integrity           : false
Replication LDIF Import         : replication-ldif-import         : true
UID Unique Attribute            : unique-attribute                : false
```

The output of the command shows (from left to right):

- **Plug-in.** The name of the plug-in, usually descriptive of what it does.
- **Type.** The type of plug-in. It is possible to have more than one plug-in of a specific type.
- **Enabled.** Plug-ins can be enabled or disabled. Disabled plug-ins remain in the server configuration but do not perform any processing.

#### 14.1.9.2.2 To Create a New Plug-In

The easiest way to configure plug-ins is to use dsconfig in interactive mode. Interactive mode walks you through the plug-in configuration, and is therefore not documented here.

This example creates and enables a new Password Policy Import Plug-in by using dsconfig in non-interactive mode.

Run the dsconfig command to create and enable a new Password Policy Import plug-in, as follows:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  create-plugin --type password-policy-import \
  --plugin-name "My Password Policy Import Plugin" --set enabled:true
```

#### 14.1.9.2.3 To Enable or Disable a Plug-In

You can enable or disable a plug-in by setting the enabled property to true or false. This example disables the Password Policy Import plug-in created in the previous example.

Run the dsconfig command to disable the new Password Policy Import plug-in.

```
$ dsconfig -h localhost -p 4444 -D cn="Directory Manager" -j pwd-file -X -n \
  set-plugin-prop --plugin-name "My Password Policy Import Plugin" \
  --set enabled:false
```

#### 14.1.9.2.4 To Display and Configure Plug-In Properties

To display the properties of a plug-in, use the get-plugin-prop subcommand. To change the properties of a plug-in, use the set-plugin-prop subcommand. This example displays the properties of the plug-in created in the previous example, then

enables the plug-in and sets the default authentication password storage scheme to Salted SHA-512.

1. Display the plug-in properties.

```
$ dsconfig -h localhost -p 4444 -D cn="Directory Manager" -j pwd-file -X -n \
  get-plugin-prop --plugin-name "My Password Policy Import Plugin"
```

   Depending on your installation, the output will be similar to the following.

```
Property                            : Value(s)
------------------------------------:---------
default-auth-password-storage-scheme : -
default-user-password-storage-scheme : -
enabled                             : false
```

2. Enable the plug-in and set the default authentication password storage scheme to Salted SHA-512.

```
$ dsconfig -h localhost -p 4444 -D cn="Directory Manager" -j pwd-file -X -n \
  set-plugin-prop --plugin-name "My Password Policy Import Plugin" \
  --set enabled:true\
  --set default-auth-password-storage-scheme:"Salted SHA-512"
```

3. Display the plug-in properties again to verify the change.

```
$ dsconfig -h localhost -p 4444 -D cn="Directory Manager" -j pwd-file -X -n \
  get-plugin-prop --plugin-name "My Password Policy Import Plugin"

Property                            : Value(s)
------------------------------------:---------------
default-auth-password-storage-scheme : Salted SHA-512
default-user-password-storage-scheme : -
enabled                             : true
```

#### 14.1.9.2.5 To Configure Plug-In Invocation Order

By default, the order in which plug-ins are invoked is undefined. You can specify that plug-ins be invoked in a specific order by using the `set-plugin-root-prop --set plugin-type:value` subcommand. The *value* in this case is the plug-in order, expressed as a comma-delimited list of plug-in names. The plug-in order string should also include a single asterisk element, which is a wildcard that will match any plug-in that is not explicitly named.

This example specifies that the Entry UUID plug-in should be invoked before any other pre-operation add plug-ins.

1. Display the current plug-in invocation order.

```
$ dsconfig -h localhost -p 4444 -D cn="Directory Manager" -j pwd-file -X -n \
  get-plugin-root-prop

Property                                  : Value(s)
------------------------------------------:---------
plugin-order-intermediate-response        : -
plugin-order-ldif-export                  : -
plugin-order-ldif-import                  : -
plugin-order-post-connect                 : -
...
```

2. Set the plug-in order.

```
$ dsconfig -h localhost -p 4444 -D cn="Directory Manager" -j pwd-file -X -n \
```

```
set-plugin-root-prop --set plugin-order-pre-operation-add:"Entry UUID,*"
```

**Note:** Plug-in order values are not validated. Values that do not match defined plug-ins are ignored.

## 14.1.10 Configuring Suffixes with dsconfig

Oracle Unified Directory allows you to configure multiple suffixes, either during the setup or later.

This section contains the following topics:

- Section 14.1.10.1, "Configuring Suffixes with dsconfig During Setup"
- Section 14.1.10.2, "Configuring Suffixes with dsconfig on a Running Server"

You can also use dsconfig in interactive mode to achieve the configuration described in the following sections.

### 14.1.10.1 Configuring Suffixes with dsconfig During Setup

You can configure suffixes with the dsconfig command during the setup by creating the base entries.

Perform one of the following steps to create the base entries, dc=example,dc=com;dc=other,dc=com;dc=test,dc=com.

- Create the base entries using the following command:

```
oud-setup --cli --baseDN dc=example,dc=com --baseDN dc=test,dc=com --baseDN \
dc=other,dc=com --addBaseEntry --ldapPort 2389 --adminConnectorPort 24444 \
--rootUserDN cn=Directory Manager --rootUserPassword password --no-prompt \
--noPropertiesFile
```

- Create the base entries with sample data using the following command:

```
oud-setup --cli --baseDN dc=example,dc=com --baseDN dc=test,dc=com --baseDN \
dc=other,dc=com --sampleData 15 --ldapPort 2389 --adminConnectorPort 24444 \
--rootUserDN cn=Directory Manager --rootUserPassword password --no-prompt \
--noPropertiesFile
```

You can now access data below all the suffixes without additional configuration.

### 14.1.10.2 Configuring Suffixes with dsconfig on a Running Server

You can configure suffixes on a running server instance either with the dsconfig command or by using ODSM. For more information about configuring suffixes with ODSM, see Section 14.2.3.1, "Create a Suffix."

Perform the following steps to configure suffixes with the dsconfig command:

1. Add the base DN to your local backend workflow element.

```
dsconfig set-workflow-element-prop \
--element-name userRoot \
--add base-dn:dc=example2,dc=com \
--hostname localhost \
--port 24444 \
--trustAll \
--bindDN cn=directory manager \
--bindPassword ****** \
--no-prompt
```

2. Create a workflow for your new base DN.

```
dsconfig create-workflow \
--set base-dn:dc=example2,dc=com \
--set enabled:true \
--set workflow-element:userRoot \
--type generic \
--workflow-name dc=example2,dc=com \
--hostname localhost \
--port 24444 \
--trustAll \
--bindDN cn=directory manager \
--bindPassword ****** \
--no-prompt
```

3. Add your new workflow to your network group.

```
dsconfig set-network-group-prop \
--group-name network-group \
--add workflow:dc=example2,dc=com \
--hostname localhost \
--port 24444 \
--trustAll \
--bindDN cn=directory manager \
--bindPassword ****** \
--no-prompt
```

4. Create the base entry, `dc=example2,dc=com`.
5. Populate your new suffix with the required entries.

## 14.2 Managing the Server Configuration With Oracle Directory Services Manager

The Configuration tab of each server instance in ODSM enables you to modify elements of the server configuration. For additional information about managing the configuration that is specific to a proxy server instance, see Section 15.2, "Managing the Proxy Configuration With ODSM."

This section provides an overview of the tasks that can be performed on the Configuration tab in ODSM, and covers the following topics:



### 14.2.1 Selecting a Configuration View

The Configuration tab presents two separate views of the server configuration:

- **Naming Contexts.** This is the default view, and shows the server configuration in terms of the naming contexts or suffixes configured on that server instance.
- **Core Configuration.** This view displays the server configuration in terms of the workflows, workflow elements and server extensions configured on that server instance.

The configuration view that you select determines the items that are available under the **Create** menu.

## 14.2.2 Shortcuts to Configuring Objects With ODSM

When you create server components by using ODSM, you can duplicate an existing component using the **Create Like**  . When you select a component on the configuration tab and click **Create Like**, a new component with the same configuration is created. You can then edit the properties of the new component to suit your requirements.

You can also use the **Create**  to create the same type of component as the one you have selected. For example, if you select LDAP Connection Handler in the left hand menu, and click **Create**, a new, unconfigured LDAP connection handler is created.

Right-clicking on a component in the left hand menu provides a list of actions related to that component. For example, if you right-click on LDAP Connection Handler, a drop-down menu is displayed, enabling you to create a new LDAP connection handler, duplicate that LDAP connection handler, or delete the connection handler.

## 14.2.3 Configuring Suffixes With ODSM

The following sections describe how to configure suffixes, or naming contexts, by using ODSM. For information about configuring suffixes by using `dsconfig`, see Section 14.1.10, "Configuring Suffixes with `dsconfig`".

### 14.2.3.1 Create a Suffix

Oracle Unified Directory allows you to configure one or more suffixes by using the ODSM interface as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager."
2. Select the **Configuration** tab.
3. Select the **Naming Contexts** view.
4. From the **Create** menu, select **Local Naming Context.**
5. In the Naming Context region, perform the following steps:
   a. In the **Base DN** field, type a name for the suffix that you want to create.
   b. From the Directory Data Options group, select one of the following options for populating the suffix with data:

      **Only Create Base Entry** creates the database along with the base entry of the suffix. Any additional entries must be added after suffix creation.

      **Leave Database Empty** creates an empty database. The base entries and any additional entries must be added after suffix creation.

      **Import Generated Sample Data** populates the suffix with sample entries.

Specify the number of entries that should be generated in the **Number of User Entries** field. You can import a maximum of 30,000 sample entries through ODSM. If you want to add more than 30,000 entries, you must use the `import-ldif` command.

6. In the Oracle Components Integration region, select one of the following option to enable the new suffix:
   - **No Specific Integration**: Select this option, if you do not want to integrate the naming context with Oracle components.
   - **Enable for Enterprise User Security (EUS)**:

     To enable a suffix for EUS, you must have at least one LDAP listener with SSL enabled, in addition to the administration listener. The suffix must contain at least one entry (in other words, you must *not* have selected "Leave Database Empty" in the previous step).

     When you select EUS, in addition to creating this suffix, two suffixes are created automatically: `"cn=oracleschemaversion"` and `"cn=oraclecontext."` An EUS workflow element is also added in front of the local backend workflow element. Further, a DN renaming workflow element for `"cn=schema"` is added, so that it can be accessed using the `"cn=subschemasubentry"` DN.
   - **Enable for Oracle Database Net Services**: Select this option if you want the naming context to store the Database Connect Identifiers.
7. In the Network Group region, attach the suffix to at least one network group by performing the following steps:
   - To attach the suffix to an existing network group, select **Use Existing** and select the required network group from the list.
   - To attach the suffix to a new network group, select **Create New** and then in the **Name** field, type a name for the network group you want to create.

   You can attach several network groups to the same suffix.
8. In the Workflow Element region, attach the suffix to the workflow element by performing either of the following steps:
   - To attach the suffix to an existing workflow element, select **Use Existing** and then select the required workflow element from the list.
   - To attach the suffix to a new workflow element, select **Create New** and then in the **Name** field, type a name for the workflow element you want to create. You can create a Local DB Workflow Element or a Local LDIF Workflow Element.
9. Click **Create.**

   The following confirmation message is displayed:

   ```
   Configuration created successfully.
   ```

   You can configure the tombstone entry purge interval and the tombstone entry lifetime after creating the suffix, in the local backend workflow element configuration.

#### 14.2.3.2 Display and Edit Suffix Properties

In the Naming Contexts view, the Configuration tab displays all of the suffixes that have been configured on the server.

To display the properties of a configured suffix, follow these steps:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager."
2. Select the **Configuration** tab.
3. Select the **Naming Contexts** view.
4. Expand the **Naming Contexts** element.
5. Click the suffix whose properties you want to display.

   The suffix properties are displayed in the right hand pane.
6. Make any required changes to the suffix configuration.

   You can change the network group to which this suffix is attached, and enable the suffix for Enterprise User Security (EUS) or Enable for Oracle Database Net Services.

   ---

   **Note:** If the Oracle Components Integration option was previously configured for the **Enable for Enterprise User Security (EUS)** or the **Enable for Oracle Database Net Services** options and if you have made changes in the Oracle Components Integration region, the **Configuration Required** dialog box appears. Depending on the option you choose, select one of the following:

   - **Keep Oracle Context**: Select this option, if you want to keep the naming context for EUS and Oracle Database Net Service.
   - **Delete Oracle Context**: Select this option, if you want to delete the naming context for EUS and Oracle Database Net Service.

   ---

   Click **Apply** to save your changes.

#### 14.2.3.3 Delete a Suffix

In the Naming Contexts view, the Configuration tab displays all of the suffixes that have been configured on the server.

To delete a suffix, follow these steps:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager."
2. Select the **Configuration** tab.
3. Select the **Naming Contexts** view.
4. Expand the **Naming Contexts** element.
5. Select the suffix that you want to delete.
6. Click the **Delete configuration** .

### 14.2.4 Configuring Workflow Elements With ODSM

A workflow element is the key building block of a workflow process. Workflow elements define how client requests that are sent to the server are treated. In a deployment that includes a proxy server, workflow elements are configured for load balancing or distribution. In a deployment that does not include a proxy server, workflow elements are configured directly for each backend.

The following sections describe how to configure workflow elements by using ODSM. For information about configuring workflow elements by using `dsconfig`, see Section 14.1.8, "Configuring Workflow Elements With `dsconfig`".

#### 14.2.4.1 Create a Workflow Element

To create a workflow element by using ODSM, follow these steps:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager."
2. Select the **Configuration** tab.
3. Select the **Core Configuration** view.

   For more information, see Section 14.2.1, "Selecting a Configuration View".
4. From the **Create** menu, select **Workflow Element** and select the type of workflow element that you want to create.

   For more information about the various workflow element types, see Section 4.1.3, "Workflow Elements".
5. The properties of the workflow element that must be configured depend on the type of workflow element that you are creating.

   All workflow elements require the following basic properties to be configured:

   **Name.** Enter a name for the workflow element.

   **Enabled.** When you create a workflow element, it is enabled by default. Clear this item to disable the workflow element.

   In addition, the following properties must be configured for each corresponding workflow element type:

   - **DN Renaming Workflow Element**
     - **Client Base DN**: Specify the base DN that is used by the client application.
     - **Source Base DN**: Specify the base DN that is stored in the LDAP server.
     - **Next Workflow Element**: Select the workflow element that should be next in the workflow.
     - **Attribute White List**: Click **Add** to select the list of attributes that contain DNs and must be transformed by the renaming operation.
     - **Attribute Black List**: Click **Add** to select the list of attributes that contain DNs but must *not* be transformed by the renaming operation.
   - **EUS Workflow Element**
     - **EUS Realm**: Enter the part of the DIT to which the EUS workflow element applies.
     - **Next Workflow Element**: Select the workflow element that should be next in the workflow.
     - **Server Type**: Select the server containing the EUS user entries.
     - **Password Attribute**: Enter the attribute type that should be used to hold the EUS user passwords.
   - **EUS Context Workflow Element**

  - **EUS Context**: Enter the DN that contains the Oracle Context. The oracle context is a top-level directory entry that contains the data used by any installed Oracle product that uses the directory.
  - **EUS Administrator**: Enter the DN of the administration user. This user will be the `uniquemember` of the groups created in Oracle Context.
  - **Next Workflow Element**: Select the workflow element that should be next in the workflow.

- **Kerberos Authentication Provider Workflow Element**
  - **Realm**: Specify the realm to be used for Kerberos authentication. If you do not specify any realm then the server attempts to determine the realm from the underlying system configuration.
  - **Principal Name Attribute**: Click **Select** and specify the Principal Name Attribute.
  - **KDC Address**: Specify the Key Distribution Center (KDC) server address.
- **Local DB Workflow Element**
  - **Writability Mode**: Specify whether the backend associated with this workflow element should process write operations.
  - **Base DN**: Specify one or more base DNs for the data that is handled by the backend.
  - **Database Properties**: Specify any specific properties for the database. For a detailed list of these properties, and their values, see "DB Local Backend Workflow Element" in the *Oracle Unified Directory Configuration Reference*.
  - **Tombstone Configuration**: Specify how tombstone entries should be handled for the database. For a detailed list of these properties, and their values, see "DB Local Backend Workflow Element" in the *Oracle Unified Directory Configuration Reference*.
  - **Index Properties**: Specify the following parameters:

    **Index Subtrees**: Enable or disable the check box to indicate whether or not the backend should index subtrees to maintain subtree specific data retaining information on direct and indirect children entries of each parent entry.

    **Local DB Index**: Specify the local DB index configuration for the database. For a detailed list of these properties, and their values, see "DB Local Backend Workflow Element" in the *Oracle Unified Directory Configuration Reference*.

    **Local DB VLV Index**: Specify the local DB VLV index configuration for the database. For a detailed list of these properties, and their values, see "DB Local Backend Workflow Element" in the *Oracle Unified Directory Configuration Reference*.
- **Local LDIF Workflow Element**
  - **Writability Mode**: Specify whether the backend associated with this workflow element should process write operations.
  - **Base DN**: Specify one or more base DNs for the data that is handled by the backend.

- **Private Backend**: Specify whether the backend should be considered a private backend, which indicates that it is used for storing operational data rather than user-defined information.
- **LDIF File**: Enter the path to the LDIF file containing the data for this backend.

- **Pass Through Authentication Workflow Element**
  - **User Provider Workflow Element**: Select the workflow element providing the requested user entry.
  - **Authentication Provider Workflow Element**: Select the workflow element providing the authentication service for the user entry. For example, you can use Kerberos Authentication Provider workflow element or Local DB workflow element as the authentication provider.
- **Local Memory Workflow Element**
  - **Base DN**: Specify one or more base DNs for the data that is handled by the backend.
- **RDN Changing Workflow Element**
  - **Next Workflow Element**: Select the workflow element that should be next in the workflow.
  - **Object Class**: Select the object class type for RDN changing operation.
  - **Source RDN Attribute**: Select the original RDN attribute name from the source directory to be replaced or renamed in Oracle Unified Directory.
  - **Client RDN Attribute**: Select the new RDN attribute name to be used in Oracle Unified Directory.
  - **Replace RDN Value**: Specify whether the original RDN value should be replaced by the new RDN value. It is enabled by default.
  - **DN Attributes**: Click **Add** to select the list of attributes with DNs to perform RDN renaming on.
- **Transformations Workflow Element**
  - **Next Workflow Element**: Select the workflow element that should be next in the workflow.
  - **Entry Matching Filter**: This is an LDAP filter. If this option is selected then entries will be transformed only if they match this LDAP Filter.
  - **Entry Parent Suffixes**: This is optional, you can specify a list of suffixes to restrict applying transformation to entries under specific subtrees. If you specify this option then the entries will be specified only if they are in subtrees rooted at any of these suffixes.
  - **Excluded Operations**: If you specify this option, the entries will not be transformed during any of the specified operations
  - **Transformations**: The list of transformations that the Transformations Workflow Element will process. The order in which transformations are listed here does not guarantee the order in which these transformations will be applied when processing a request at runtime.)

6. Click **Create.**

   The following confirmation message is displayed:

```
Workflow Element created successfully.
```

#### 14.2.4.2 Display and Edit Workflow Element Properties

To display or modify the properties of an existing workflow element, follow these steps:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager."
2. Select the **Configuration** tab.
3. Select the **Core Configuration** view.

   For more information, see Section 14.2.1, "Selecting a Configuration View".
4. Expand the **Core Configuration** element.
5. Expand the **Workflow Elements** element.
6. Click on the workflow element that you want to view, or modify.

   The properties of the workflow element are displayed in the right hand pane
7. The properties that you can edit depend on the type of workflow element that is configured.
8. Click **Apply** to save your changes.

#### 14.2.4.3 Delete a Workflow Element

To delete an existing workflow element, follow these steps:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager."
2. Select the **Configuration** tab.
3. Select the **Core Configuration** view.

   For more information, see Section 14.2.1, "Selecting a Configuration View".
4. Expand the **Core Configuration** element.
5. Expand the **Workflow Elements** element.
6. Click on the workflow element that you want to delete and click the **Delete configuration** .
7. Click **OK** to confirm the deletion.

### 14.2.5 Configuring Workflows With ODSM

A workflow is defined by a naming context, or suffix, and a workflow element that define how Oracle Unified Directory should handle an incoming request. A workflow must be registered with at least one network group, but can be attached to several network groups.

For more information about workflows, workflow elements and the other components of Oracle Unified Directory, see Section 4.1, "Oracle Unified Directory Components."

The following sections describe how to configure workflows by using ODSM. For information about configuring workflows by using `dsconfig`, see Section 14.1.7, "Configuring Workflows With `dsconfig`".

#### 14.2.5.1 Create a Workflow

To create a workflow by using ODSM, follow these steps:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager."
2. Select the **Configuration** tab.
3. Select the **Core Configuration** view.

   For more information, see Section 14.2.1, "Selecting a Configuration View".
4. From the **Create** menu, select **Workflow.**
5. In the Workflow Properties region, enter the following information:
   a. In the **Name** field, type a name for the workflow that you want to create.
   b. Select the **Enabled** check box if you want this workflow to be enabled.

      Deselect this check box if you do not want to enable the workflow at this stage.
6. In the **Base DN** field, enter the naming context that will be accessible through this workflow.
7. Select the **Workflow Element** with which this workflow should be associated.

   The workflow element must already exist before you create the workflow.
8. Select **True**, **False**, or **Partial** depending on whether the workflow is critical enough to fail a search operation involving multiple workflows and if the operation fails on this workflow.
9. Click **Create.**

   The following confirmation message is displayed:

   ```
   Workflow created successfully.
   ```

#### 14.2.5.2 Display and Edit Workflow Properties

In the Core Configuration view, the Configuration tab displays all of the workflows and workflow elements that have been configured on the server.

To display the properties of a configured workflow, follow these steps:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager."
2. Select the **Configuration** tab.
3. Select the **Core Configuration** view.

   For more information, see Section 14.2.1, "Selecting a Configuration View".
4. Expand the **Workflows** element.
5. Click the workflow whose properties you want to display.

   The workflow properties are displayed in the right hand pane.
6. Make any required changes to the suffix configuration.

   You can disable the workflow, or change the workflow element with which this workflow is associated.

   Click **Apply** to save your changes.

#### 14.2.5.3 Delete a Workflow

You can delete a workflow by using ODSM, only if that workflow is not referenced by any network group.

To delete a workflow, follow these steps:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager."
2. If the workflow is referenced by a network group, modify the properties of the network group to remove that workflow.

   For more information, see Section 14.2.7.2, "Modify a Network Group".
3. Select the **Configuration** tab.
4. Select the **Core Configuration** view.

   For more information, see Section 14.2.1, "Selecting a Configuration View".
5. Expand the **Workflows** element.
6. Select the workflow that you want to delete and click the **Delete configuration** .
7. Click **OK** to confirm the deletion.

### 14.2.6 Configuring Connection Handlers With ODSM

Connection handlers are responsible for accepting connections from clients, reading and parsing requests submitted by the clients, ensuring that they are processed by the server, and sending the corresponding responses back to the client. A connection handler manages all communication with the client and therefore needs to implement support for the associated protocol.

The following sections describe how to configure connection handlers by using ODSM. For information about configuring connection handlers by using `dsconfig`, see Section 14.1.5, "Configuring Connection Handlers With `dsconfig`".

#### 14.2.6.1 Create a Connection Handler

To create a connection handler by using ODSM, follow these steps:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager."
2. Select the **Configuration** tab.
3. From the **Create** menu, select **Connection Handler**.
4. Select the type of connection handler that you want to create:
   - **LDAP.** This connection handler is used to interact with clients using LDAP. It provides full support for LDAPv3 and limited support for LDAPv2.
   - **LDAPS.** This connection handler is used to interact with clients using LDAP over SSL.
   - **LDIF.** This connection handler is used to process changes in the server using internal operations.
   - **JMX.** This connection handler allows interactions with clients using the Java Management Extensions (JMX) framework and the Remote Method Invocation (RMI) protocol.

- **SNMP.** This connection handler is used to process SNMP requests to retrieve monitoring information described by MIB 2605. The supported SNMP protocols are SNMP V1, V2c and V3.

5. Enter the properties to configure the connection handler in the right hand pane.

   The configurable properties will depend on the type of connection handler you have selected. For a comprehensive list of all configurable properties, and their allowed values, see "The Connection Handler Configuration" in the *Oracle Unified Directory Configuration Reference*.

6. When you have configured the required properties for your specific connection handler type, click **Create**.

   The following confirmation message is displayed:

   ```
   Connection Handler created successfully.
   ```

#### 14.2.6.2 Modify a Connection Handler

To view or modify connection handler properties by using ODSM, follow these steps:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager."
2. Select the **Configuration** tab.
3. Expand the **General Configuration** element.
4. Expand the **Connection Handlers** element.
5. Click on the connection handler whose properties you want to modify.

   The properties are displayed in the right hand pane.

   For a comprehensive list of all configurable properties, and their allowed values, see "The Connection Handler Configuration" in the *Oracle Unified Directory Configuration Reference*.

6. Change the required properties and click **Apply**.

#### 14.2.6.3 Delete a Connection Handler

To delete an existing connection handler by using ODSM, follow these steps:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager."
2. Select the **Configuration** tab.
3. Expand the **General Configuration** element.
4. Expand the **Connection Handlers** element.
5. Click on the connection handler that you want to delete and click the **Delete configuration**  .
6. You are prompted to confirm the deletion. Click **OK**.

### 14.2.7 Configuring Network Groups With ODSM

Network groups are the entry point of all client requests that are handled by an Oracle Unified Directory server. The properties of a network group indicate how client requests are directed.

The following sections describe how to configure network groups by using ODSM. For information about configuring network groups by using `dsconfig`, see Section 14.1.6, "Configuring Network Groups With `dsconfig`".

#### 14.2.7.1 Create a Network Group

To create a network group by using ODSM, follow these steps:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager."
2. Select the **Configuration** tab.
3. From the **Create** menu, select **Network Group**.
4. Enter the properties to configure the network group in the right hand pane.
   - **Name.** Enter a name for the network group.
   - **Enabled.** Select or deselect this check box to enable or disable the network group. If you disable a network group, no client requests can be handled by that network group. If you disable the only configured network group, you effectively stop client applications from accessing the backend.
   - **Priority.** In the event of multiple network groups, set priority for this network group. Client requests are handled by the network group with the highest priority, for which the criteria are met. The highest priority a network group can have is 0.
   - **Workflow.** Click the **Add** ( Add ) to add one or more workflows that can be accessed through this network group.
   - **Root DSE to Expose.** Select the Root DSE that you want this network group to expose. You can expose the Root DSE of the local server, the Root DSE stored in a remote server, or the Root DSE defined in a local file.

     Click **Other** and select one of the following option:

     – **Root DSE Defined in LDIF File**: Enter the path of the LDIF file containing the Root DSE. The server must have access to this file.

     – **Root DSE of a Remote Server**: Enter the following parameters:

       **Host Name**: Enter the host name of the remote server.

       **Ports Available**: Enter the LDAP port, LDAPS port, or LDAP and LDAPS ports of the remote server.

       **Trust All**: Select this check box to trust all the certificates presented by the remote server.

       **Trust Manager**: Select the trust manager that the server will use when connecting to the LDAPS ports of the remote server to forward requests.
   - **Security Mandatory.** Select this option if you require clients to use a secure connection to access this network group. By default, a secure connection is not required.
   - **Allowed auth method.** Specify the authentication method/s that are allowed between the client and the network group.
   - **Allowed protocol.** Specify the protocol/s that are allowed for client connections. If you do not specify a protocol, all protocols are allowed.

- **Allowed BindDN.** Click the Add to add one or more bind DNs that are allowed to connect to this network group. Click the **Delete** (  ) to remove the bind DNs that should not be accepted by the network group.
- **Allowed Client.** Click the Add to add one or more clients that are authorized to access this network group. Clients can be expressed by their IP addresses or names, or by a subnet mask. If no allowed client list is provided, all clients are allowed, unless they are specifically listed on the denied client list.
- **Denied Client.** Click the Add to add one or more clients that are prohibited from accessing this network group. Clients can be expressed by their IP addresses or names, or by a subnet mask. If no denied client list is provided, all clients are allowed, unless a limitation is set by using the allowed client list.
- **QoS Policy.** Select a quality of service policy for this network group. For more information, see Section 14.1.6.3, "Creating a Network Group Quality of Service Policy."

5. When you have configured the required properties for the network group, click **Create**.

   The following confirmation message is displayed:

   ```
   Network Group created successfully.
   ```

#### 14.2.7.2 Modify a Network Group

You can display or modify the properties of a network group, as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager."
2. Select the **Configuration** tab.
3. Expand the **General Configuration** element.
4. Expand the **Network Groups** element.
5. Select the network group whose properties you want to modify.

   The properties of the network group are displayed in the right hand pane
6. Change the required properties and click **Apply**.

   For an explanation of each of the configured properties, see Section 14.2.7.1, "Create a Network Group".

#### 14.2.7.3 Delete a Network Group

To delete an network group by using ODSM, follow these steps:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager."
2. Select the **Configuration** tab.
3. Expand the **General Configuration** element.
4. Expand the **Network Groups** element.
5. Click on the network group that you want to delete and click the **Delete configuration**  .
6. You are prompted to confirm the deletion. Click **OK**.

### 14.2.8 Modify the General Server Configuration

Certain elements of the general server configuration can be modified by using ODSM, as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager."
2. Select the **Configuration** tab.
3. Select the **General Configuration** element.

   The properties are displayed in the right hand pane.
4. You can modify the following properties:
   - **Default Password Policy**
   - **Etime Resolution**
   - **Idle Time Limit**
   - **Max Allowed Client Connections**
   - **Maintain Authenticated Users**
   - **Reject Unauthenticated Requests**
   - **Size Limit**
   - **Writability Mode**
   - **Root DSE Properties**
   - **Work Queue Properties**
   - **Number of Worker Threads**

   Click **Apply** to save your changes.

   For a comprehensive list of the configurable properties, and their allowed values, see "Global Configuration" in the *Oracle Unified Directory Configuration Reference*.

## 14.3 Managing Administration Traffic to the Server

Connection handlers are responsible for handling all interaction with client applications, including accepting connections, reading requests, and sending responses.

Oracle Unified Directory includes a special connection handler, the administration connector, to manage administration traffic to the server. The administration connector enables the separation of user traffic and administration traffic to simplify monitoring, and to ensure that administrative commands take precedence over commands that manipulate user data.

This section describes how administration traffic is handled, and covers the following topics:

- Section 14.3.1, "Overview of the Administration Connector"
- Section 14.3.2, "Accessing Administrative Suffixes"
- Section 14.3.3, "To Configure the Administration Connector"
- Section 14.3.4, "Key Managers and Trust Managers for the Administration Connector"

### 14.3.1 Overview of the Administration Connector

The administration connector is based on the LDAP protocol and uses LDAP over SSL by default. All command-line utilities that access the administrative suffixes use the administration connector. This includes the following commands:

- `backup`
- `dsconfig`
- `dsreplication`
- `export-ldif`
- `import-ldif`
- `manage-account`
- `manage-tasks`
- `restore`
- `status`
- `stop-ds`
- `uninstall`

The administration connector is always present and enabled. You cannot disable or delete the connector but you can use `dsconfig` to manipulate the following properties of the connector:

- `listen-address`. The address on which the server listens for administration traffic.
- `listen-port`. The default port of the administration connector is 4444. You can change this port during setup if required. If you use the default port, you do not need to specify a port when running the administration commands (the default port is assumed). If you change the port, you must specify the new port when running the administration commands.

  If you have multiple directory server instances running on the same host, you will have specified multiple separate administration listen ports during setup. In this case, for the server instances whose administration connectors do not use the default listen port (4444), you will need to specify the port when running the administration commands.
- **Security-related properties.** Traffic using the administration connector is always secured. As with the LDAPS connection handler, the administration connector is configured with a self-signed certificate (`admin-cert`) during server setup. This self-signed certificate is generated the first time the server is started. You can manage the administration connector certificate using external tools, such as `keytool`.

  The security-related properties of the administration include the following:

  - `ssl-cert-nickname`
  - `ssl-cipher-suite`
  - `key-manager-provider`
  - `trust-manager-provider`

  When you run the administration commands, you are prompted as to how you want to trust the certificate. If you run the administration commands in

non-interactive mode, you must specify the `-X` or `--trustAll` option to trust the certificate, otherwise the command will fail.

## 14.3.2 Accessing Administrative Suffixes

The *administrative suffixes* include the following:

- `cn=config`
- `cn=monitor`
- `cn=tasks`
- `cn=backups`
- `cn=ads-truststore`
- `cn=schema`
- `cn=admin data`

In general, direct LDAP access to the administrative suffixes (using the `ldap*` utilities) is discouraged, with the exception of the `cn=monitor` suffix. In most cases, it is preferable to use the dedicated administrative command-line utilities to access these suffixes.

If you must use the `ldap*` commands to access the administrative suffixes, you must use the administration connector port (with the `--useSSL` or `-Z` option). Using the administration connector ensures that monitoring data is not polluted and that server administration takes precedence over user traffic. The same restriction applies if you are accessing the administrative suffixes using an LDAP browser.

## 14.3.3 To Configure the Administration Connector

This example displays the default properties of the administration connector, and changes the listen port of the connector to `5555`.

1. View the default properties of the administration connector, using the `dsconfig` command.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  get-administration-connector-prop
```

The output is similar to the following.

```
Property               : Value(s)
-----------------------:---------------
key-manager-provider   : Administration
listen-address         : 0.0.0.0
listen-port            : 4444
ssl-cert-nickname      : admin-cert
ssl-cipher-suite       : -
trust-manager-provider : Administration
```

2. Change the listen port, using the `dsconfig` command.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
 set-administration-connector-prop --set listen-port:5555
```

**Note:** You must restart the server for changes to this property to take effect.

### 14.3.4 Key Managers and Trust Managers for the Administration Connector

The administration connector is an LDAPS connector. As with all SSL-based connectors, the administration connector requires a key manager and trust manager.

Oracle Unified Directory provides a dedicated key manager and trust manager for the administration connector, that are enabled by default. You can change the properties of the default administration key manager and trust manager. For more information, see Section 20.2, "Configuring Key Manager Providers" and Section 20.3, "Configuring Trust Manager Providers".

## 14.4 Configuring Commands As Tasks

Certain command-line utilities can be used to schedule tasks to run within the directory server as well as to perform their functions locally. Tasks that can be scheduled support the options used to connect to the directory server to interact with the task back end.

This section contains the following topics:

- Section 14.4.1, "Commands That Can Schedule Tasks"
- Section 14.4.2, "Controlling Which Tasks Can Be Run"
- Section 14.4.3, "Scheduling and Configuring Tasks"
- Section 14.4.4, "Managing and Monitoring Scheduled Tasks"

### 14.4.1 Commands That Can Schedule Tasks

The following utilities can schedule tasks:

- `import-ldif`
- `export-ldif`
- `backup`
- `restore`
- `stop-ds`
- `stop-ds --restart`
- `rebuild-index`
- `dsreplication purge-historical`

For a proxy server, only the `stop-ds` command can be scheduled to run as a task.

### 14.4.2 Controlling Which Tasks Can Be Run

You can control the tasks that can be run by setting the `allowed-tasks` advanced global configuration property. By default, all tasks supported by the tasks back end are allowed. To prevent a task from being run, remove its value from the `allowed-tasks` property. For example, to prevent the server from being stopped using a task, run the following command:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  set-global-configuration-prop --remove \
  allowed-task:org.opends.server.tasks.ShutdownTask
```

## 14.4.3 Scheduling and Configuring Tasks

The procedures in this section indicate how to schedule a task, how to configure task notification, and how to configure task dependencies. All of the examples in this section assume that the commands are being run on the local host, using the default administration port (4444), and the local certificate configuration. If you are running the commands remotely, you might need to specify the certificate parameters. For more information, see Section 14.3, "Managing Administration Traffic to the Server."

This section describes procedures to schedule and configure tasks, and contains the following topics:



### 14.4.3.1 To Schedule a Task

To schedule a task, invoke the required utility with the options used to connect to the directory server, an optional start time, and any options that will be used as arguments for the task execution.

If the `-t` or `--start` option is provided, the utility exits immediately after scheduling the task. To schedule a task for immediate execution and have the utility exit immediately after scheduling the task, specify `0` as the value for the start time.

If the `-t` or `--start` option is omitted, the utility schedules the task for immediate execution and tracks the task's progress, printing log messages as they are available and exiting when the task has completed.

Schedule the `export-ldif` task to start at 12:15 on September 24th, 2009.

```
$ export-ldif -D "cn=directory manager" -j pwd-file \
  -l /ldif-files/example.ldif --start 20090924121500 -n userRoot
```

### 14.4.3.2 To Schedule a Recurring Task

To schedule a recurring task, invoke the required utility with the options used to connect to the directory server, specifying the recurring task schedule, and any options that will be used as arguments for the task execution. The following commands can be scheduled as recurring tasks:

- `import-ldif`
- `export-ldif`
- `backup`
- `restore`
- `rebuild-index`
- `dsreplication purge-historical`

The `--recurringTask` option specifies a recurring task schedule that is used by the task scheduler to determine when and how often a recurring task should run. The pattern used to specify the schedule is based on UNIX `crontab`(5) scheduling patterns and rules and includes the following five integer pattern fields, separated by blank spaces:

- Minute [0,59]

- Hour [0,23]
- Day of the month [1,31]
- Month of the year [1,12]
- Day of the week [0,6] (with 0=Sunday)

Each of these patterns can be either an asterisk (meaning all valid values), an element, or a list of elements separated by commas. An element is either a number or two numbers separated by a dash (meaning an inclusive range).

The task scheduler spawns regular task iterations according to the specified schedule.

Schedule the task using the `--recurringTask` option.

The following command schedules a backup task to execute at the beginning of every hour.

```
$ backup -D "cn=directory manager" -j pwd-file --recurringTask \
  "00 * * * *" --backupDirectory /example/backup --backUpAll --backupID "Hourly
Backup"
```

The following example shows an export task that is scheduled to run every 15 minutes, every Sunday.

```
$ export-ldif -D "cn=directory manager" -j pwd-file --recurringTask \
  "0,15,30,45 * * * 0" -l PATH/export-recurring.ldif -n userRoot
Recurring Export task  ExportTask-a614e45d-6ba5-4c29-a8e1-d518c20e46ab scheduled
successfully
```

#### 14.4.3.3 To Configure Task Notification

The task scheduling options of a utility enable you to notify an administrator when a task completes or if an error occurs during the task's execution. To use the notification facility, an SMTP server must be configured for the directory server.

1. Specify an SMTP server by setting the `smtp-server` global configuration property.

   The following command configures the SMTP server named `mailserver.example.com`:

   ```
   $ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
     set-global-configuration-prop --set smtp-server:mailserver.example.com
   ```

2. Use the `completionNotify` and `errorNotify` options to specify the email address to which the task notification should be sent.

   The following command schedules a `backup` task and specifies that `admin@example.com` should be notified when the task completes, or when an error occurs:

   ```
   $ backup -D "cn=directory manager" -j pwd-file -a -d /tmp/backups \
     --start 20080924121500 --completionNotify admin@example.com \
     --errorNotify admin@example.com
   Backup task 20080924121500 scheduled to start Sep 24, 2008 12:15:00 PM SAST
   ```

#### 14.4.3.4 To Configure Task Dependencies

Certain tasks might require that another task be completed before the task begins. The task dependency options of a utility enable you to specify that the task depends on another task, and what the task should do should the other task fail.

Schedule the task and specify the `dependency` and `failedDependencyAction`.

The following example schedules a backup task that depends on another task, and specifies that the backup should be canceled should the other task fail:

```
$ backup -D "cn=directory manager" -j pwd-file -a -d /tmp/backups \
  --start 2008102914530410 --dependency 20080924121500 \
  --failedDependencyAction cancel
Backup task 2008102914530410 scheduled to start Oct 29, 2008 14:53:04 PM SAST
```

## 14.4.4 Managing and Monitoring Scheduled Tasks

The `manage-tasks` utility can be used to obtain a list of scheduled tasks, to display task status, and to cancel scheduled tasks. The following procedures provide examples of managing scheduled tasks:

- Section 14.4.4.1, "To Obtain Information About Scheduled Tasks"
- Section 14.4.4.2, "To Cancel a Scheduled Task"
- Section 14.4.4.3, "To Cancel a Recurring Task"

### 14.4.4.1 To Obtain Information About Scheduled Tasks

1. Display a summary of all scheduled tasks.

```
$ manage-tasks -D "cn=directory manager" -j pwd-file -n -s
ID               Type    Status
-------------------------------------------------
2008100912550010  Backup  Completed successfully
2008100912554710  Backup  Completed successfully
2008100912560510  Backup  Waiting on start time
2008100912561410  Backup  Waiting on start time
```

2. Display additional information on a particular task, specified by its task ID.

```
$ manage-tasks -D "cn=directory manager" -j pwd-file -n -i 2008100912550010

Task Details
-----------------------------------------------------
ID                        2008100912550010
Type                      Backup
Status                    Completed successfully
Scheduled Start Time      Immediate execution
Actual Start Time         Oct 9, 2008 12:55:00 PM SAST
Completion Time           Oct 9, 2008 12:55:01 PM SAST
Dependencies              None
Failed Dependency Action  None
Email Upon Completion     None Specified
Email Upon Error          None Specified

Backup Options
----------------------------
Backup All        true
Backup Directory  ../backups

Last Log Message
------------------------------------------------------------------------------
[09/Oct/2008:12:55:01 +0200] severity="NOTICE" msgCount=4 msgID=10944795
message="The backup process completed successfully"
```

### 14.4.4.2 To Cancel a Scheduled Task

Run the `manage-tasks` utility with the `-c` or `--cancel` option.

The following command cancels a particular task, specified by its task ID:

```
$ manage-tasks -D "cn=directory manager" -j pwd-file -n -c 2008100912561410
```

#### 14.4.4.3 To Cancel a Recurring Task

You can cancel an entire recurring task, in which case both the recurring task and its next scheduled iteration are canceled. Alternatively, you can cancel only the next scheduled task iteration, in which case future recurring task iterations will be spawned by the task scheduler.

1. Use the manage-tasks command to display the summary of scheduled tasks.

```
$ manage-tasks -D "cn=directory manager" -j pwd-file -n -s

 ID                                              Type    Status
------------------------------------------------------------------------------
 Hourly Backup                                   Backup  Recurring
 Hourly Backup - Wed Jan 14 13:00:00 SAST 2009   Backup  Waiting on start time
```

2. Run the `manage-tasks` utility with the `-c` or `--cancel` option.

   a. Cancel the entire recurring task by specifying its task ID.

```
$ manage-tasks -D "cn=directory manager" -j pwd-file -n -c "Hourly Backup"
Task Hourly Backup canceled
```

   b. Cancel the next scheduled task by specifying its task ID.

```
$ manage-tasks -D "cn=directory manager" -j pwd-file -n \
  -c "Hourly Backup - Wed Jan 14 13:00:00 SAST 2009 "
Task Hourly Backup - Wed Jan 14 13:00:00 SAST 2009  canceled
```

## 14.5 Deploying and Configuring the DSML Gateway

The Directory Services Markup Language (DSML) is a SOAP-based mechanism that can communicate with directory servers using an XML-based representation instead of the LDAP protocol. Oracle Unified Directory 11*g* Release 2 PS1 (11.1.2.1.0) supports the use of DSML through a web application that acts as a DSML-to-LDAP gateway, in which clients communicate with the gateway using DSML, but the gateway communicates with the directory server through LDAP.

This section describes how to configure and deploy the DSML gateway, and contains the following topics:

- Section 14.5.1, "Deploying the DSML Gateway"
- Section 14.5.2, "Confirming the DSML Gateway Deployment"

### 14.5.1 Deploying the DSML Gateway

The DSML gateway can be deployed like any other web application, in most common application containers. The following section describes how to deploy the DSML gateway in Oracle WebLogic Server 10.3.5, on a UNIX system.

- Section 14.5.1.1, "Deploying the DSML Gateway in Oracle WebLogic Server"
- Section 14.5.1.2, "Deploying the DSML Gateway in IBM WebSphere"

#### 14.5.1.1 Deploying the DSML Gateway in Oracle WebLogic Server

This section assumes that you have Oracle WebLogic Server installed. If you do not, install Oracle WebLogic Server, as described in "Installing Oracle WebLogic Server" in *Oracle Fusion Middleware Installation Guide for Oracle Unified Directory*.

**Configuring WebLogic Server for the DSML Gateway**

1. Run the configuration wizard from the following location:

   *OUD_BASE_LOCATION_HOME*/wlserver_10.3/common/bin/config.sh

2. On the Welcome screen, select **Create a new WebLogic domain** and click **Next**.
3. On the Select Domain Source screen, accept the default selection (**Basic WebLogic Server Domain**) and click **Next**.
4. On the Specify Domain Name and Location screen, type a domain name and specify its location.

   A new WebLogic domain is created in this location. The DSML gateway will be deployed into this domain.

5. On the Configure Administrator User Name and Password screen, type a name and password for the user who will administer this domain.

   The password must be at least eight characters and must contain at least one number or special character. Confirm the password and click **Next.**

   Make a note of these details as you will need them to start or restart the WebLogic domain.

6. On the Configure Server Start Mode screen, select **Production Mode.**

   Select a valid JDK (at least Java 1.6) and click **Next.**

7. On the Optional Configuration screen, click **Next.**
8. On the Configuration Summary screen, verify the domain details and click **Create.**
9. On the Creating Domain Screen, click **Done.**
10. Set the Java options for the WebLogic Server.

    ```
    $ export
    JAVA_OPTIONS=-Djavax.xml.soap.MessageFactory=weblogic.xml.saaj.MessageFactoryIm
    pl
    ```

    If you do not set the Java options, an error will be returned.

11. Set the `enforce-valid-basic-auth-credentials` flag in the configuration file of the WebLogic domain (*DOMAIN_HOME*/config/config.xml, where `DOMAIN_HOME` is the domain that you created in Step 4.

    For example, edit the file *OUD_BASE_LOCATION_HOME*/user_projects/domains/base_domain/config/config.xml by adding the following line to the `security-configuration` element:

    ```
    <enforce-valid-basic-auth-credentials>false</enforce-valid-ba
    sic-auth-credentials>
    ```

    For more information, see http://download.oracle.com/docs/cd/E12840_01/wls/docs103/security/thin_client.html#understanding_basic_atn.

12. Start the WebLogic Server by running *DOMAIN_HOME*/bin/startWebLogic.sh (where `DOMAIN_HOME` is the domain that you created in Step 4.

    For example:

    *OUD_BASE_LOCATION_HOME*/user_projects/domains/base_domain/bin/startWebLogic.sh

13. Deploy the DSML Gateway WAR file, as described in the following section.

**Deploying the DSML Gateway WAR File**

1. Create a `DSML` directory in the addons directory and change to that directory.

```
$ cd OUD_BASE_LOCATION_HOME/ORACLE_HOME/addons
$ mkdir DSML
$ cd DSML
```

2. Explode the DSML gateway WAR file.

```
$ jar xvf ../OUD-DSML.zip
```

3. Edit the DSML configuration, if required.

   The WEB-INF/web.xml file includes initialization parameters that can be used to specify the address (in the `ldap.host` parameter) and port number (in the `ldap.port` parameter) of the directory server to which DSML requests should be forwarded.

   By default, the DSML gateway is configured to communicate with a directory server on the same system, that is, `localhost`) on port `389`. If you need to change the host address and port number, edit the `web.xml` file and restart the web container.

4. In a browser window, connect to the WebLogic Administration Console (for example `http://hostname:7001/console`), where `hostname` is the host on which WebLogic Server is running.

   Use the administrator user name and password that you established in Step 5 of the preceding procedure.

5. Follow the WebLogic Server Documentation to install a Web application (http://download.oracle.com/docs/cd/E12840_01/wls/docs103/ConsoleHelp/taskhelp/web_applications/InstallWebApplications.html).

   - In step 4 of the procedure, provide the path to the exploded application (*OUD_BASE_LOCATION_HOME/ORACLE_HOME*/addons/DSML).
   - In step 6 of the procedure, select **Install this deployment as an application**.
   - Accept the default values for the other steps.

6. On the left panel of the Administration Console, click **Deployments**.
7. Select the checkbox next to the DSML application and click **Start** then **Servicing all requests**.
8. On the Start Deployments panel, click **Yes**.
9. The DSML application is now deployed and available for use.

#### 14.5.1.2 Deploying the DSML Gateway in IBM WebSphere

To deploying the DSML Gateway WAR File in IBM Websphere, complete the following:

**Note:** Ensure that you have installed and configured IBM WebSphere, as described in the "Configuring IBM WebSphere for Oracle Directory Services Manager" in *Oracle Fusion Middleware Installation Guide for Oracle Unified Directory*.

1. Create a `DSML` directory in the addons directory and change to that directory.

   ```
   $ cd OUD_BASE_LOCATION_HOME/ORACLE_HOME/addons
   $ mkdir DSML
   $ cd DSML
   ```

2. Explode the DSML gateway WAR file.

   ```
   $ jar xvf ../OUD-DSML.war
   ```

3. Edit the DSML configuration, if required.

   The WEB-INF/web.xml file includes initialization parameters that can be used to specify the address (in the `ldap.host` parameter) and port number (in the `ldap.port` parameter) of the directory server to which DSML requests should be forwarded.

   By default, the DSML gateway is configured to communicate with a directory server on the same system, that is, `localhost`) on port `389`. If you need to change the host address and port number, edit the `web.xml` file and restart the web container.

4. In a browser, log in to the IBM WebSphere Administrative Console.

   ```
   http://Hostname:Port-Number/ibm/console/
   ```

5. Choose **Application**, and then **New Application** in the left panel.
6. Click **New Enterprise Application**.

   The **Preparing for the application installation** page is displayed.

   a. Specify the path to the `OUD-DSML.war` file and click **Next**.
      - If the `OUD-DSML.war` file is located on the local system, type its complete path under Local File System.
      - If the file is on a remote IBM WebSphere machine, specify the path for the remote file system.

   b. Select **Detailed - Show me all installation options and parameters**.

   c. On the **Choose to generate default bindings and mappings page** keep the default configuration and click **Next**.

7. Click **Continue** on the Application Security Warning page.

   The **Select Installation Options** page is displayed.

8. Enter the following details:

   a. To install the application to a location other than the default location, type the path in the **Directory to Install Application** field. For example, on UNIX systems:

      ```
      /opt/IBM/Websphere/AppServer/installed Apps/Hostname
      ```

   b. Verify that the following options are selected:

- Precompile JavaServer Pages files
- Distribute application
- User binary configuration

c. Enter the name of the application in the Application Name field, and click **Next**. The default application name is set to **OUD-DSML.war**.

9. On the **Map modules to servers** page, verify that the Oracle Unified Directory application mapping is set to the appropriate cluster/server and click **Next**.

   The **Provide options to compile JSPs** page is displayed.

10. Make the following changes and click **Next**:

    a. Select Web Module **Oracle Unified Directory**.

    b. Change JDK Source Level to 15.

11. On the **Provide JSP reloading options for Web Modules** page keep the default configuration.

12. On the **Map shared libraries** page, verify your settings and click **Next**.

    On the **Map shared library relationships** page, verify your settings and click **Next**.

    On the **Map virtual hosts for Web modules** page, verify that the Oracle Unified Directory application mapping is set to the appropriate virtual host and click **Next**.

13. On the **Map context roots for Web modules** page, provide context root `/DSML` and click **Next**.

14. Choose **Applications > Application Types > WebSphere Enterprise Applications > OUD-DSML_war > Manage Modules**, and do the following:

    a. Verify that the cluster/server mapping is correct.

    b. Select **OUD-DSML.war > Class loader order**.

    c. Under Class loader order, select **Class loaded with local class loader first (parent last)**, and save your changes.

    **Note:** When running two or more applications in the same cell, it is important that you configure the application server such that the application class loader can override the parent and provide its own version of a class. For WebSphere, choose **Class loaded with local class loader first (parent last)**, as documented above.

15. Restart IBM WebSphere.

## 14.5.2 Confirming the DSML Gateway Deployment

After the DSML gateway has been deployed and configured, you can communicate with it by using any DSMLv2 client. The following sections describe two ways to accomplish this:

- Section 14.5.2.1, "To Confirm the DSML Gateway Deployment with JXplorer"
- Section 14.5.2.2, "Confirming the DSML Gateway Deployment with the Directory Server Resource Kit"

#### 14.5.2.1 To Confirm the DSML Gateway Deployment with JXplorer

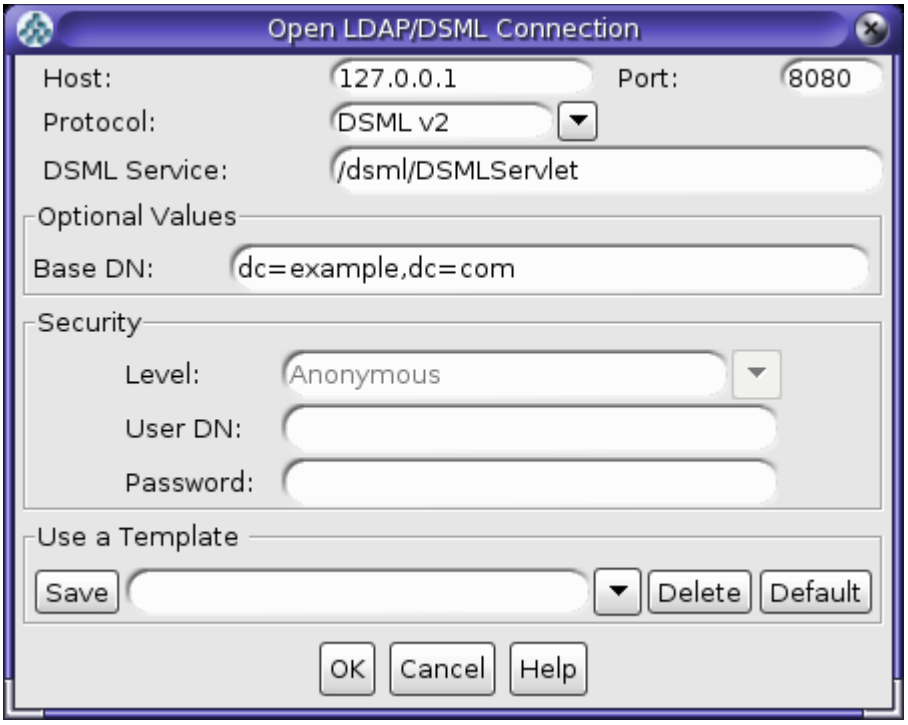
The JXplorer tool is a Java-based LDAP browser that can be used to browse, search, and edit the contents of an Oracle Unified Directory instance. This tool can communicate using both LDAP and DSML. Although JXplorer's DSML support does not allow authentication (and therefore is restricted to the set of operations available to anonymous users), it is still possible to use it to verify that the DSML gateway is functioning as expected.

You can download JXplorer, and the accompanying documentation, at jxplorer.org.

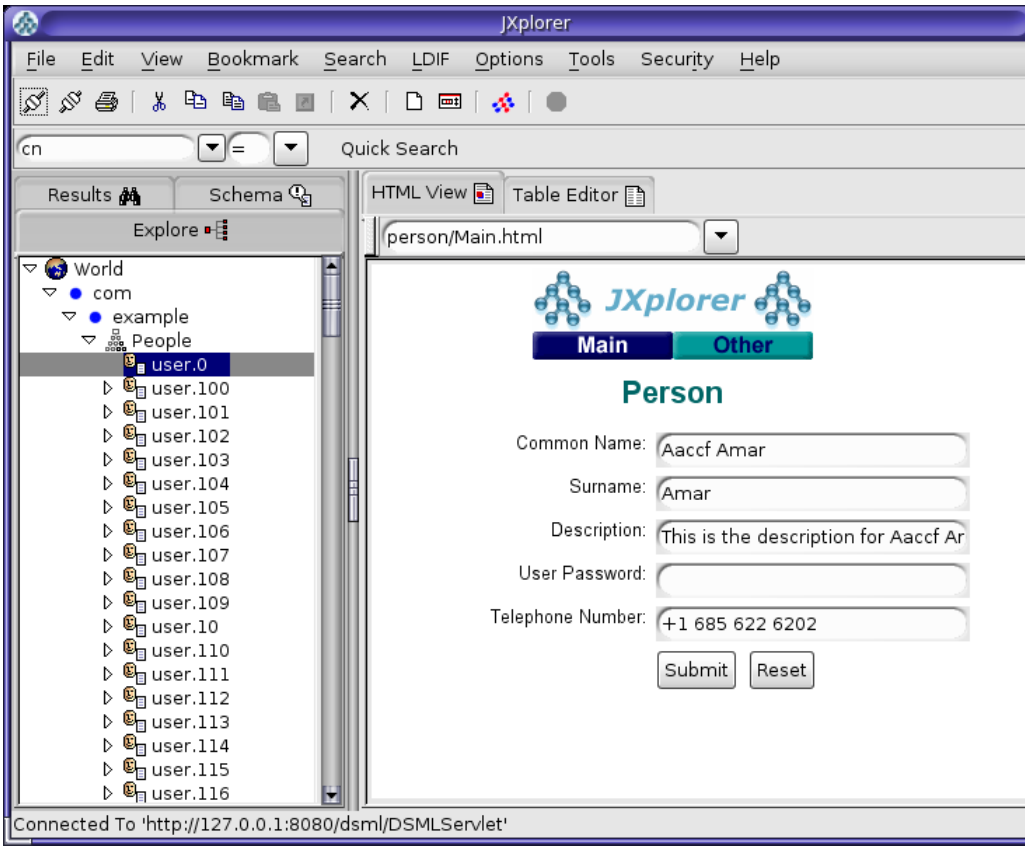
To confirm a DSML gateway by using JXplorer, follow these steps:

1. Start JXplorer and chose the Connect option from the File menu.

   The Open LDAP/DSML Connection window opens with fields for connection information. The following figure shows typical entries.




2. Enter the address and port number of the Web application on which the DSML gateway is running.
3. Choose `DSMLv2` from the Protocol list.
4. Specify the path to the DSMLServlet in the DSML Service field.
5. Provide an appropriate base DN value for your directory.
6. Click OK to connect the directory server and display a JXplorer window where you can search and browse the tree (with the limitations imposed for anonymous users).




### 14.5.2.2 Confirming the DSML Gateway Deployment with the Directory Server Resource Kit

The Directory Server Resource Kit (DSRK) is a collection of utilities that can be used in conjunction with directory servers. The DSRK was originally intended for use with Oracle Directory Server Enterprise Edition, but in most cases the applications also work with Oracle Unified Directory. The most recent version of the DSRK is included as part of Oracle Directory Server Enterprise Edition 11g Release 1 PS1 (11.1.1.7.0), and contains the `dsmlsearch` and `dsmlmodify` tools that can interact with a directory server using DSML rather than LDAP.

Note that even though an older version of these DSML tools was provided with earlier versions of the Directory Server Resource Kit, the version provided with Oracle Directory Server Enterprise Edition 11g Release 1 PS1 (11.1.1.7.0) is strongly recommended because it is easier to use. You can download Oracle Directory Server

Enterprise Edition 11g Release 1 PS1 (11.1.1.7.0) from Oracle Technology Network (OTN) here:

```
http://www.oracle.com/technetwork/middleware/downloads/oid-11g-161194.html
```

#### 14.5.2.2.1 Using the `dsmlsearch` Command

The `dsmlsearch` command is a DSML-based counterpart to the `ldapsearch` command. `dsmlsearch` operates in a similar manner to `ldapsearch` but there are certain key differences. To see usage information, invoke the command with no arguments, as in the following example:

```
$ ./dsmlsearch
usage: dsmlsearch -h http://host:port -b basedn [options] filter [attributes...]
where:
-h hostURL URL of the directory server
-b basedn  base dn for search
-D binddn  bind dn
-w passwd  bind password (for simple HTTP authentication)
use "-w - " to prompt for a password
-j pwfile  file where password is stored
-s scope   specify the scope of the search
baseObject - For searching only the base entry
singleLevel - For searching only the children
wholeSubtree - For searching the base entry and all childrens
-a deref   specify how aliases are deferenced
neverDerefAliases - Aliases are never dereferenced
derefFindingBaseObj - Dereferenced when finding the base DN
derefAlways - Dereferenced when finding below the base DN
-l seconds specify the maximum number of seconds to wait for the search
-z number  specify the maximum number of entries to return for the search
-f file    specify the name of the file containing the search filter
```

The `dsmlsearch` command differs in usage from `ldapsearch`:

- The `-h` argument is used to provide a URL to use to access the server. It should include the host and port number, as well as the URI for the gateway servlet (for example, `http://127.0.0.1:8080/dsml/DSMLServlet`).
- The `-b` argument is used to specify the search scope, but note that the values you provide are different (`baseObject`instead of `base`, `singleLevel`instead of `one`, and `wholeSubtree`instead of `sub`).
- The results are output in DSML format, which is not as user-friendly or human-readable as the LDIF output provided by `ldapsearch`.

An example usage of this tool is as follows. Note that the DSML output does not contain any line breaks, but line breaks are added here for readability.

```
$ ./dsmlsearch -h http://127.0.0.1:8080/dsml/DSMLServlet \-b "dc=example,dc=com"
-s baseObject \"(objectClass=*)"
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Body><dsml:batchResponse xmlns:dsml="urn:oasis:names:tc:DSML:2:0:core">
<dsml:searchResponse><dsml:searchResultEntry dn="dc=example,dc=com"><dsml:attr
name="objectClass"><dsml:value>domain</dsml:value><dsml:value>top</dsml:value>
</dsml:attr><dsml:attr name="dc"><dsml:value>example</dsml:value></dsml:attr>
</dsml:searchResultEntry><dsml:searchResultDone><dsml:resultCode code="0"/>
</dsml:searchResultDone></dsml:searchResponse></dsml:batchResponse>
</SOAP-ENV:Body></SOAP-ENV:Envelope>
```

#### 14.5.2.2.2 Using the `dsmlmodify` Utility

`dsmlmodify` utility is a DSML-based counterpart to the `ldapmodify` command, and it can perform add, delete, modify, and modify DN operations over DSML. To see the usage information for this tool, run it with no arguments, as shown in this example:

```
$ ./dsmlmodify
usage: dsmlmodify -h http://host:port [options] -f file
where:
-h hostURL URL of the directory server
-D binddn  bind dn
-w passwd  bind password (for simple HTTP authentication)
use "-w - " to prompt for a password
-j pwfile  file where password is stored
-f file    specify the name of the file containing
the modifications
```

As with the `dsmlsearch` utility, the `-h` argument specifies a URL, and the output is returned in DSML form. Unlike `ldapmodify`, the `dsmlmodify` tool does not accept the changes through standard input. Changes must be specified in a file, and that file must be in DSML format instead of than LDIF, and the changes cannot contain an outer `batchRequest` wrapper. The following example shows a typical input file.

```
<addRequest dn="uid=test.user,dc=example,dc=com">
<attr name="objectClass">
<value>top</value>
<value>person</value>
<value>organizationalPerson</value>
<value>inetOrgPerson</value>
</attr>
<attr name="uid">
<value>test.user</value>
</attr>
<attr name="givenName">
<value>Test</value>
</attr>
<attr name="sn">
<value>User</value>
</attr>
<attr name="cn">
<value>Test User</value>
</attr>
<attr name="userPassword">
<value>password</value>
</attr>
</addRequest>
<modifyRequest dn="uid=test.user,dc=example,dc=com">
<modification name="description" operation="replace">
<value>This is the new description</value>
</modification>
</modifyRequest>
<modDNRequest dn="uid=test.user,dc=example,dc=com" newrdn="cn=Test User"
 deleteoldrdn="false" newSuperior="ou=People,dc=example,dc=com" />
<delRequest dn="cn=Test User,ou=People,dc=example,dc=com" />
```

The following example shows the output from applying these changes. Line breaks have been added to the output to make it more readable:

```
$ dsmlmodify -h http://127.0.0.1:8080/dsml/DSMLServlet \  -D "cn=Directory
Manager" -j pwd-file -f /tmp/test.dsml
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Body><dsml:batchResponse xmlns:dsml="urn:oasis:names:tc:DSML:2:0:core">
<dsml:addResponse><dsml:resultCode code="0"/></dsml:addResponse>
```

```
<dsml:modifyResponse><dsml:resultCode code="0"/></dsml:modifyResponse>
<dsml:modDNResponse><dsml:resultCode code="0"/></dsml:modDNResponse>
<dsml:delResponse><dsml:resultCode code="0"/><dsml:errorMessage>The number of
entries deleted was 1</dsml:errorMessage></dsml:delResponse></dsml:batchResponse>
</SOAP-ENV:Body></SOAP-ENV:Envelope>

$ dsmlmodify -h http://localhost:8080/dsml/DSMLServlet \  -D "cn=directory
manager" -j pwd-file -f /tmp/dsml.ldif
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Body><batchResponse xmlns="urn:oasis:names:tc:DSML:2:0:core">
<addResponse><resultCode code="0"/></addResponse>
<modifyResponse><resultCode code="0"/></modifyResponse>
<modDNResponse><resultCode code="0"/></modDNResponse>
<delResponse><resultCode code="0"/></delResponse></batchResponse>
</SOAP-ENV:Body></SOAP-ENV:Envelope>
```

# 15

# Configuring the Proxy Components

This chapter describes how to configure the server elements that are specific to a proxy instance. Note that many of these elements are configured automatically when you configure a load balancing or distribution topology while setting up a proxy instance.

This chapter covers the following topics:

- Section 15.1, "Managing the Proxy Configuration With `dsconfig`"
- Section 15.2, "Managing the Proxy Configuration With ODSM"

For more information about the `dsconfig` command, see Section 14.1, "Managing the Server Configuration With `dsconfig`".

For more information about ODSM, see Chapter 18, "Accessing Oracle Unified Directory by Using Oracle Directory Services Manager."

## 15.1 Managing the Proxy Configuration With `dsconfig`

This section describes the procedures to manage a proxy configuration with the `dsconfig` command, and covers the following topics:

- Section 15.1.1, "Configuring Communication With Remote LDAP Servers"
- Section 15.1.2, "Configuring the Bind Mode"
- Section 15.1.3, "Configuring Load Balancing With `dsconfig`"
- Section 15.1.4, "Configuring Distribution With `dsconfig`"
- Section 15.1.5, "Configuring DN Renaming With `dsconfig`"
- Section 15.1.6, "Configuring RDN Changing With `dsconfig`"
- Section 15.1.7, "Configuring Global Indexes By Using the Command Line"

### 15.1.1 Configuring Communication With Remote LDAP Servers

This section describes how to configure communication between a proxy instance and one or more remote LDAP servers. The section covers the following topics:

- Section 15.1.1.1, "Components of Communication with the Remote Server"
- Section 15.1.1.2, "Configuring LDAP Server Extensions"
- Section 15.1.1.3, "Configuring Proxy LDAP Workflow Elements"

#### 15.1.1.1 Components of Communication with the Remote Server

The following two elements are involved in communication between a proxy instance and a remote LDAP server:

- **LDAP Server Extension:** This element manages the connectivity with the remote server by periodically checking the response from the remote peer and providing valid connections maintained by the connection pool.
- **Proxy LDAP Workflow Element:** This element retrieves the connections from the LDAP server extension element and executes operations received from the user as defined in the configured mode.

#### 15.1.1.2 Configuring LDAP Server Extensions

This section describes how to configure the LDAP server extensions required to communicate with the remote LDAP server. The section covers the following topics:



##### 15.1.1.2.1 To Display the Existing LDAP Server Extensions

To display all the LDAP server extensions configured for a proxy instance, use the `dsconfig list-extensions` command, as follows:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  list-extensions

Extension  : Type
-----------:--------------------
gi-catalog : global-index-catalog
proxy1     : ldap-server
proxy2     : ldap-server
```

The extensions with type `ldap-server` are the LDAP server extensions. You should have one LDAP server extension for each remote LDAP server.

##### 15.1.1.2.2 To Display LDAP Server Extension Properties

To view the properties of a specific LDAP server extension, use the `dsconfig get-extension-prop` command, as follows:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  get-extension-prop --extension-name proxy1

Property                  : Value(s)
--------------------------:---------
enabled                   : true
remote-ldap-server-address : server1.example.com
remote-ldap-server-port    : 1389
```

The following properties are displayed:

`enabled`
indicates if the LDAP server extension is enabled (`true`) or not (`false`)

`remote-ldap-server-address` **and** `remote-ldap-server-port`
indicate the address and port of the remote LDAP server to which requests will be forwarded

`monitoring-bind-dn` **and** `monitoring-bind-password`
These properties are displayed only if the `--advanced` option is specified. They provide the credentials of the user that the extension will use to perform monitoring of the data source. If these properties have not been changed from the default, they are not displayed. Monitoring is then performed anonymously. To configure these properties, see Section 29.5, "Monitoring the Server With LDAP."

#### 15.1.1.2.3 To View Advanced LDAP Server Extension Properties

To view all the LDAP server extension properties, use the `dsconfig --advanced get-extension-prop` command. For example:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  --advanced get-extension-prop --extension-name proxy1
```

Properties similar to the following are displayed.

```
        Property                          Value(s)
        ----------------------------------------------------------------------
    1)  enabled                           true
    2)  java-class                        com.sun.dps.server.workflowelement
                                          .proxyldap.LDAPServerExtension
    3)  monitoring-check-interval         30000
    4)  monitoring-connect-timeout        5000
    5)  monitoring-inactivity-timeout     120000
    6)  monitoring-ping-timeout           5000
    7)  pool-increment                    5
    8)  pool-initial-size                 10
    9)  pool-max-size                     1000
    10) pool-max-write                    0
    11) pool-release-connection-interval  300000
    12) pool-use-max-write                false
    13) proxied-auth-use-v1               false
    14) remote-ldap-server-address        localhost
    15) remote-ldap-server-connect-timeout 10000
    16) remote-ldap-server-port           1389
    17) remote-ldap-server-read-only      false
    18) remote-ldap-server-read-timeout   10000
    19) remote-ldap-server-ssl-policy     never
    20) remote-ldap-server-ssl-port       636
    21) saturation-precision              5
    22) ssl-client-alias                  -
    23) ssl-key-manager-provider          -
    24) ssl-trust-all                     false
    25) ssl-trust-manager-provider        -
```

**Note:** Most of the advanced properties (except SSL properties) are set by default when the LDAP server extensions are created.

To modify these values, see Section 15.1.1.2.5, "To Modify the Properties of an LDAP Server Extension."

For information about the monitoring properties, see Section 15.1.1.2.7, "LDAP Data Source Monitoring Connection Properties." For information about the SSL (security) properties, see Chapter 21, "Configuring Security Between the Proxy and the Data Source."

#### 15.1.1.2.4 To Create an LDAP Server Extension

To create a new LDAP server extension, use the `dsconfig create-extension` command, as follows:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  create-extension \
  --extension-name DS-proxy5 \
  --type ldap-server \
  --set enabled:true \
  --set remote-ldap-server-address:DS5-hostname
  --set remote-ldap-server-port:1389
```

The extension `type` must be `ldap-server`. The name of the new extension is defined by `extension-name`, in this example `DS-proxy5`.

You must also specify the name of the remote LDAP server with which this extension is associated (`remote-ldap-server-address`). You can specify either the hostname or the IP address of the remote LDAP server.

If you do not specify a `remote-ldap-server-port`, the default LDAP port of 1389 is assumed.

#### 15.1.1.2.5 To Modify the Properties of an LDAP Server Extension

To modify the LDAP server extension properties, use the `set-extension-prop` subcommand. This subcommand enables you to do the following:

- set whether the LDAP server extension is enabled (`true`) or not (`false`)
- modify the remote LDAP directory server address and port (`remote-ldap-server-address` and `remote-ldap-server-port`)
- set the credentials of the user that the extension will use to perform monitoring of the data source (`monitoring-bind-dn` and `monitoring-bind-password`). If left blank, the monitoring will be performed anonymously, which is the default.

For example, a typical operation would be to change the remote LDAP server used. To do so, you need to set the new remote LDAP server address and port, as follows:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  set-extension-prop \
  --extension-name DS-proxy5 \
  --set remote-ldap-server-address:DS5-hostname \
  --set remote-ldap-server-port:3388
```

To modify advanced LDAP server extension properties, see Section 15.1.1.2.6, "To Modify the Advanced Properties of an LDAP Server Extension."

#### 15.1.1.2.6 To Modify the Advanced Properties of an LDAP Server Extension

You can configure the following advanced properties:

pool-increment
The increment by which the size of a connection pool is increased or decreased. If the remote-ldap-server-ssl-policy property is set to user, two pools of connections are created and the incremental change in size of each pool is set to pool-increment.

The default value is 5 connections.

pool-initial-size
The initial size of a connection pool. This is the initial number of connections to be created when a pool is initialized. Note that pool-initial-size is also minimum size of a pool.

The default value is 10 connections.

If the remote-ldap-server-ssl-policy property is set to user, two pools of connections are created and the initial size, and minimum size, of each pool is set to pool-initial-size. Therefore there can initially be twice the total number of connections indicated in pool-initial-size. For more information, see Section 21.2, "Modes of Secure Connection."

pool-max-size
The maximum size of a connection pool. This is the maximum number of connections that a pool can allocate. If the remote-ldap-server-ssl-policy property is set to user, two pools of connections are created and the maximum size of each pool is set to pool-max-size.

The default value is 1000 connections.

pool-max-write
The maximum number of write connections that a connection pool can allocate at the same time. This is an integer. This parameter is taken into account only if the pool-use-max-write parameter is set to true.

The default value is 0 connections.

pool-release-connection-interval
The time after which a connection is considered by the proxy to be unused if no traffic has been sent on it. This reduces the size of the pool of connections, if the pool has been previously increased. If the number of unused connections is greater than pool-increment, then the size of the pool is reduced by pool-increment. This means that unused connections are closed and are removed from the pool.

The default value is 300000 milliseconds (30 seconds).

pool-use-max-write
If this boolean is set to true, the pool-max-write parameter is taken into account, otherwise it is not. By default, pool-use-max-write is set to false.

proxied-auth-use-v1
When using the proxy authorization control mode, the default version of the control is v2. To use an older version for compatibility reasons, set proxied-auth-use-v1 to true. By default, proxied-auth-use-v1 is set to false. For more information about controls, see Appendix B, "Supported Controls and Operations."

remote-ldap-server-read-timeout
The timeout for reads. If the timeout is reached before the remote LDAP server sends back a response, an error is returned by the proxy to the client. By default, this value is 10000 milliseconds (10 seconds).

saturation-precision
The saturation precision is used in calculating the saturation threshold. Since the saturation limit can vary as requests are sent and received, the saturation precision indicates how much change the saturation should get before the saturation is taken into account. By default the saturation can vary by 5% before it is taken into account.

The monitoring properties are described in Section 15.1.1.2.7, "LDAP Data Source Monitoring Connection Properties."

The SSL properties are security features. For information about these properties, see Chapter 21, "Configuring Security Between the Proxy and the Data Source."

To modify the advanced LDAP server extension properties, use the `set-extension-prop --advanced` command.

**Note:** These advanced properties are set by default and typically are not modified.

An example of an advanced property that you may want to change is the `pool-max-size`. If you have a powerful remote LDAP server and you have configured the proxy so that it receives a maximum of requests, you can increase the `pool-max-size` as follows:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  set-extension-prop --advanced \
  --extension-name DS-proxy5 \
  --set pool-max-size:2000
```

#### 15.1.1.2.7 LDAP Data Source Monitoring Connection Properties

Using the `dsconfig --advanced` command for the LDAP server extension, you can view or change the following monitoring properties. All properties relate to proactive monitoring unless otherwise specified.

`monitoring-check-interval`
The monitoring check interval. This is the interval in milliseconds at which the proxy proactive monitoring checks the data source. The default value is 30000 milliseconds (30 seconds).

`monitoring-connect-timeout`
The maximum time in milliseconds after which the proactive monitoring facility will stop attempting to connect to the remote LDAP server. The default value is 5000 milliseconds (5 seconds). 0 means unlimited.

`monitoring-inactivity-timeout`
The time interval in milliseconds after which an idle connection is regularly checked to avoid connection closure by the remote server. The value of this parameter must be superior to the `monitoring-check-interval`. The default value is 120000 milliseconds (120 seconds).

`monitoring-ping-timeout`
The maximum time in milliseconds the proactive monitoring attempts to ping the remote server. The default value is 5000 milliseconds (5 seconds).

`remote-ldap-server-read-timeout`
The maximum time in milliseconds during which the LDAP Server Extension waits for a response from the remote server before the connection is regarded as having failed. 0 means unlimited.

remote-ldap-server-connect-timeout
The maximum time in milliseconds during which monitoring attempts to connect to the remote server before the connection is regarded as having failed. 0 means unlimited. The default is 10000 milliseconds (10 seconds).

### 15.1.1.3 Configuring Proxy LDAP Workflow Elements

This section describes how to configure the LDAP proxy workflow elements required to communicate with the remote LDAP server. The section covers the following topics:

- Section 15.1.1.3.1, "To Display the Existing Proxy LDAP Workflow Elements"
- Section 15.1.1.3.2, "To Display the Properties of a Proxy LDAP Workflow Element"
- Section 15.1.1.3.3, "To Create a Proxy LDAP Workflow Element"
- Section 15.1.1.3.4, "To Modify the Properties of a Proxy LDAP Workflow Element"

#### 15.1.1.3.1 To Display the Existing Proxy LDAP Workflow Elements

To display all the workflow elements configured on a particular proxy server instance, use the dsconfig list-workflow-elements command, as follows:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  list-workflow-elements

Workflow Element : Type               : enabled
-----------------:--------------------:--------
adminRoot        : ldif-local-backend : true
load-bal-we1     : load-balancing     : true
proxy-we1        : proxy-ldap         : true
proxy-we2        : proxy-ldap         : true
```

The proxy workflow elements are the ones with the type proxy-ldap.

#### 15.1.1.3.2 To Display the Properties of a Proxy LDAP Workflow Element

To view the proxy workflow element properties, use the dsconfig get-workflow-element-prop command, as follows:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  get-workflow-element-prop --element-name proxy-we1

Property                            : Value(s)
------------------------------------:--------------------
client-cred-mode                    : use-client-identity
enabled                             : true
ldap-server-extension               : proxy1
remote-ldap-server-bind-dn          : -
remote-ldap-server-bind-password    : -
use-proxy-auth                      : false
```

The following properties are displayed:

client-cred-mode
indicates how the proxy connects to the remote LDAP server. In this example, the status is use-client-identity, which means that the proxy will connect to the remote LDAP server with the same credentials that the client used to connect to the proxy. This is the default mode.

For more information, see Chapter 21, "Configuring Security Between the Proxy and the Data Source."

enabled
indicates if the workflow is enabled (true) or not (false)

ldap-server-extension
the name of the LDAP server extension with which the workflow element is associated

**remote-ldap-server-bind-dn and remote-ldap-server-bind-password**
the credentials of the user that the proxy uses to connect to the remote LDAP server when client-cred-mode is use-specific-identity or use-proxy-auth.

#### 15.1.1.3.3 To Create a Proxy LDAP Workflow Element

You must have configured an LDAP server extension before you create a proxy LDAP workflow element.

To create a proxy LDAP workflow element, use the dsconfig create-workflow-element command, as follows:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  create-workflow-element \
  --element-name proxy-we5 \
  --type proxy-ldap \
  --set enabled:true \
  --set client-cred-mode:use-client-identity \
  --set ldap-server-extension:DS-proxy5
```

The workflow element type must be proxy-ldap. The name of the new proxy LDAP workflow element is defined by element-name, in this example proxy-we5.

The client credential mode (client-cred-mode) indicates how the proxy will connect to the remote LDAP server. In this example, the credential mode is use-client-identity, which means that the proxy will connect to the remote LDAP server with the same credentials as those used by the client to connect to the proxy. This is the default mode.

**Notes:**

- If you use Oracle Unified Directory remote LDAP servers and the client credential mode is set to `use-proxy-auth`, the user as which you are connecting *must* exist on the remote LDAP server. If the user does not exist, requests will be rejected. If you cannot guarantee that the user exists on the remote LDAP server, rather set the client credential mode to `use-specific-identity`.
- If the user deployment performs an internal operations then you must define the root credentials. For example, if you are using RDN changing as described in Section 15.1.6, "Configuring RDN Changing With `dsconfig`" then the root credentials are defined by the following properties:

  `remote-root-dn`

  `remote-root-password`

  These are the credentials of the root user of the remote LDAP server when the server performs internal operations.
- When managing passwords in a proxy LDAP workflow element (`remote-ldap-server-bind-password` or `remote-root-passord`), the following syntax are valid:

  `<password-value>` or `file://<password-file>`

For more information, see Chapter 21, "Configuring Security Between the Proxy and the Data Source."

#### 15.1.1.3.4 To Modify the Properties of a Proxy LDAP Workflow Element

To modify the proxy LDAP workflow element properties, use the `set-workflow-element-prop` command.

You can modify the following properties:

- Set whether the proxy LDAP workflow element is enabled (`true`) or not (`false`)
- Set the client credential mode that is used (`client-cred-mode`)
- Associate an LDAP server extension, to indicate which remote LDAP server to use (`ldap-server-extension`)
- Set the credentials of the user that the proxy uses to connect to the remote LDAP server (`remote-ldap-server-bind-dn` and `remote-ldap-server-bind-password`). The following syntaxes are supported:
  - `<password-value>`
  - `file://<password-file>`

  Passing a password in clear on the command line is supported but not recommended. It is recommended to use a password-file. You can delete the password-file once the command is executed.

For example, if you want to modify the LDAP server extension used by the workflow element in order to use a different remote LDAP server, do the following:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
```

```
set-workflow-element-prop --advanced \
--element-name proxy-we5 \
--set remote-ldap-server-bind-dn:uid=Specific\ User,dc=example,dc=com \
--remote-ldap-server-bind-password:file://pwd-file \
--set ldap-server-extension:DS-proxy3 \
--set client-cred-mode:use-specific-identity
```

## 15.1.2 Configuring the Bind Mode

When an end user executes an authenticated operation, the proxy LDAP workflow element receives the following two distinct operations:

1. A BIND operation that authenticates the user against the remote server.
2. An operation to execute.

When a bind operation is executed, the proxy LDAP workflow element retrieves a connection from the LDAP server extension, performs the BIND operation, then releases the connection.

When the actual operation arrives, the proxy LDAP workflow element again retrieves a connection from the LDAP server extension. If a connection is found that is still bound with the appropriate credentials, that connection is reused. If not, a new connection must be authenticated. This additional authentication operation is called a *silent bind*.

The set of credentials used to perform a silent bind is determined by the *bind mode*, which is a property of the LDAP workflow element. These credentials can be the client credentials or the proxy credentials. Table 15–1 lists the bind modes that are supported by Oracle Unified Directory.

***Table 15–1 Supported Bind Modes by Oracle Unified Directory***

| Mode | Description |
|---|---|
| `use-client-identity` | Use the client credentials to perform the silent bind. |
| `use-specific-identity` | Use the proxy credentials to perform the silent bind. |

### 15.1.2.1 Configuring the Bind Mode Parameters to Optimize the Server

For each of the bind modes described in Table 15–1, you can configure additional parameters to tweak the behavior of the server. These parameters are described in the following sections:

- Section 15.1.2.1.1, "Configuring the `use-client-identity` Bind Mode"
- Section 15.1.2.1.2, "Configuring the `use-specific-identity` Bind Mode"

#### 15.1.2.1.1 Configuring the `use-client-identity` Bind Mode

When the bind mode is set to `use-client-identity`, the server uses the client credentials to perform the silent bind, unless specific parameters prevent it from doing so. The parameters that prevent the server from using the client credentials are the following:

- Using Include and Exclude Lists
- Using the `never-bind` Parameter

**Using Include and Exclude Lists**

You can configure the following lists:

- **Include List:** Lists the suffixes that are handled by the remote server.
- **Exclude List:** Lists the suffixes that are not handled by the remote server.

If the client bind DN is a descendant of one DN on the include list, and the client bind DN is not a descendant of any DN on the exclude list, the proxy server uses the client credentials to perform a silent bind. Otherwise the proxy server uses the proxy credentials to perform the silent bind. If both lists are empty, the proxy server always uses the client credentials.

The include and exclude lists are not mutually exclusive and can be used simultaneously. However, it is recommended that you define only one list. In addition, you cannot define the same suffixes in both the lists.

#### Using the `never-bind` Parameter

The `never-bind` parameter is applicable whenever the proxy needs to perform a bind with the client credentials. If this flag is set to `true`, the proxy server reads the user entry from the remote data source, and validates the user password itself, instead of forwarding the bind to the remote server. Note that the credentials used to read the user entry are proxy credentials, defined in the following properties of the proxy LDAP workflow element: `remote-ldap-server-bind-dn` and `remote-ldap-server-bind-password`.

If the incoming bind operation contains controls that are critical, an error result is returned as controls dedicated to bind operations are incompatible with the `never-bind` feature.

> **Note:** If the proxy uses its own credentials to read the user entry, the proxy authorization control can be added to operations, to indicate the identity of the client at the origin of the request. The value of the `use-proxy-auth` property determines whether the control should be added.

#### 15.1.2.1.2 Configuring the `use-specific-identity` Bind Mode

When the bind mode is set to `use-specific-identity`, the proxy server uses the proxy credentials to perform all silent binds. The proxy credentials are defined in the following properties of the proxy LDAP workflow element: `remote-ldap-server-bind-dn` and `remote-ldap-server-bind-password`.

In `use-specific-identity` bind mode, you can set the following parameters:

- Using the `use-proxy-auth` Parameter
- Using the `never-bind` Parameter

#### Using the `use-proxy-auth` Parameter

If the `use-proxy-auth` flag is set to `true`, the proxy server adds a proxy authorization control to all requests, except bind requests. The value of the proxy authorization identifier is the client bind DN.

#### Using the `never-bind` Parameter

The `never-bind` parameter is applicable whenever the proxy needs to perform a bind with the client credentials. When this flag is set to `true`, the proxy server reads the user entry from the remote data source, and validates the user password itself, instead of forwarding the bind to the remote server. Note that the credentials used to read the user entry are proxy credentials, defined in the following properties of the

proxy LDAP workflow element: `remote-ldap-server-bind-dn` and `remote-ldap-server-bind-password`.

## 15.1.3 Configuring Load Balancing With `dsconfig`

To forward client requests to remote LDAP servers using load balancing, you need the following elements:

- a load balancing workflow element
- a load balancing algorithm
- a load balancing route, for each remote LDAP server

A load balancing workflow element can only have one load balancing algorithm. However, the same load balancing algorithm is used by all the load balancing routes in the deployment.

This section covers all the administration tasks related to load balancing. For information about setting up a load balancing deployment during installation, see "To Configure Simple Load Balancing" section in *Oracle Fusion Middleware Installation Guide for Oracle Unified Directory*. It contains the following topics:

- Section 15.1.3.1, "To Configure Load Balancing"
- Section 15.1.3.2, "Creating a Load Balancing Workflow Element"
- Section 15.1.3.3, "Creating a Load Balancing Algorithm"
- Section 15.1.3.4, "Creating Load Balancing Routes"
- Section 15.1.3.5, "Modifying Load Balancing Properties"

The following examples describe how to configure load balancing using the `dsconfig` command. All of the examples specify the proxy hostname (`-h`), the proxy admin port (`-p`), the bind DN (`-D`), and the bind password file (`-j`), and use the `-X` option to trust all certificates.

### 15.1.3.1 To Configure Load Balancing

1. Create a load balancing workflow element.

   See Section 15.1.3.2, "Creating a Load Balancing Workflow Element."

2. Create a load balancing algorithm.

   See Section 15.1.3.3, "Creating a Load Balancing Algorithm.".

3. Create one load balancing route for each load balancing workflow element.

   See Section 15.1.3.4, "Creating Load Balancing Routes."

### 15.1.3.2 Creating a Load Balancing Workflow Element

To configure load balancing, you must create a load balancing workflow element using the `dsconfig create-workflow-element` command. For example:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  create-workflow-element \
  --element-name load-bal-we1 \
  --type load-balancing \
  --set enabled:true
```

To create a load balancing workflow element, the type must be load-balancing. The name of the workflow element is defined by element-name, in this example load-bal-we1.

### 15.1.3.3 Creating a Load Balancing Algorithm

In order to determine how the requests will be forwarded in a load balancing deployment, you must configure the load balancing algorithm. The load balancing algorithm set determines how client requests will be dispatched across the pool of remote LDAP servers. The possible load balancing types are: failover, optimal, proportional, or saturation.

To create the load balancing algorithm, you must have a load balancing workflow element. See Section 15.1.3.2, "Creating a Load Balancing Workflow Element."

Create a load balancing algorithm using the dsconfig create-load-balancing-algorithm command. For example:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  create-load-balancing-algorithm \
  --element-name load-bal-we1 \
  --type failover
```

To create a load balancing algorithm, the you must indicate the type as proportional, optimal, failover, or saturation. The name of the workflow element is defined by element-name, in this example load-bal-we1.

### 15.1.3.4 Creating Load Balancing Routes

You should have one load balancing route per data source. Before you create a load balancing route, the load balancing workflow element and load balancing algorithm must already be created.

To create a load balancing route, use the dsconfig create-load-balancing-route command. For example:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  create-load-balancing-route \
  --element-name load-bal-we1 \
  --route-name load-bal-route1 \
  --type failover \
  --set workflow-element:proxy-we1 \
  --set add-priority:1 \
  --set bind-priority:2 \
  --set compare-priority:2 \
  --set delete-priority:1 \
  --set extended-priority:2 \
  --set modify-priority:1 \
  --set modifydn-priority:1 \
  --set search-priority:2
```

In this example, load-bal-route1 is the name of the new load balancing route, load-bal-we1 is the name of the existing load balancing workflow element, and proxy-we1 is the name of the LDAP proxy workflow element. The type must be the same as the one defined by the load balancing algorithm associated, in this case failover.

The properties set (in this case priority) are related to the type of load balancing created. For more information about the properties of the routes, linked to the algorithm type see Section 15.1.3.5, "Modifying Load Balancing Properties."

### 15.1.3.5 Modifying Load Balancing Properties

After a load balancing deployment has been set up, you can modify certain properties, such as the priority, weight, and saturation threshold. Most of these properties are changed at the load balancing route level.

You can modify the following load balancing properties, depending on the load balancing algorithm:

| Failover | Optimal | Proportional | Saturation | Search Filter |
|---|---|---|---|---|
| add-priority | alert-threshold | add-weight | alert-threshold | priority |
| bind-priority | saturation-precision* | bind-weight | priority | allowed-attributes |
| compare-priority | workflow-element | compare-weight | threshold | prohibited-attributes |
| delete-priority | | delete-weight | saturation-precision* | workflow-element |
| extended-priority | | extended-weight | workflow-element | |
| modify-priority | | modify-weight | | |
| modifydn-priority | | modifydn-weight | | |
| search-priority | | search-weight | | |
| workflow-element | | workflow-element | | |
| switch-back flag | | | | |

* saturation precision is a property of the LDAP server extension.

To modify load balancing route properties, use the `dsconfig set-load-balancing-route-prop` command.

New routes can be added on a running algorithm, or routes can be deleted or have their priorities modified without the need to restart the server.

---

**Note:** You cannot modify the load balancing algorithm type.

To change a failover load balancing deployment to a proportional one, for example, you must create a new load balancing deployment. See Section 15.1.3, "Configuring Load Balancing With dsconfig."

---

The following sections describe the different settings possible in a load-balancing deployment:

- Section 15.1.3.5.1, "Setting the Priority in a Failover Algorithm"
- Section 15.1.3.5.2, "Setting the `switch-back` Flag"
- Section 15.1.3.5.3, "Setting the Saturation Precision for the Optimal or Saturation Algorithm"
- Section 15.1.3.5.4, "Setting the Weight of a Proportional Algorithm"
- Section 15.1.3.5.5, "Setting the Threshold for a Saturation Algorithm"

- Section 15.1.3.5.6, "Setting the Saturation Threshold Alert"
- Section 15.1.3.5.7, "Setting Client Connection Affinity"
- Section 15.1.3.5.8, "Deleting Load Balancing Elements"

#### 15.1.3.5.1 Setting the Priority in a Failover Algorithm

In a load balancing deployment that uses the failover algorithm, you can modify the proxy workflow element to change the route that is used, as well as the priority of the route for a given operation type.

In a failover algorithm, a priority of 1 is the highest priority and indicates the main route that will be used for a specific operation type. A route with priority 2 (or more) is the secondary route used in case of failure on the primary route. The priority is set for each operation type. This means that a route with a priority of 1 for Add operations, can have a priority of 2 for Bind and Search operations.

For example, if the route `load-bal-route1` was initially set as the main route with a priority of `1` for Add operations, but you now want to make it the backup route, you can set the priority to `2` using the following command.

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  set-load-balancing-route-prop \
  --element-name load-bal-we1 \
  --route-name load-bal-route1
  --set add-priority: 2
```

> **Note:** If two routes have the same priority for a given operation type, the choice of the active route which treats the request is random.

#### 15.1.3.5.2 Setting the `switch-back` Flag

After failover in a load balancing deployment, the backup route continues to handle all incoming requests, even after the priority server that had failed becomes available. Switch-back or failback to the primary route does not automatically occur unless the `switch-back` flag has been set to `true`. By default, the `switch-back` flag is set to `false`.

The `switch-back` flag is an advanced property. To set the `switch-back` flag to `true`, do the following:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  --advanced set-load-balancing-algorithm-prop \
  --element-name load-bal-we1 \
  --set switch-back:true
```

#### 15.1.3.5.3 Setting the Saturation Precision for the Optimal or Saturation Algorithm

In a load balancing deployment that uses the optimal or the saturation algorithm, you can set the saturation precision level. The saturation precision is the delta between two saturation levels, and is used to determine the route with the lowest saturation level. By default, the saturation precision level is set to 5.

If you find that the saturation precision level is too low, and that the routes are changing too frequently, you can modify the saturation precision level as follows:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  --advanced set-extension-prop \
  --extension-name proxy1 \
```

```
  --set saturation-precision:10
```

#### 15.1.3.5.4 Setting the Weight of a Proportional Algorithm

Once you have created a load balancing deployment using the proportional algorithm, you can modify the proxy workflow element to change the route used, as well as the weight of a route. The weight can be different for each operation type. The value of the weight should be 0 or more, were 0 indicates that the route will not be used for the specified operation.

Using the interactive mode of dsconfig, you can see that the following properties can be modified:

```
>>>> Configure the properties of the Proportional Load Balancing Route

        Property          Value(s)
        --------------------------
    1)  add-weight        1
    2)  bind-weight       1
    3)  compare-weight    1
    4)  delete-weight     1
    5)  extended-weight   1
    6)  modify-weight     1
    7)  modifydn-weight   1
    8)  search-weight     1
    9)  workflow-element  proxy-we1
```

For example, if you initially set all your routes to a weight of 1 on all operations, then all the servers will handle an equal ratio of operations. However, if you want a remote LDAP server to handle more search requests than the other servers in the deployment, then you can set its search-weight to a higher value, such as 5. To do so, use the following command:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  set-load-balancing-route-prop \
  --element-name load-bal-we1 \
  --route-name load-bal-route1 \
  --set search-weight:5
```

**Note:** To modify the weight for all operations, you must modify the weight for each operation individually.

To modify load-bal-route1 to handle twice as many operations as the other route, you would set the weight of all operations to 2 (assuming the weight on the other route is set to 1). In other words, run the command as follows:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  set-load-balancing-route-prop \
  --element-name load-bal-we1 \
  --route-name load-bal-route1 \
  --set add-weight:2 \
  --set bind-weight:2 \
  --set compare-weight:2 \
  --set delete-weight:2 \
  --set extended-weight:2 \
  --set modify-weight:2 \
  --set modifydn-weight:2 \
  --set search-weight:2
```

If the weight is set to 0 for any operations, the route will not perform the specified operation. For example, if add-weight is set to 0, then load-bal-route1 will not forward any add requests to the associated remote LDAP server. If all configured routes indicate a weight of 0 for a specific operation, that operation will not be supported.

#### 15.1.3.5.5 Setting the Threshold for a Saturation Algorithm

Once you have created a load balancing deployment using the saturation algorithm, you can modify the proxy workflow element used, the priority of the route, the saturation threshold, and the saturation threshold alert.

With a saturation algorithm, requests are distributed based on two criteria: the priority of the server and the saturation threshold of the server. The saturation threshold is the limit at which the server is considered "maximized" and service may become degraded. In a load balancing deployment with saturation algorithm, requests are sent to the server with the highest priority (1) until the server reaches the saturation threshold indicated.

For example, if you indicate load-bal-route1 as the server with the highest priority, with a threshold of 80%, all requests will be sent to load-bal-route1 until its saturation threshold goes over 80%. Once it exceeds 80%, then requests are routed to the next server in the priority list.

```
>>>> Configure the properties of the Saturation Load Balancing Route

        Property          Value(s)
        --------------------------
    1)  alert-threshold   85
    2)  priority          1
    3)  threshold         80
    4)  workflow-element  proxy-we1
```

To modify the saturation threshold, use the following command:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  set-load-balancing-route-prop \
  --element-name load-bal-we1 \
  --route-name load-bal-route1 \
  --set threshold:90
```

In this example, the saturation threshold has been set to 90%.

#### 15.1.3.5.6 Setting the Saturation Threshold Alert

The saturation threshold alert is used to set at which point a notification will be sent to the system administrator to indicate that the server has passed the saturation limit. Generally, the saturation threshold alert is set higher than the saturation limit, in order to indicate if the saturation continues to increase past the saturation threshold (which may indicate a problem). The alert should bet set with an acceptable buffer, as there may be a short delay in which saturation continues to increase slightly before requests are forwarded to another route.

To modify the saturation threshold, use the following command:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  set-load-balancing-route-prop \
  --element-name load-bal-we1 \
  --route-name load-bal-route1 \
  --set alert-threshold:85
```

You can set the saturation threshold alert to a value lower than the saturation threshold, in order to perform preventative actions (For example, If the main route is a set of load balanced servers, then you can add one or more servers to that set of servers as a preventive action). This may imply receiving notifications even in cases where the saturation threshold is not reached. That is, a saturation threshold alert is sent, but the saturation limit drops and does not reach the saturation threshold. However, the requests will only be sent to the next priority route when the saturation threshold is reached.

For more information on setting the notification message, see Section 29.4, "Configuring Alerts and Account Status Notification Handlers."

#### 15.1.3.5.7 Setting Client Connection Affinity

When client connection affinity is defined, requests from a specified client connection are routed to the same server, bypassing the load balancing algorithm that has been set. Client connection affinity is set at the network group level.

To set client connection affinity, use the `dsconfig create-network-group-qos-policy` command. For more information, see Section 14.1.6.3, "Creating a Network Group Quality of Service Policy."

***Example 15–1 Example of Client Connection Affinity Rejected***

When client connection affinity is set, the load balancing algorithm is bypassed as long as the constraints of the weights that have been defined are respected.

For example, assume that the following routes are set with the following weights:

- `LB-route1: add=10, search= 0`
- `LB-route2: add=0, search=10`

It is clear that `LB-route1` receives all the add requests, and `LB-route2` receives all the search requests.

Assume that the load balancing deployment in this example is set with a client connection affinity of `all-requests-after-first-write-request`. If the load balancing deployment receives the following string of requests: Add, Search, Add, typically, the client connection affinity would send the Search request to the same route (`LB-route1`) as the first Add request. However, in this case, since Search requests are not allowed on `LB-route1`, the load balancing algorithm is *not* bypassed by the client affinity.

#### 15.1.3.5.8 Deleting Load Balancing Elements

To delete a complete load balancing workflow (workflow element, algorithm, and routes), you need only delete the load balancing workflow element. When you delete a load balancing workflow element, the associated load balancing algorithm and routes are silently deleted.

## 15.1.4 Configuring Distribution With `dsconfig`

To forward client requests to remote LDAP servers using distribution, the following components must be configured on the proxy server:

- a distribution workflow element
- a distribution algorithm
- one or more distribution partitions (typically one per remote LDAP server)

A distribution workflow element can only have one distribution algorithm, that defines how data is distributed. A distribution algorithm can use several partitions.

The following examples describe how to configure distribution using the dsconfig command. For information about setting up a distribution deployment during setup, see "To Configure Simple Distribution" section in *Oracle Fusion Middleware Installation Guide for Oracle Unified Directory*.

All the commands in the following procedures specify the proxy hostname (-h), the proxy admin port (-p), the bind DN (-D), and the bind password file (-j). The examples use the -X option to trust all certificates.

This section contains the following topics:



### 15.1.4.1 To Configure Distribution

1. Create a distribution workflow element.

   See Section 15.1.4.2, "Creating a Distribution Workflow Element."

2. Create a distribution algorithm.

   See Section 15.1.4.3, "Creating a Distribution Algorithm."

3. Create one partition for each chunk of partitioned data. A partition must be associated with one remote LDAP server, or with a set of replicated remote LDAP servers.

   - For a capacity-based distribution see Section 15.1.4.4.1, "Creating a capacity Distribution Partition."
   - For a lexico or numeric distribution see Section 15.1.4.4.2, "Creating a lexico or numeric Distribution Partition."
   - If you are using DN pattern algorithm, see Section 15.1.4.4.3, "Creating a dnpattern Distribution Partition."

### 15.1.4.2 Creating a Distribution Workflow Element

To configure distribution, you must create a distribution workflow element using the dsconfig create-workflow-element command. For example:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  create-workflow-element \
  --element-name distrib-we \
  --type distribution \
  --set enabled:true \
  --set base-dn:ou=people,dc=example,dc=com
```

To create a distribution workflow element, the type must be distribution. The name of the workflow element is defined by element-name, in this example distrib-we.

> **Note:** When declaring the base-dn using the create-workflow-element subcommand as shown above, ensure that you specify the full tree structure.

To complete the distribution element of your configuration, create the distribution algorithm and the appropriate partitions.

### 15.1.4.3 Creating a Distribution Algorithm

To determine how the requests will be forwarded in a distribution deployment, you must configure the distribution algorithm. The algorithm set determines how the data is partitioned and to which partition a request is sent. The possible distribution types are: numeric, lexico, or dnpattern.

To create the distribution algorithm, you must have a distribution workflow element. See Section 15.1.4.2, "Creating a Distribution Workflow Element."

Create a distribution algorithm using the dsconfig create-distribution-algorithm command. For example:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  create-distribution-algorithm \
  --element-name distrib-we \
  --type numeric \
  --set distribution-attribute:uid
```

The name of the workflow element is defined by element-name, in this example distrib-we. The distribution algorithm type must be set as capacity, numeric, lexico, or dnpattern. The properties set depend on the algorithm type. In this example, distribution-attribute must be set, as the algorithm type is numeric.

### 15.1.4.4 Creating Distribution Partitions

You can create the following types of distribution partitions:

- Section 15.1.4.4.1, "Creating a capacity Distribution Partition"
- Section 15.1.4.4.2, "Creating a lexico or numeric Distribution Partition"
- Section 15.1.4.4.3, "Creating a dnpattern Distribution Partition"

#### 15.1.4.4.1 Creating a capacity Distribution Partition

To create a capacity distribution partition, the distribution workflow element and distribution algorithm must already be created. You must create one distribution partition per data set.

To create a distribution partition, use the dsconfig create-distribution-partition command. For example:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  create-distribution-partition \
  --element-name distrib-we \
  --partition-name distrib-partition1 \
  --type capacity \
  --set partition-id:1 \
```

```
  --set workflow-element: proxy-we1 \
  --set max-entries:1000
```

> **Note:** You must create a global index catalog and have the DNs indexed to use the capacity-based algorithm. To create global index catalogs, see Section 15.1.7.1.1, "To Create a Global Index Catalog Containing Global Indexes."

A distribution partition is identified by both a partition name, in this example, `distrib-partition1` and a partition id. The partition id must be an simple integer, as it will be used for the global index catalog reference. The type must be the same as the one defined by the distribution algorithm associated, in this case `capacity`.

To create a distribution partition, you must also indicate the name of the existing distribution workflow element (`element-name`) that manages the partition (here `distrib-we`), and the name of the next element in the work flow (`workflow-element`), such as an LDAP workflow element (in this example `proxy-we1`). The proxy workflow element indicates the path used to reach the data on the remote LDAP server. For more information on the proxy, see Section 15.1.1, "Configuring Communication With Remote LDAP Servers."

When creating a `capacity` distribution partition, you must indicate the maximum number of entries the partition can hold, for example 1000.

#### 15.1.4.4.2 Creating a `lexico` or `numeric` Distribution Partition

Lexico and numeric distribution are very similar, so you set the same properties when you create a distribution partition for lexico or numeric distribution. You must create one distribution partition per data set.

To create `lexico` or `numeric` distribution partitions, the distribution workflow element and distribution algorithm must already be created.

To create a distribution partition, use the `dsconfig create-distribution-partition` command. For example for a numeric distribution, you might create a partition as follows:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  create-distribution-partition \
  --element-name distrib-we \
  --partition-name distrib-partition1 \
  --type numeric \
  --set partition-id:1 \
  --set workflow-element: proxy-we1 \
  --set lower-bound:1000 \
  --set upper-bound:2000
```

A distribution partition is identified by both a partition name, in this example, `distrib-partition1` and a partition id. The partition id must be an simple integer, as it will be used for the global index catalog reference. The type must be the same as the one defined by the distribution algorithm associated, in this case `numeric`.

In order to create a distribution partition, you must also indicate the name of the existing distribution workflow (here `distrib-we`), and the name of the associated workflow element, such as an LDAP workflow element (in this example `proxy-we1`). The proxy workflow element indicates the path used to reach the data on the remote LDAP server. For more information on the proxy, see Section 15.1.1, "Configuring Communication With Remote LDAP Servers."

When creating a lexico or numeric distribution partition, you must indicate the lower and upper boundaries of the partition. The proxy checks to ensure that there is no overlap in the boundaries of any two partitions. This means that you cannot set partition 1 with boundaries 1000-3000 and partition 2 with boundaries 2000-4000.

The upper boundary is exclusive, which means that in the example above, the partitioned data only includes values between 1000 up to 1999. If you want the upper boundary or lower boundary to be unlimited, use the keyword unlimited.

The properties set (in this example boundaries) are related to the type of distribution created. For more information about the properties of the partitions, linked to the algorithm type see Section 15.1.4, "Configuring Distribution With dsconfig."

Note that for a lexico distribution algorithm, the sort sequence that is used is ASCII.

#### 15.1.4.4.3 Creating a dnpattern Distribution Partition

Before you create a dnpattern distribution partition, the distribution workflow element and distribution algorithm must already be created.

To create a dnpattern distribution partition, use the dsconfig create-distribution-partition command. For example:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  create-distribution-partition \
  --element-name distrib-we \
  --partition-name distrib-partition5 \
  --type dnpattern \
  --set partition-id:5 \
  --set workflow-element: proxy-we1 \
  --set dn-pattern:uid=[0-9]*[01].*
```

A distribution partition is identified by both a partition name, in this example, distrib-partition5 and a partition ID. The partition ID is used for the global index catalog reference, and be an simple integer. To create a distribution partition, you must also indicate the name of the existing distribution workflow (here distrib-we), and the name of the associated workflow element, such as an LDAP proxy (in this example proxy-we1). The type must be the same as the one defined by the distribution algorithm associated, in this case dnpattern.

In a distribution scenario that uses a dnpattern algorithm, requests are sent to a partition when the request RDNs below the distribution base DN match the DN string pattern. For example, if the distribution base DN is ou=people,dc=example,dc=com and the request base DN is uid=1,ou=people,dc=example,dc=com, the check against the string pattern is done on the RDN uid=1.

Similarly, if the distribution base DN is ou=people,dc=example,dc=com and the request base DN is uid=1,ou=region1,ou=people,dc=example,dc=com, the check against the string pattern is done on the RDNs uid=1,ou=region1.

#### 15.1.4.4.4 DN Pattern String Syntax

The DN string pattern must comply with the DN syntax and with a subset of the Java Pattern class.

| DN Pattern String | Description |
|---|---|
| . | any character |
| \\ | backslash |

| DN Pattern String | Description |
|---|---|
| \t | TAB character |
| [abc] | a, b, or c |
| [^abc] | any character except a, b, or c |
| [a-zA-Z] | a through z, or A through Z, inclusive (range) |
| [a-d[m-p]] | a through d, or m through p (union) |
| [a-z&&[def]] | d, e, or f (intersection) |
| [a-z&&[^bc]] | a through z, except for b and c (subtraction) |
| [A-Z&&[^M-P]] | a through z, and not m through p (subtraction) |

The following quantifiers can be used:

| | |
|---|---|
| X? | X, once or not at all |
| X* | X, zero or more times |
| X+ | X, one or more times |
| X{n} | X, exactly n times |
| X{n,} | X, at least n times |
| X{n,m} | X, at least n times but no more than m times |

#### 15.1.4.4.5 Using DN Pattern negative-match

The distribution property called negative-match allows you to specify the opposite of the DN pattern that should be matched. That is, you specify a DN pattern to be ignored; any value that *does not* match the specified DN pattern will be distributed. By default, the negative-match property is set to false.

Create a dnpattern distribution partition using negative-match as follows:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  create-distribution-partition \
  --element-name distrib-we \
  --partition-name distrib-partition5 \
  --type dnpattern \
  --set partition-id:5 \
  --set workflow-element: proxy-we1 \
  --set dn-pattern:uid=[123]*[0-9].* \
  --set negative-match:true
```

In the example above, since negative-match has been set to true, any requests for uid that does not start with 1, 2, or 3, with n characters following will be forwarded to the partition.

### 15.1.4.5 Managing Modify DN Requests

You can modify a DN so that the new entry remains in the same partition as the original entry. By default, the proxy does not allow you to modify the DN to a value that is outside the range of the current partition.

If you want to allow modifyDN requests to change the DN to a value that is outside the boundaries of the partition in which the entry is located, set the force-modify-dn flag to true.

Assume, for example, that you have two partitions: Partition 1 with uid boundaries from 0-999 and Partition 2 with uid boundaries from 1000-1999. If the force-modify-dn flag is set to true and you modify the uid of an entry from 1 to 1001, the change will be allowed, but the entry with uid 1001 will remain in Partition 1. It is not moved to Partition 2.

If you then search for uid=1001, the server will return an error, indicating that no such entry is found. To locate the entry, you must use a global index catalog. This ensures that modified entries are always found. To configure a global index catalog, see Section 15.1.7, "Configuring Global Indexes By Using the Command Line."

To force a modify DN operation, set the force-modify-dn flag to true, as follows:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  --advanced set-workflow-element-prop --element-name distrib-we \
  --set force-modify-dn:true
```

#### 15.1.4.6 Configuring Criticality in Workflows

The criticality configuration determines the server behavior when a search operation fails. Criticality applies only to search requests. All other requests are processed normally by the server.

You can configure criticality by setting the criticality flag at the workflow level. When a search request is executed on a workflow, then it is executed on several workflows if there are subordinate workflows. The criticality setting of a workflow can be one of the following:

- true

  This is the default setting and indicates that the workflow is considered as critical. If a workflow fails to return a result the processing is stopped regardless of whether the execution of the operation was successful on any other workflow.

- false

  This setting indicates that the workflow is non-critical. A criticality setting of false tells the server that the failure to perform an operation in the workflow is not critical to the overall result. If the non-critical workflow fails to return a result the server simply omits the results (as if the workflow did not return any data), returns a SUCCESS result code to the client, and does not indicate any error.

- Partial

  This setting indicates that the workflow is partially critical. This implies that the application can notify its own users that partial results were obtained. If a partially-critical partition fails to return a result because, for example, it is fully saturated or disabled, the server returns an Admin Limit Exceeded error. While this is not the expected error, the intention of this setting is to cause client application logic to indicate that not all results are shown.

To set the criticality of a workflow, use the dsconfig set-workflow-prop command. For example, the following command sets the criticality of a workflow named workflow-1 to true:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  set-workflow-prop --workflow-name workflow-1 \
  --set criticality:true
```

### 15.1.4.7 Configuring Criticality in Workflow Elements

In a distribution deployment, the *criticality* configuration determines the server behavior when a search operation fails, due to a host error. Criticality applies only to search requests. All other requests are processed normally by the server.

Criticality is configured for each distribution partition in a distribution workflow element. The criticality setting of a distribution partition can be one of the following:

- `true`

  This is the default setting and indicates that the partition is considered as critical. If a partition fails to return a result because, for example, it is fully saturated or disabled, the server returns an UNAVAILABLE error to the client regardless of whether data was found in any other partition.

- `false`

  This setting indicates that the partition is non-critical. A criticality setting of `false` tells the server that the failure to perform an operation in the partition is not critical to the overall result. If the non-critical partition fails to return a result because, for example, it is fully saturated or disabled, the server simply omits the results (as if the partition did not return any data), returns a SUCCESS result code to the client, and does not indicate any error.

- `Partial`

  This setting indicates that the partition is partially critical. This implies that the application can notify its own users that partial results were obtained. If a partially-critical partition fails to return a result because, for example, it is fully saturated or disabled, the server returns an `Admin Limit Exceeded` error. While this is not the expected error, the intention of this setting is to cause client application logic to indicate that not all results are shown.

For all types of workflow element, other than a distribution workflow element, criticality is implicit and is handled as follows:

- Load Balancing: All routes are considered as non critical, because if a route is not functional then it is not taken into consideration by the load balancer while determining the selected route.
- LDAP Proxy Workflow Element: An LDAP server is always considered as critical.
- Local Backend Workflow Element: A local backend server is always considered as critical.

To set the criticality of a distribution partition, use the `dsconfig set-distribution-partition-prop` command. For example, the following command sets the criticality of a partition named `distrib-partition-1` to `true`:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  set-distribution-partition-prop --element-name distrib-we \
  --partition-name distrib-partition-1 --set criticality:true
```

### 15.1.4.8 Deleting a Distribution Configuration

To delete a complete distribution workflow (workflow element, algorithm, and partitions), you need only delete the distribution workflow element. When you delete a distribution workflow element, the associated distribution algorithm and partitions are silently deleted.

## 15.1.5 Configuring DN Renaming With dsconfig

To configure DN renaming, create a DN renaming workflow element, using the dsconfig create-workflow-element command.

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
 create-workflow-element \
 --type dn-renaming \
 --element-name RenameorgDN \
 --set client-base-dn:ou=myorg,dc=example,dc=com \
 --set next-workflow-element:load-bal-we1 \
 --set source-base-dn:ou=people,dc=example,dc=com \
 --set enabled:true
```

- --set client-base-dn indicates the client base DN, which is the workflow entry point
- --set source-base-dn indicates the base DN which the entries should have after transformation, which is the workflow exit point.
- --set next-workflow-element indicates the workflow element that will follow the DN renaming workflow element in the proxy architecture. This can be any type of workflow element.

### 15.1.5.1 Modifying a DN Renaming Configuration

Once you have configured DN renaming, you can modify the following DN renaming properties:

- client base DN
- source base DN
- next workflow element
- black list attributes
- white list attributes

1. To view the current DN renaming properties, use the dsconfig get-workflow-element-prop command.
2. To modify a DN renaming property, use the dsconfig set-workflow-element-prop command.

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
 set-workflow-element-prop \
 --element-name RenameorgDN \
 --set source-base-dn:ou=admin,dc=example,dc=com
```

In the preceding example, only the source-base-dn is modified. There is no need to specify the old source base DN. Only the new one is required.

To create a black list of DN attributes that should not be renamed, use the dsconfig set-workflow-element-prop command.

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  set-workflow-element-prop --element-name RenameorgDN \
  --set black-list-attributes:manager
```

The attribute must have a DN type.

## 15.1.6 Configuring RDN Changing With dsconfig

To rename RDN, create a RDN Changing workflow element, using the dsconfig create-workflow-element command.

```
dsconfig create-workflow-element \
          --set client-rdn:cn \
          --set enabled:true \
          --set next-workflow-element:localproxy \
          --set source-rdn:uid \
          --type rdn-changing \
          --element-name myrdnchangingwfe \
          --hostname localhost \
          --port "4444" \
          --trustAll \
          --bindDN cn=directory\ manager \
          --bindPasswordFile pwd-file \
          --no-prompt
```

- --set client-rdn indicates the client base RDN, which is the workflow entry point.
- --set source-rdn indicates the base RDN which the entries should have after transformation, which is the workflow exit point.
- --set next-workflow-element:localproxy indicates the workflow element that will follow the RDN changing workflow element in the proxy architecture. This can be any type of workflow element.

  **Note:** You must create the Proxy LDAP workflow element with the parameters

  - remote-root-dn
  - remote-root-password

  The RDN Changing workflow element uses these credentials to perform internal searches on the remote server.

- --element-name myrdnchangingwfe indicates the name of the RDN Changing workflow element you are creating.

  This configuration replaces uid=user.1,ou=people,dc=example,dc=com with cn=User CN,ou=people,dc=example,dc=com.

### 15.1.6.1 Modifying a RDN Renaming Configuration

Once you have configured RDN renaming, you can modify the following RDN renaming properties:

- client RDN
- source RDN
- next workflow element
- objectclass
- dn attributes
- replace-value

1. To view the current RDN renaming properties, use the `dsconfig get-workflow-element-prop` command.
2. To modify a RDN renaming property, use the `dsconfig set-workflow-element-prop` command.

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
 set-workflow-element-prop \
 --element-name myrdnchangingwfe \
 --set source-rdn:uid
```

In the preceding example, only the `source-rdn` is modified. There is no need to specify the old source-rdn. Only the new one is required.

## 15.1.7 Configuring Global Indexes By Using the Command Line

Global indexes map entries to a specific distribution partition to speed up search and modify operations in distributed topologies. A global index maps entries based on a unique attribute, such as a phone number. Lists of global indexes are contained in a global index catalog. A proxy instance can contain one or more global index catalogs.

> **Note:** To configure and manage global indexes and global index catalogs, you must enable specific controls on the remote servers, particularly the LDAP Pre-Read Control and the CSN Control. For more information, see Appendix B, "Supported Controls and Operations."

This section contains the following topics:

- Section 15.1.7.1, "Configuring Global Index Catalogs by Using `gicadm`"
- Section 15.1.7.2, "Replication of Global Index Catalogs"
- Section 15.1.7.3, "Configuring Controls Required by the Global Index Catalog with Oracle Unified Directory"

### 15.1.7.1 Configuring Global Index Catalogs by Using `gicadm`

Global index catalogs are stored in a Berkeley database under `INSTANCE_DIR/OUD/catalogs`. To ensure high availability of a distributed topology, replication of global index catalogs is recommended. For more information, see Replication of Global Index Catalogs.

The `gicadm` command is located in the server instance directory:

- for Unix: `INSTANCE_DIR/OUD/bin/gicadm`
- for Windows: `INSTANCE_DIR\OUD\bat\gicadm.bat`

For more information, see Appendix A.2.7, "gicadm."

The procedures in this section assume that the proxy is deployed in a distribution architecture and presume that you are using the default proxy administration port (4444). This section contains the following topics:

- Section 15.1.7.1.1, "To Create a Global Index Catalog Containing Global Indexes"
- Section 15.1.7.1.2, "To View Global Index Catalog Properties"
- Section 15.1.7.1.3, "Modifying the Properties of a Global Index Catalog"
- Section 15.1.7.1.4, "To Modify the Properties of a Global Index Catalog"

- Section 15.1.7.1.5, "To Modify Multi-Valued Global Index Catalog Properties"
- Section 15.1.7.1.6, "To Reset Global Index Catalog Properties To the Default Values"
- Section 15.1.7.1.7, "To View Global Index Properties"
- Section 15.1.7.1.8, "To Import Content into a Global Index Catalog"
- Section 15.1.7.1.9, "To Export Contents of a Global Index Catalog to a Directory"
- Section 15.1.7.1.10, "To Associate a Global Index Catalog With a Distribution Element"
- Section 15.1.7.1.11, "To Disassociate a Global Index Catalog From a Distribution Element"
- Section 15.1.7.1.12, "To Add a Global Index to a Global Index Catalog"
- Section 15.1.7.1.13, "To Remove a Global Index From a Global Index Catalog"

#### 15.1.7.1.1 To Create a Global Index Catalog Containing Global Indexes

To create global indexes, you must first create global index catalogs, as described in the following procedure. This procedure describes how to create global index catalogs, create and add global indexes, and add data to the global indexes. You can add the data to your global indexes later, if you prefer.

Before you begin, the proxy must be deployed for distribution.

1. Use the `gicadm` command to create a global index catalog:

   ```
   $ gicadm -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X \
   create-catalog --catalogName sampleCatalog
   ```

   The catalog name must be unique.

2. Create and add at least one global index to the global index catalog.

   The following command creates a global index of `telephoneNumber` attribute values and adds that global index to the global index catalog that was created in the previous step.

   ```
   $ gicadm -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X \
     add-index --catalogName sampleCatalog --attributeName telephoneNumber
   ```

   You can use the `add-index` subcommand later to add additional global indexes to the global index catalog.

3. Associate the global index catalog to a distribution.

   ```
   $ gicadm -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X \
     associate --catalogName sampleCatalog \
     --distributionWorkflowElement myDistributionName
   ```

   For information about workflow elements, see Section 14.1.8, "Configuring Workflow Elements With `dsconfig`." For information about distribution, see Section 15.1.4, "Configuring Distribution With `dsconfig`."

4. Use the `split-ldif` command to generate multiple files from one LDIF file.

   The `split-ldif` command separates the content of one LDIF file into several LDIF files based on the distribution algorithm configured with your proxy. It can also generate files that contain data to load in a global index. You should use `split-ldif` during global index initialization if the remote LDAP servers will

contain data that needs to be indexed when you start your Directory service. If you plan to start without data in your directory, you can skip this step.

For information on the split-ldif command, including examples on how to use the command to populate a global index with one or several indexed attributes, see Appendix A.3.15, "split-ldif."

5. Use the gicadm import command to import data into the global index.

For more information, see Section 15.1.7.1.8, "To Import Content into a Global Index Catalog."

#### 15.1.7.1.2 To View Global Index Catalog Properties

Global index catalog properties are related to global index catalog **replication**. For a list of the global index catalog properties and an explanation of their use, see Section 15.1.7.1.3, "Modifying the Properties of a Global Index Catalog."

To view all the properties of a global index catalog, use the gicadm command with the get-catalog-prop subcommand.

```
$ gicadm -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X \
  get-catalog-prop --catalogName sampleCatalog --property all
```

The output will be similar to the following.

```
Property   : Value(s)
-------------------:------------------------------
replication-server : localhost:3390, localhost:4390
server-id          : 4247
window-size        : 100
heartbeat-interval : 1000
group-id           : 1
```

To view the value for a specific global index catalog property, specify the property.

```
$ gicadm -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X \
  get-catalog-prop --catalogName sampleCatalog --property heartbeat-interval
```

#### 15.1.7.1.3 Modifying the Properties of a Global Index Catalog

Global index properties are related to the replication of global index catalogs. The following global index catalog properties are available:

- replication-server: This lists the servers in the replication topology, in the format host:port. This property should not be modified with the set-catalog-prop command, but with enable-replication.
- server-id: Specifies a unique identifier for the global index within the global index catalog replication domain. Each instance within the same global index catalog replication domain must have a different server ID. An instance which is a member of multiple global index catalog replication domains may use the same server ID for each of its global index catalog replication domain configurations.

  Syntax: 1 <= INTEGER <= 65535 or text. This property should not be modified.
- window-size: Specifies the window size that the instance will use when communicating with replication servers. Default value is 100.

  Syntax: 0 <= INTEGER or text.
- heartbeat-interval: Specifies the heartbeat interval that the instance will use when communicating with replication servers. The instance expects a regular heartbeat from the replication server within the specified interval. If a heartbeat is

not received within this interval, the instance closes its connection and connects to another replication server.

Syntax: 100 ms <= DURATION (ms)

- group-id: The id associated with a specific replicated domain. This value defines the group id of the replicated domain. The replication system will preferably connect and send updates to replicate to a replication server with the same group id as itself.

  Syntax: 1 <= INTEGER <= 127

  **Note:** This property should not be modified.

#### 15.1.7.1.4 To Modify the Properties of a Global Index Catalog

For a list of the global index catalog properties, see Section 15.1.7.1.3, "Modifying the Properties of a Global Index Catalog."

Use the gicadm command with the set-catalog-prop subcommand.

```
$ gicadm -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X \
  set-catalog-prop --catalogName sampleCatalog --set property:value
```

For example, one of the properties that can be modified is the heartbeat interval. In this case, use:

```
--set heartbeat-interval:500
```

#### 15.1.7.1.5 To Modify Multi-Valued Global Index Catalog Properties

For multi-valued global index or global index catalog properties, you can add or remove a value using the --add or --remove options.

For global index catalog, only the property replication-server can be multi-valued. For multi-valued global index properties, use the set-index-prop subcommand instead

1. To add a value, use the gicadm command with the set-catalog-prop subcommand.

   ```
   $ gicadm -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X \
     set-catalog-prop --catalogName sampleCatalog --add
   replication-server:hostname
   ```

2. To remove a value from a multi-valued property, use the gicadm command with the set-catalog-prop subcommand.

   ```
   $ gicadm -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X \
     set-catalog-prop --catalogName sampleCatalog \
     --remove replication-server:hostname
   ```

#### 15.1.7.1.6 To Reset Global Index Catalog Properties To the Default Values

If you have modified any of the global index catalog properties and want to reset them to the default values, use the following procedure.

Use the gicadm command with the set-catalog-prop subcommand.

For example, to reset the heartbeat interval:

```
$ gicadm -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X \
  set-catalog-prop --catalogName sampleCatalog --reset heartbeat-interval
```

#### 15.1.7.1.7 To View Global Index Properties

To view the properties of a global index, use the gicadm command with the get-index-prop subcommand.

```
$ gicadm -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X \
  get-index-prop --catalogName sampleCatalog --attributeName telephoneNumber \
  --property all
```

The properties should be similar to the following:

```
Property Names                                 : Property Values
-----------------------------------------------:---------------------------------
global-index-deleted-entry-retention-timeout : 500
db-cleaner-min-utilization                   : 50
db-log-file-max                              : 10000000
db-checkpointer-bytes-interval               : 20000000
db-checkpointer-wakeup-interval              : 30
db-num-lock-tables                           : -
db-num-cleaner-threads                       : -
db-txn-no-sync                               : false
db-txn-write-no-sync                         : true
je-property                                  : -
db-directory                                 : catalogs
db-directory-permissions                     : 700
global-index-catalogs-shared-cache           : global-index-catalogs-shared-cac
global-index-attribute                       : telephoneNumber
```

**Note:** Generally, these values should not be modified.

#### 15.1.7.1.8 To Import Content into a Global Index Catalog

You can import the contents of a specific file into one or multiple global indexes in a global index catalog. You must specify the name of the catalog into which the content of the file is to be imported. You can filter the content of the file to data related to a particular index by optionally providing the attributeName parameter.

The data file to be imported can be created by executing the split-ldif command or from executing the gicadm export command, for example.

To import the contents of a file into a global index catalog, use the gicadm command with the import subcommand. For example:

```
$ gicadm -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X \
  import --file /usr/local/import-file --catalogName sampleCatalog
```

If the proxy server stops while a gicadm import task is being executed, the global index catalog workflow element is disabled. In this case, re-enable the global index catalog workflow element by using dsconfig, as follows, where *sampleCatalog* is the name of the global index catalog:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  set-workflow-element-prop --element-name sampleCatalog set enabled:true
```

#### 15.1.7.1.9 To Export Contents of a Global Index Catalog to a Directory

To export the contents of a global index catalog to a directory, use the gicadm command with the export subcommand.

```
$ gicadm -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X \
```

```
export --exportDirectory directory-path --catalogName sampleCatalog
```

#### 15.1.7.1.10 To Associate a Global Index Catalog With a Distribution Element

To associate a global index catalog with a distribution element, use the gicadm command with the associate subcommand.

```
$ gicadm -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X \
  associate --catalogName sampleCatalog --distributionWorkflowElement element-name
```

When the global index catalog is associated with a distribution workflow element, the global index catalog will be listed in the properties of the distribution. To confirm which global index catalog is associated to a distribution, use the dsconfig get-workflow-element-prop command. For information on workflow elements, see Section 14.1.8, "Configuring Workflow Elements With dsconfig."

#### 15.1.7.1.11 To Disassociate a Global Index Catalog From a Distribution Element

To disassociate a global index catalog from a distribution topology, you must know the distribution workflow element with which the global index catalog is associated. To confirm the name of the distribution workflow element that is using the global index catalog, view the properties of the distribution topology by using the dsconfig --get-workflow-element-prop command.

To disassociate a global index catalog from a distribution workflow element, use the gicadm command with the disassociate subcommand.

```
$ gicadm -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X \
 disassociate --distributionWorkflowElement element-Name
```

#### 15.1.7.1.12 To Add a Global Index to a Global Index Catalog

To add a new global index to an existing global index catalog, for example to map a new attribute, use the following procedure. This procedure creates and adds the global index to the global index catalog. It is not possible to create a global index without adding it to a global index catalog.

Before you begin, you must already have configured a global index catalog.

Use the gicadm command with the add-index subcommand.

```
$ gicadm -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X \
  add-index --catalogName sampleCatalog --attributeName telephoneNumber
```

#### 15.1.7.1.13 To Remove a Global Index From a Global Index Catalog

Use the gicadm command with the remove-index subcommand.

```
$ gicadm -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X \
  remove-index --catalogName sampleCatalog --attributeName telephoneNumber
```

### 15.1.7.2 Replication of Global Index Catalogs

To ensure high availability, global index catalogs should be replicated. A standard hardware load balancer can be used and replication of global index catalogs can be configured in a deployment as shown by the graphic in Section 3.8, "Multiple Replicated Proxies."

This section contains the following topics:

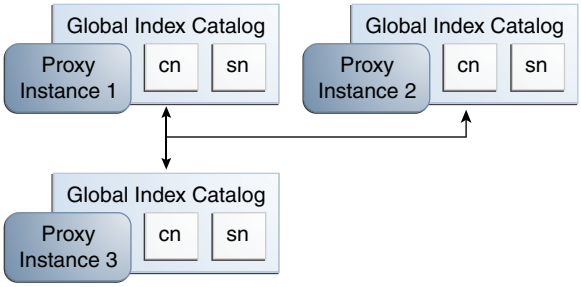
- Section 15.1.7.2.1, "To Create a Replicated Topology and Enable Global Index Catalog Replication"

- Section 15.1.7.2.2, "To Enable Global Index Catalog Replication"
- Section 15.1.7.2.3, "To Initialize Global Index Catalog Replication"
- Section 15.1.7.2.4, "To Disable Global Index Catalog Replication"
- Section 15.1.7.2.5, "To View the Status of a Replicated Global Index Catalog Configuration"
- Section 15.1.7.2.6, "Logging of Replication Activities"
- Section 15.1.7.2.7, "Lifecycle Examples for Replicated Global Index Catalogs"

#### 15.1.7.2.1 To Create a Replicated Topology and Enable Global Index Catalog Replication

Follow the steps below in order to create a replicated topology with three proxy instances, and enable global index catalog replication, as illustrated in Figure 15–1.

***Figure 15–1 Replicated Global Index Catalogs***




1. Install at least two proxy instances in your server topology.

   These instances should be on separate physical machines, for redundancy.

2. Configure a global index catalog for each instance of the proxy in your topology and add one or more global indexes.

   For more information on configuring a global index catalog using the `gicadm` command, see Section 15.1.7.1.1, "To Create a Global Index Catalog Containing Global Indexes."

3. Enable global index catalog replication.

   The proxy instance whose global index catalog is to be replicated across the topology is referred to, for the purposes of CLI syntax, as the *local* instance, while the other proxy instance declared in the command is referred to as the *remote* instance. For more information on running the `gicadm enable-replication` command, see Section 15.1.7.2.2, "To Enable Global Index Catalog Replication."

   Repeat this step for each proxy that is part of your replicated topology.

4. Choose a proxy instance on which to initialize replication. Consider which proxy instance has the most up to date global index catalog content.

   Otherwise, you can import the LDIF file to each proxy that is part of the topology. See Section 15.1.7.1.8, "To Import Content into a Global Index Catalog."

5. On the proxy instance chosen in the previous step, run the `gicadm initialize-replication --all` command. For more information, see Section 15.1.7.2.3, "To Initialize Global Index Catalog Replication."

**Note:** When using a global index catalog with replicated remote LDAP servers, only one remote LDAP server must handle write operations if such operations can concurrently modify the same value *and* if that value is indexed. For this, you could set the weights in your load balancing workflow element to direct all write traffic to the same server. For more information, see Section 15.1.3.5, "Modifying Load Balancing Properties."

#### 15.1.7.2.2 To Enable Global Index Catalog Replication

This command configures replication but does not initialize replication. The command is executed on the local host, declared by the `-h` option, using the administration port of the local host. The remote host is declared by the `--remoteHost` option, and must be a fully qualified host name or IP address. The command creates a global index catalog replication administrator with a bind ID of *adminUID*.

If you created global index catalogs during installation, the global index administrator is already created, with the same password as the directory manager. For more information on installing a distribution deployment with global index, see "To Configure Simple Distribution" section in *Oracle Fusion Middleware Installation Guide for Oracle Unified Directory*.

To enable replication of global index catalogs, use the `gicadm enable-replication` command.

```
$ gicadm -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X \
  enable-replication --catalogName sampleCatalog --adminUID adminUID
  --localReplicationPort 8989 --remoteReplicationPort 8989 \
  --remoteAdminPort 4444 --remoteHost host
```

This command updates the proxy configuration to replicate the content of the global index catalog called *sampleCatalog* on the local host. If one of the proxy instances in the topology already replicates the global index catalog, this command updates the configuration of all other proxy instances in the topology. It is therefore sufficient to execute the `gicadm enable-replication` once for the first two proxy instances in the topology, and once for each new proxy instance that is added to extend the topology.

The proxy instance on which you execute the command must be the instance whose replication port is declared by the `--localReplicationPort` option. It is this local instance whose global index catalog is replicated across the topology later by the `gicadm initialize-replication` command. The `--remoteReplicationPort` option will replicate the content of the global index catalog called *sampleCatalog* from the local instance on to the remote instance. The `--remoteAdminPort` is the administration port of the remote proxy instance.

You can declare the password for the local proxy instance in a file, by using the `--adminPasswordFile` suboption.

You can optionally declare a DN for binding to the remote server by using the `--remoteBindDN` suboption and the password for the remote proxy instance in a file, by using the `--remoteBindPasswordFile` suboption. If you do not declare these, the global administrator that is declared by `--adminUID` will be used to bind.

You can also optionally require the communication through the replication port of the local server to be secure, using the `--localSecureReplication` suboption, and the communication through the replication port of the remote server to be secure, using the `--remoteSecureReplication` suboption.

#### 15.1.7.2.3 To Initialize Global Index Catalog Replication

This command initializes the content of the global index catalog called *sampleCatalog* from the proxy instance on the server declared by the `-h` option to all instances that are part of the topology. The port specified is the administration port, and not the replication port.

1. To initialize the replication of a global index catalog to all proxy instances that are part of the replication topology, use the `gicadm initialize-replication --all` as follows:

```
$ gicadm -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X \
  initialize-replication --catalogName sampleCatalog \
  --adminUID adminUID --all
```

2. Check that replication is complete by using the `gicadm status-replication` command.

   If replication is complete, the status for all proxy instances in the topology is given as `running replicated`.

   Replication must be complete before restarting any proxy instances in the topology, for example after applying a patch.

   For information about using the `gicadm status-replication` command, see Section 15.1.7.2.5, "To View the Status of a Replicated Global Index Catalog Configuration."

#### 15.1.7.2.4 To Disable Global Index Catalog Replication

To disable replication of global index catalogs, use the `gicadm disable-replication` command.

```
$ gicadm -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X \
  disable-replication --catalogName sampleCatalog --adminUID adminUID
```

The `gicadm disable-replication` command must be executed for each proxy instance in the topology on which you want to disable replication.

#### 15.1.7.2.5 To View the Status of a Replicated Global Index Catalog Configuration

To display basic configuration information about a replicated global index catalog, use the `gicadm status-replication` command.

```
$ gicadm -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X \
  status-replication --catalogName sampleCatalog --adminUID adminUID
```

If you do not declare a catalog name, status information for all replicated global index catalogs is displayed.

#### 15.1.7.2.6 Logging of Replication Activities

Replication logs are stored in the replication repair logs. Changes are recorded in the change logs. For information on accessing these logs, see Section 29.5.4, "Accessing Logs."

When replicating global index catalogs, provision disk space for change logs. By default, these logs store changes for a 24 hour period. Approximately 100Mb is required for 300,000 write operations. With the default value of 24 hours, the log must be configured based on the expected size of the service during that period. A hint is to provision approximately 150Gb for 5 000 modifications per second over 24 hours. For information how to configure logs, see Section 29.3, "Configuring Logs."

#### 15.1.7.2.7 Lifecycle Examples for Replicated Global Index Catalogs

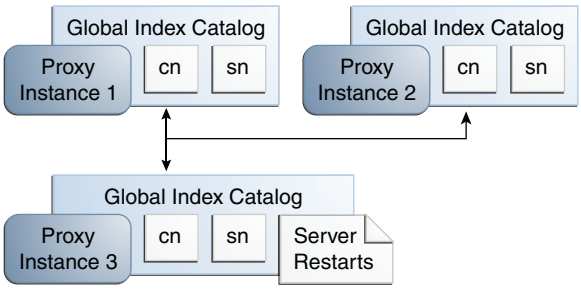
This section describes several typical lifecycle examples in which events take place in a replication topology. The basic replication topology used in all of these examples is the one created in Section 15.1.7.2.1, "To Create a Replicated Topology and Enable Global Index Catalog Replication."

***Example 15–2 To Restart a Global Index Catalog in a Replicated Topology***

In the example illustrated by Figure 15–2, three proxy instances are running with a replicated global index catalog. If proxy instance 3 goes down or is stopped, for whatever reason, follow these steps to ensure that the three instances of the proxy are replicated.

1. Issue the `start-ds` command on proxy instance 3.
2. You can check to see if replication is complete by executing the `gicadm status-replication` command, as described in Section 15.1.7.2.5, "To View the Status of a Replicated Global Index Catalog Configuration."

***Figure 15–2 Restarting a Global Index Catalog***

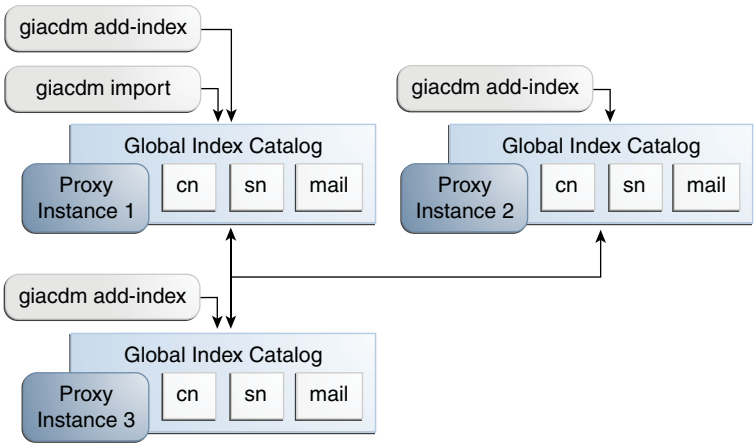



***Example 15–3 Adding a Global Index to a Replicated Global Index Catalog Topology***

In the example illustrated by Figure 15–3, three proxy instances are running with a replicated global index catalog. If you want to add an additional attribute, for example, `mail`, to the replicated global index catalog, follow these steps.

1. First, run the command `gicadm add-index mail` on each of the three proxy instances.
2. Export the directory data under the distribution route from one of the remote LDAP servers to an LDIF file named *file1* by using `export-ldif`.
3. Run `split-ldif` to generate GIC content in the specified directory.
4. On proxy instance 1, execute the command `gicadm import --importDirectory directory-name`.
5. On proxy instance 1, execute the `gicadm initialize-replication --all` command. This command pushes the changes from proxy 1 to all the other proxies in the topology, and adds the new global index.

***Figure 15–3 Adding a Global Index to a Replicated Global Index Catalog Topology***

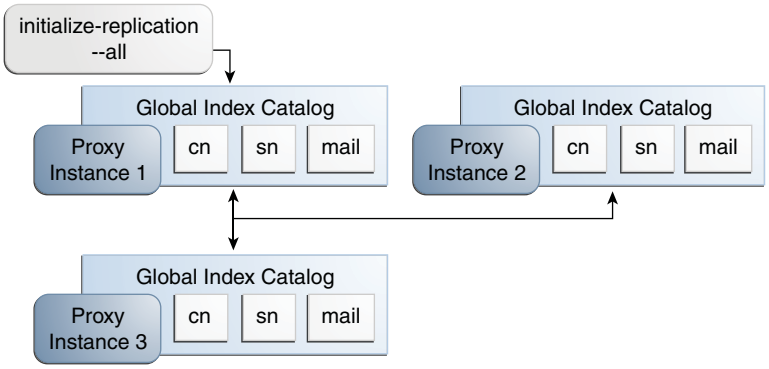



***Example 15–4 Overwriting the Contents of Replicated Global Index Catalogs***

In the example illustrated by Figure 15–4, three proxy instances are running with a replicated global index catalog. To overwrite the content of the global index catalogs on proxy instances 2 and 3 with the content of the global index catalog on instance 1, follow these steps.

1. On proxy instance 1, execute the `gicadm initialize-replication --all` command. This replaces the content of the global index catalog on proxy instance 2 and 3 with the content of the global index catalog on proxy instance 1.

***Figure 15–4 Overwriting the Contents of Replicated Global Index Catalogs***




***Example 15–5 Adding a Proxy to a Replicated Topology***

In the example illustrated by Figure 15–5, three proxy instances are running with a replicated global index catalog. To add a fourth proxy instance with a replicated global index catalog, follow these steps on the new proxy instance.

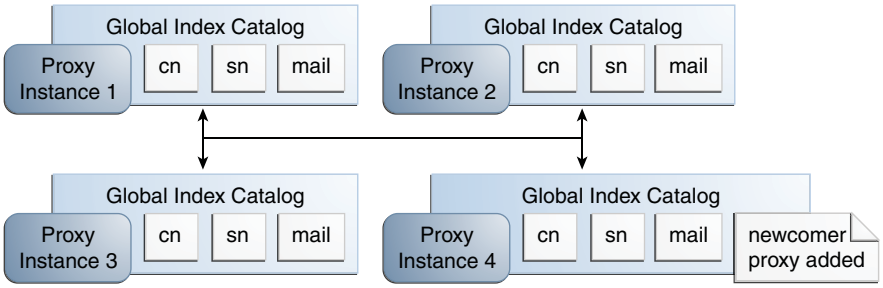
1. On the new proxy instance 4, execute the `gicadm create-catalog` command.
2. Run the commands`gicadm add-index cn`, `gicadm add-index sn`, and `gicadm add-index mail`.
3. Execute the `gicadm associate` command.
4. Run the following command:

```
gicadm enable-replication --localReplicationPort replication port of instance 4
--remoteHost name or IP address of host running instance 1
```

This command configures replication between instance 1 and instance 4.

5. Run the `initialize replication --from proxy 1` command.

***Figure 15–5 Adding a Proxy to a Replicated Topology***




### 15.1.7.3 Configuring Controls Required by the Global Index Catalog with Oracle Unified Directory

If you are using the proxy server with an Oracle Unified Directory directory server as the LDAP data source, the connections between the proxy and directory servers must be bound using the directory server's administrator ID. Otherwise, some configuration is required on the directory server to allow the global index catalog to function correctly.

Provided that global ACIs for controls have not been modified, use the `ldapmodify` command to apply the following changes to the directory server:

```
dn: cn=Access Control Handler,cn=config
changetype: modify
add: ds-cfg-global-aci
ds-cfg-global-aci:
(targetcontrol="2.16.840.1.113730.3.4.2 || 2.16.840.1.113730.3.4.17 |
| 2.16.840.1.113730.3.4.19 || 1.3.6.1.4.1.4203.1.10.2 || 1.3.6.1.4.1.42.2.27.8.5.1
|
| 2.16.840.1.113730.3.4.16 || 1.3.6.1.1.13.1 || 1.3.6.1.4.1.42.2.27.9.5.9")
(version 3.0; acl "Anonymous control access"; allow(read)
userdn="ldap:///anyone";)
 ds-cfg-global-aci: (targetattr="createTimestamp||creatorsName||modifiersName|
|modifyTimestamp||entryDN||entryUUID||subschemaSubentry||aclRights||aclRightsInfo"
)
(version 3.0; acl "User-Visible Operational Attributes"; allow
(read,search,compare)
 userdn="ldap:///anyone";)
```

If you are deleting the ACI from an Oracle Unified Directory 11*g* R1 directory instance, then you need to delete the following ACI:

```
dn: cn=Access Control Handler,cn=config
changetype: modify
delete: ds-cfg-global-aci
ds-cfg-global-aci: (targetcontrol="2.16.840.1.113730.3.4.2 ||
2.16.840.1.113730.3.4.17 || 2.16.840.1.113730.3.4.19 || 1.3.6.1.4.1.4203.1.10.2 ||
1.3.6.1.4.1.42.2.27.8.5.1 || 2.16.840.1.113730.3.4.16") (version 3.0; acl
"Anonymous control access"; allow(read) userdn="ldap:///anyone";)
ds-cfg-global-aci:
(targetattr="createTimestamp||creatorsName||modifiersName||modifyTimestamp||entryD
N||entryUUID||subschemaSubentry")(version 3.0; acl "User-Visible Operational
Attributes"; allow (read,search,compare) userdn="ldap:///anyone";)
```

If you are deleting the ACI from an Oracle Unified Directory 11*g* R2 directory instance, then you need to delete the following ACI:

```
dn: cn=Access Control Handler,cn=config
changetype: modify
delete: ds-cfg-global-aci
ds-cfg-global-aci: (targetcontrol="2.16.840.1.113730.3.4.2 ||
2.16.840.1.113730.3.4.17 || 2.16.840.1.113730.3.4.19 || 1.3.6.1.4.1.4203.1.10.2 ||
1.3.6.1.4.1.42.2.27.8.5.1 || 2.16.840.1.113730.3.4.16 ||
2.16.840.1.113894.1.8.31") (version 3.0; acl "Anonymous control access";
allow(read) userdn="ldap:///anyone";)
```

Note that the OIDs provided above are correct for an unmodified configuration of Oracle Unified Directory. If the default OIDs have been changed, modify the command include the correct OIDs.

The following controls are required for global index catalogs:

- The Pre-Read Control, with OID = `1.3.6.1.1.13.1`
- The CSN Control, with OID = `1.3.6.1.4.1.42.2.27.9.5.9`

## 15.1.8 Configuring Microsoft Active Directory Paging

Retrieving the contents of a multi-valued attribute sometimes result in a large number of returned values. Microsoft Active Directory server limits the maximum number of attribute values that can be retrieved in a single query.

Microsoft Active Directory provides a **range retrieval** mechanism that allows you to retrieve all the values of a multi-valued attribute. This mechanism permits a client-specified subset of the values to be retrieved in a search request. By performing multiple search requests, each retrieving a distinct subset, the complete set of values for the attribute can be retrieved.

Oracle Unified Directory handles Active Directory range retrieval by providing support for Microsoft Active Directory paging. The main purpose of Microsoft Active Directory paging is to detect if a range option is present among the options of the returned attributes and to retrieve the complete range of attribute values from the Microsoft Active Directory server. This complete set of attribute values is returned, so that the client application does not have to deal with the range option.

Microsoft Active Directory paging is implemented as a workflow element that is relevant only if the leaf of the workflow element chain is connected to an Active Directory server. You can configure the following properties of an Active Directory Paging workflow element:

- The next workflow element in the chain as this workflow element is not a leaf workflow element
- An optional list of attributes that can reduce the processing of scanning attributes to detect the range option

### 15.1.8.1 Configuring Active Directory Paging Workflow Elements

To configure support for Microsoft Active Directory paging, create and enable an Active Directory paging workflow element that points to an LDAP proxy workflow element.

The following example creates an Active Directory paging workflow element named `ad-paging-we1` that points to the LDAP proxy workflow, `proxy-we1`.

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  create-workflow-element --element-name ad-paging-we1 --type ad-paging \
  --set next-workflow-element:proxy-we1 --set enabled:true
```

#### 15.1.8.2 Scanning Specific Attributes Returned by an Active Directory

To improve efficiency, you can configure the Active Directory paging workflow element to scan only specific attributes by setting the multi-valued `handled-attributes` property of the workflow element. You can add as many values for this property as required.

By default all attributes are scanned. This can have a direct impact on performance. To reduce the performance impact, list only the attributes that need to be scanned as values of the `handled-attributes` property.

The following example modifies the workflow element created in the previous example to scan only for the `memberOf` attribute:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  set-workflow-element-prop --element-name ad-paging-we1 \
  --set handled-attributes:memberOf
```

## 15.2 Managing the Proxy Configuration With ODSM

This section describes the elements of the proxy server configuration that can be managed with Oracle Directory Services Manager, and covers the following topics:

- Section 15.2.1, "Configuring Load Balancing With ODSM"
- Section 15.2.2, "Configuring Distribution With ODSM"
- Section 15.2.3, "Configuring Criticality in Workflows With ODSM"
- Section 15.2.4, "Configuring Transformations With ODSM"

### 15.2.1 Configuring Load Balancing With ODSM

If you have set up a proxy server instance without configuring either load balancing or distribution, you can configure load balancing by using ODSM. Before you begin, it is useful to understand the components that make up a load balancing deployment. For more information, see Section 3.2, "Configuration 1: Simple Load Balancing."

To configure load balancing by using ODSM, perform the following steps:

1. Connect to the proxy server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager."
2. Select the **Home** tab.
3. Under the **Configuration** item, select **Set up Load Balancer**.
4. On the **Load Balancing: Backend Servers** screen, complete the following information:
   - In the **Load Balancing Name** field, provide a name for this load balancing workflow element.
   - Click **Add** to provide the connection details of at least two replicated backend LDAP servers across which client requests will be balanced.

     ODSM attempts to connect to these backend LDAP servers, to verify that they are accessible. If the connection attempt is unsuccessful, you are prompted to use the server details anyway, or to verify the connection details.

5. When you have added all the backend LDAP servers, click **Next** to continue.
6. On the **Load Balancing: Options** screen, complete the following information:
   - Select the **Load Balancing Algorithm**.
   - Depending on the load balancing algorithm you have selected, specify the relative weight or priority for each backend LDAP server.

   For information about the load balancing algorithms, see Section 11.1, "Load Balancing Using the Proxy."
7. When you have specified the load balancing options, click **Next** to continue.
8. On the **Load Balancing: Naming Contexts** screen, click **Add** to specify at least one naming context, or suffix, that will be handled by this proxy instance.
9. When you have added all of the required naming contexts, click **Next** to continue.
10. On the **Load Balancing Setup: Summary** screen, review the load balancing configuration and click **Finish** to complete the configuration.

When you have configured load balancing, you can modify any aspect of the configuration on the ODSM Configuration tab.

### 15.2.2 Configuring Distribution With ODSM

If you have set up a proxy server instance without configuring either load balancing or distribution, you can configure distribution by using ODSM. Before you begin, it is useful to understand the components that make up a distribution deployment. For more information, see Section 3.3, "Configuration 2: Simple Distribution."

To configure distribution by using ODSM, perform the following steps:

1. Connect to the proxy server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager."
2. Select the **Home** tab.
3. Under the **Configuration** item, select **Set Up Distributor**.
4. On the **Distribution: Data Partitioning** screen, complete the following information:
   - Select the **Number of Partitions**.
   - Select the **Distribution Algorithm**. For more information about the available distribution algorithms, see Section 11.2, "Data Distribution Using the Proxy."
   - Enter the **Naming Context**, or suffix, that will be handled in this distribution deployment.
   - Select the **Network Group** in which the distributor will be configured.
   - Enter the capacity, DN Pattern, or boundaries for each partition, depending on the distribution algorithm that you have selected.
5. When you have entered all of the partition details, click **Next** to continue.
6. On the **Distribution: Server Partitions**, for each partition, click **Add** to enter the connection details of each backend LDAP server that will hold the partitioned data.

   ODSM attempts to connect to these backend LDAP servers, to verify that they are accessible. If the connection attempt is unsuccessful, you are prompted to use the server details anyway, or to verify the connection details.

7. When you have added all of the required servers, click **Next** to continue.
8. On the **Distribution: Global Index** screen, specify the global index details. For more information about global indexes, see Section 11.3, "Global Index Catalog."
9. When you have configured the global index, click **Next** to continue.
10. On the **Distribution: Summary** screen, review the distribution configuration and click **Finish** to complete the configuration.

When you have configured distribution, you can modify any aspect of the configuration on the ODSM Configuration tab.

## 15.2.3 Configuring Criticality in Workflows With ODSM

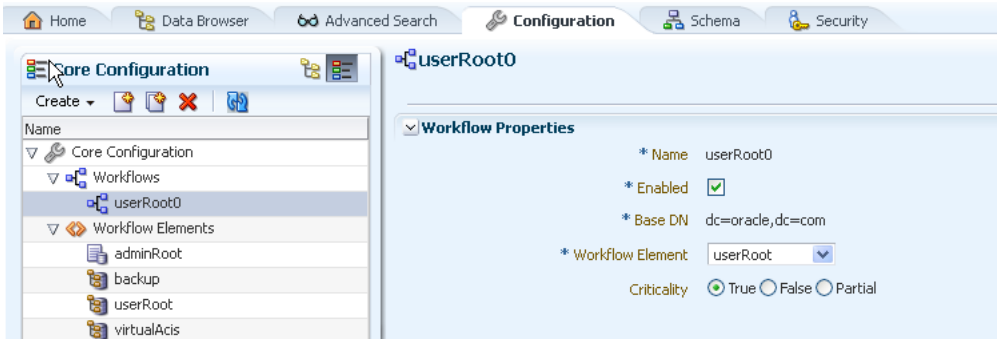
A new parameter known as the criticality flag is added to configure workflows. By default, the criticality flag is set `True`.

The following sections describe how to configure criticality in workflows using ODSM. For information about configuring criticality using dsconfig, Section 15.1.4.6, "Configuring Criticality in Workflows."

To configure criticality in workflows using ODSM, perform the following steps:

1. Connect to the proxy server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager."
2. Select the **Configuration** tab.
3. Select the **Core Configuration** view.
4. Under **Workflows** element, select the required workflow for which you want to set the criticality flag.
5. Select the criticality value (`True`, `False`, or `Partial`) that you want to set for the workflow. For instance, click **True** to set the criticality for the selected workflow element.

***Figure 15–6 Criticality Flag***




## 15.2.4 Configuring Transformations With ODSM

ODSM allows you to create, modify, and delete Transformations workflow elements for Oracle Unified Directory Proxy Servers. For more information about transformation workflow element, see Section 14.2.4, "Configuring Workflow Elements With ODSM."

The following sections describe how to configure transformation using ODSM. For information about configuring transformation using dsconfig, Section 11.6.3.2, "Example of Configuring Transformation Using CLI."

This section contains the following topics:

- Section 15.2.4.1, "Create Transformations"
- Section 15.2.4.2, "Modifying Transformations"
- Section 15.2.4.3, "Deleting Transformations"
- Section 15.2.4.4, "Selecting Values from Value Definition Screen"

### 15.2.4.1 Create Transformations

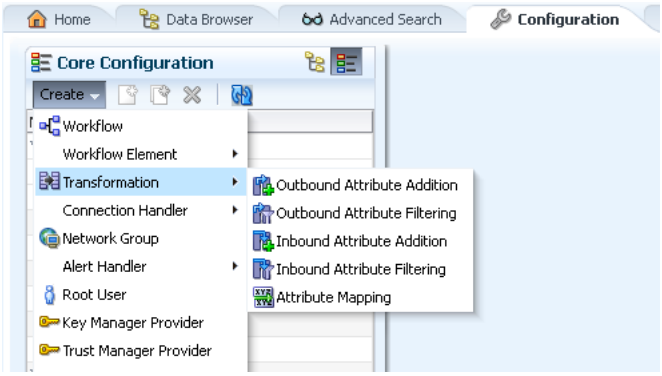
If you are connected to an Oracle Unified Directory Proxy Server, then ODSM allows you to create five different types of transformations. For more information about the types of transformations supported, see Section 11.6.2.1, "Transformation Types."

> **Note:** If you are connected to an Oracle Unified Directory server instance, then the option to create a new Transformation is not available because transformation functionality is supported by proxy servers only.

To create a transformation using ODSM, follow these steps:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager."
2. Select the **Configuration** tab.
3. Select the **Core Configuration** view.
4. From the **Create** menu, select **Transformation.**
5. From the **Transformation** submenu, select the desired transformation type.

***Figure 15–7 Transformation Types***




In this example, consider the following properties for an Outbound Attribute Addition transformation type.

> **Note:** The properties that appear while creating a transformation vary depending on the type of transformation you create. For more information about each transformation type and the associated properties, see Section 11.6.2.1, "Transformation Types."

6. In the **Name** field, type the name for the transformation.

7. In the Conditions region, enter the following information:

   **Note:** Conditions are optional. However, at runtime conditions specified here at the transformation level are used in conjunction with those specified at the transformation workflow element level in the transformation workflow element where the transformation is used. For more information about transformation workflow element, see Section 14.2.4, "Configuring Workflow Elements With ODSM."

   a. In the **Entry Matching Filter** field, type a valid LDAP filter.

   b. In the **Entry Parent Suffixes** box, click **Add** to specify the DN that must be an ascendant.

      To select an entry, click **Select.**

      In the **Entry Picker** window, select **Tree View** to navigate the directory tree and locate the entry, or **Search View** to search for the entry.

   c. From the **Excluded Operations** list, select the operations that you want to exclude.

8. In the **Transformation Definition** region, enter the following information:

   a. In the **Client Attribute** field, type the name of the client virtual attribute.

      To select a client attribute entry, click **Select.**

      In the **Attribute Picker** window, select locate the desired entry, or Click **Search** to search for the entry.

   b. In the **Value Definitions** box, click **Add** to specify the value definitions of the client virtual attribute.

      Click **Define** to enter an appropriate value definition. For more information about specifying value definitions, see Section 15.2.4.4, "Selecting Values from Value Definition Screen."

9. From the **Conflict Behavior** list, select the desired conflict behavior policy.

10. Click **Virtual in Source** to **Yes.**

11. Click **Create.**

### 15.2.4.2 Modifying Transformations

This section describes how to modify the properties for a transformation. In this example, modify the properties for an Outbound Attribute Addition transformation type created in Section 15.2.4.1, "Create Transformations."

To modify a transformation, perform the following steps:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager."

2. Select the **Configuration** tab.

3. Select the **Core Configuration** view.

4. Expand the **Transformations** element.

5. Click the desired transformation.

   Transformation configuration details appear for modification in the right pane.

6. Modify the required information.
7. Click **Apply.**

#### 15.2.4.3 Deleting Transformations

To delete a transformation, perform the following steps:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager."
2. Select the **Configuration** tab.
3. Select the **Core Configuration** view.
4. Expand the **Transformations** element.
5. Select the desired transformation to delete.

   The Delete configuration window appears seeking confirmation before deleting.
6. Click **OK.**

#### 15.2.4.4 Selecting Values from Value Definition Screen

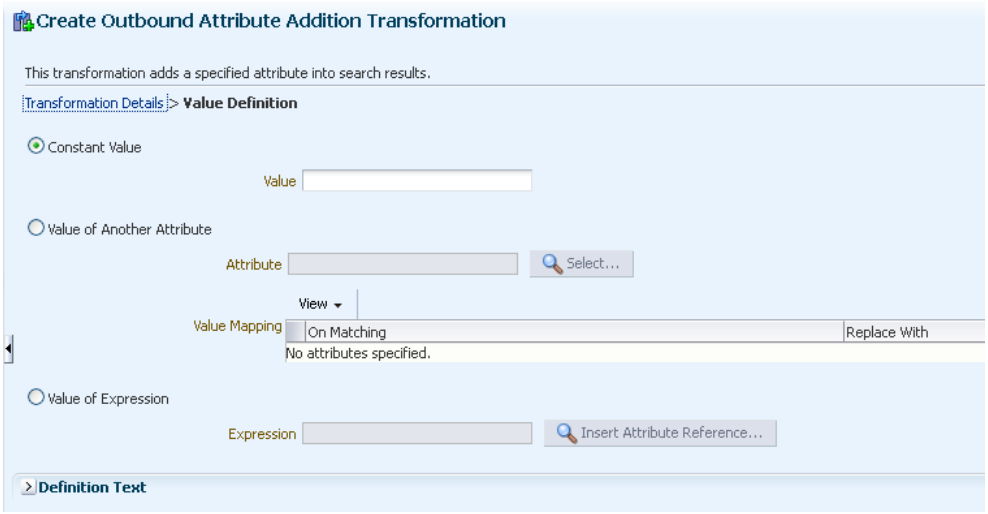
The Value Definition Builder subscreen allows you to define a value for an attribute that is being added, mapped, or deleted by a transformation.

You can specify the following values:

- Constant value: It is used to enter a constant value.
- Value of another attribute: It is used to create a new attribute from an existing attribute in the entry that is being processed or to filter a value taken from another attribute.
- Value of expression: It is used to create an attribute value or to filter an attribute value by manipulating the value of one or more existing attributes.

Figure 15–8 shows the Value Definition screen.

***Figure 15–8 Value Definition Screen***



# 16

# Example Proxy Configurations

This chapter illustrates how to configure specific proxy deployments by using the `dsconfig` command. You can also perform the configuration in interactive mode. For information, see Section 14.1.2, "Using `dsconfig` in Interactive Mode."

This chapter includes the following examples:

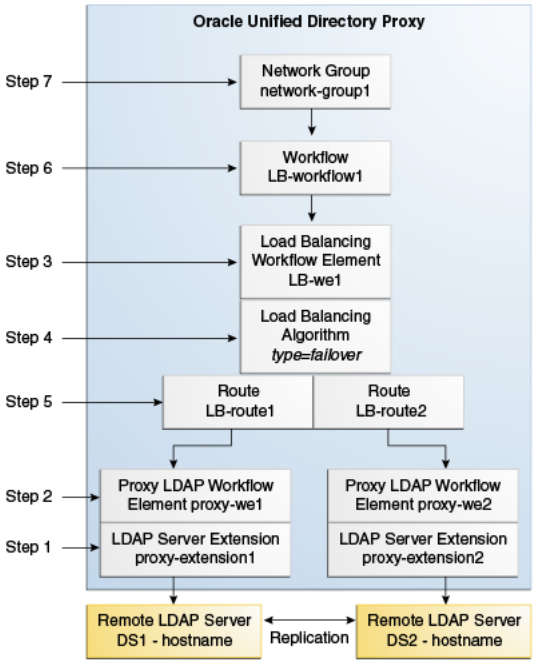
- Section 16.1, "Configuring Load Balancing"
- Section 16.2, "Configuring Distribution"
- Section 16.3, "Configuring Distribution and Load Balancing"
- Section 16.4, "Configuring Failover Between Data Centers"
- Section 16.5, "Configuring Distribution with Failover Between Data Centers"

## 16.1 Configuring Load Balancing

The following is a step by step procedure that defines all the different elements needed to set up a deployment using simple load balancing. The following example describes load balancing with failover on two LDAP servers. For more information on the different types of load balancing available, see Section 11.1, "Load Balancing Using the Proxy."

The following figure illustrates the objects that must be created to configure a proxy server for simple load balancing. The objects must be created in the order indicated.

***Figure 16–1 Load Balancing***




All the commands in this procedure specify the proxy hostname (`-h`), the proxy admin port (`-p`), the bind DN for the initial root user (`-D`) and the file containing the proxy password (`-j`). You must also indicate the authentication; if none is indicated and the client and the server are running in the same instance, the local authentication configuration is used.

## 16.1.1 To Configure Simple Load Balancing

1. Create a proxy LDAP server extension.

```
$ dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file \
create-extension \
--extension-name proxy_extension1 \
--type ldap-server \
--set enabled:true \
--set remote-ldap-server-address:DS1_hostname \
--set remote-ldap-server-port:2389
```

   The LDAP server extension is a link to the remote LDAP server. For this use case, you will need at least two remote LDAP server instances. Go through this step again, making sure to use a different LDAP hostname and port.

2. Create a proxy workflow element for each LDAP server extension.

```
$ dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file \
create-workflow-element \
--element-name proxy-we1 \
--type proxy-ldap\
--set enabled:true \
--set client-cred-mode:use-client-identity \
--set ldap-server-extension:proxy_extension1
```

The property `client-cred-mode` indicates the type of authentication used between the proxy and remote LDAP server. The client credential mode can be: `use-client-identity` or `use-specific-identity`.

3. Create a load balancing workflow element.

```
$ dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file \
create-workflow-element \
--element-name LB-we1 \
--type load-balancing \
--set enabled:true
```

You only need one load balancing workflow element to route requests to either of the two remote LDAP servers.

4. Define the load balancing algorithm.

```
$ dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file \
create-load-balancing-algorithm \
--element-name LB-we1 \
--type failover
```

The type of load balancing algorithm can be `proportional`, `saturation`, `optimal`, `searchfilter` or `failover`. The properties of the load balancing algorithm (weight, threshold, or priority) are defined with the load balancing routes, in the next step.

5. Define the load balancing routes for each proxy.

```
$ dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file \
create-load-balancing-route \
--element-name LB-we1 \
--route-name LB-route1 \
--type failover \
--set workflow-element:proxy-we1 \
--set add-priority:1 \
--set bind-priority:2 \
--set compare-priority:2 \
--set delete-priority:1 \
--set extended-priority:2 \
--set modify-priority:1 \
--set modifydn-priority:1 \
--set search-priority:2
```

Make sure that you specify the same type when defining the routes as you did when defining the load balancing algorithm.

For this use case, you will need two load balancing routes. Go through this step again, specifying a different priority for each route.

The properties in the example above set the priority for failover load balancing. If you use proportional or saturation load balancing, the properties will differ. For more information on the setting different load balancing types, see Section 15.1.3.5, "Modifying Load Balancing Properties."

6. Create a workflow.

This workflow associates the load balancing workflow element with the specified base dn.

```
$ dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file \
create-workflow \
--workflow-name LB-workflow1 \
```

```
--set enabled:true \
--set base-dn:dc=example,dc=com \
--set workflow-element:LB-we1
```

7. Create the network group.

   The network group handles all the requests between the client and the proxy.

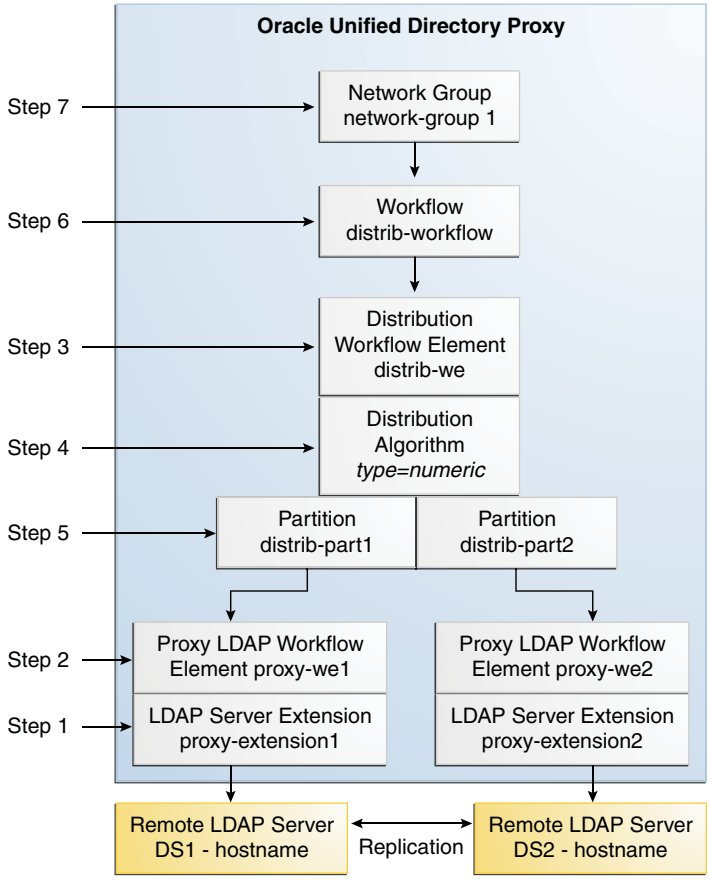
```
$ dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file \
create-network-group \
--group-name network-group1 \
--set enabled:true \
--set workflow:LB-workflow1 \
--set priority:1
```

## 16.2 Configuring Distribution

The following procedure describes the elements that are required to set up a simple distribution deployment. The example shows distribution split over two partitions. For information about the supported distribution types, see Section 11.2, "Data Distribution Using the Proxy."

The following figure illustrates the objects that must be created to configure a proxy server for simple distribution. The objects must be created in the order indicated.

***Figure 16–2 Configuring Distribution***



All the commands in this procedure specify the proxy hostname (`-h`), the proxy admin port (`-p`), the bind DN for the initial root user (`-D`) and the proxy password you want to configure (`-w`). You must also indicate the authentication; if none is indicated and the client and the server are running in the same instance, the local authentication configuration is used.

### 16.2.1 To Configure Simple Distribution

1. Create a proxy LDAP server extension.

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X \
create-extension \
--extension-name proxy_extension1 \
--type ldap-server \
--set enabled:true \
--set remote-ldap-server-address:DS1_hostname \
--set remote-ldap-server-port:2389
```

   The LDAP server extension is a link to the remote LDAP server. For this use case, you will need two remote LDAP server instances. Go through this step again, making sure to use a different LDAP hostname and port.

2. Create a proxy workflow element for each LDAP server extension.

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X \
create-workflow-element \
--element-name proxy-we1 \
--type proxy-ldap\
--set enabled:true \
--set client-cred-mode:use-client-identity \
--set ldap-server-extension:proxy_extension1
```

   You will need at least two remote LDAP servers for a distribution architecture. Go through this step again. The LDAP server extension name should be the same as those created in step 1.

   The property `client-cred-mode` indicates the type of authentication used between the proxy and remote LDAP server. The client credential mode can be: `use-client-identity` or `use-specific-identity`.

3. Set up distribution by creating a distribution workflow element.

```
$ dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file \
create-workflow-element \
--element-name distrib-we \
--type distribution \
--set base-dn:dc=example,dc=com \
--set enabled:true
```

4. Set the distribution algorithm.

```
$ dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file \
create-distribution-algorithm \
--element-name distrib-we \
--type numeric \
--set distribution-attribute:uid
```

   The type of distribution algorithm can be `capacity`, `numeric`, `lexico`, or `dnpattern`. The properties of the algorithm are defined when you create the distribution partitions, in the next step.

5. Define the distribution partitions.

```
$ dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file \
create-distribution-partition \
--element-name distrib-we \
--partition-name distrib-part1\
--type numeric \
--set lower-bound:0 \
--set upper-bound:1000 \
--set partition-id:1 \
--set workflow-element:proxy-we1
```

For this use case, you will need to create two partitions. Make sure that the partition ID and the partition name are unique for each workflow element. You must specify the same type when defining the partitions as you did when defining the distribution algorithm.

---

**Note:** The upper boundary indicated is exclusive. This means that if you indicate 1000 as the upper boundary, the partition will only include values from 0 to 999, inclusive.

---

**To create a global index**

Depending on the type of distribution algorithm defined, you need to create a global index. If you created a capacity algorithm, then you must create a global index.

For lexico, numeric, and dnpattern, a global index is optional.

Perform the following steps to create a global index:

**a.** Create a global index catalog.

```
$ gicadm -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file \
create-catalog \
--catalogName gi-catalog
```

**b.** Add a global index which indexes the dn attribute to the catalog.

```
$ gicadm -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file \
add-index \
--catalogName gi-catalog \
--attributeName dn
```

**c.** Associate the global index catalog to the distribution.

```
$ gicadm -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file \
associate \
--catalogName gi-catalog \
--distributionWorkflowElement distrib-we
```

**6.** Create a workflow.

This workflow associates the distribution workflow element with the distribution partition.

```
$ dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file \
create-workflow \
--workflow-name distrib-workflow \
--set enabled:true \
--set base-dn:dc=example,dc=com \
--set workflow-element:distrib-we
```

7. Create the network group.

   The network group handles all the requests between the client and the proxy.

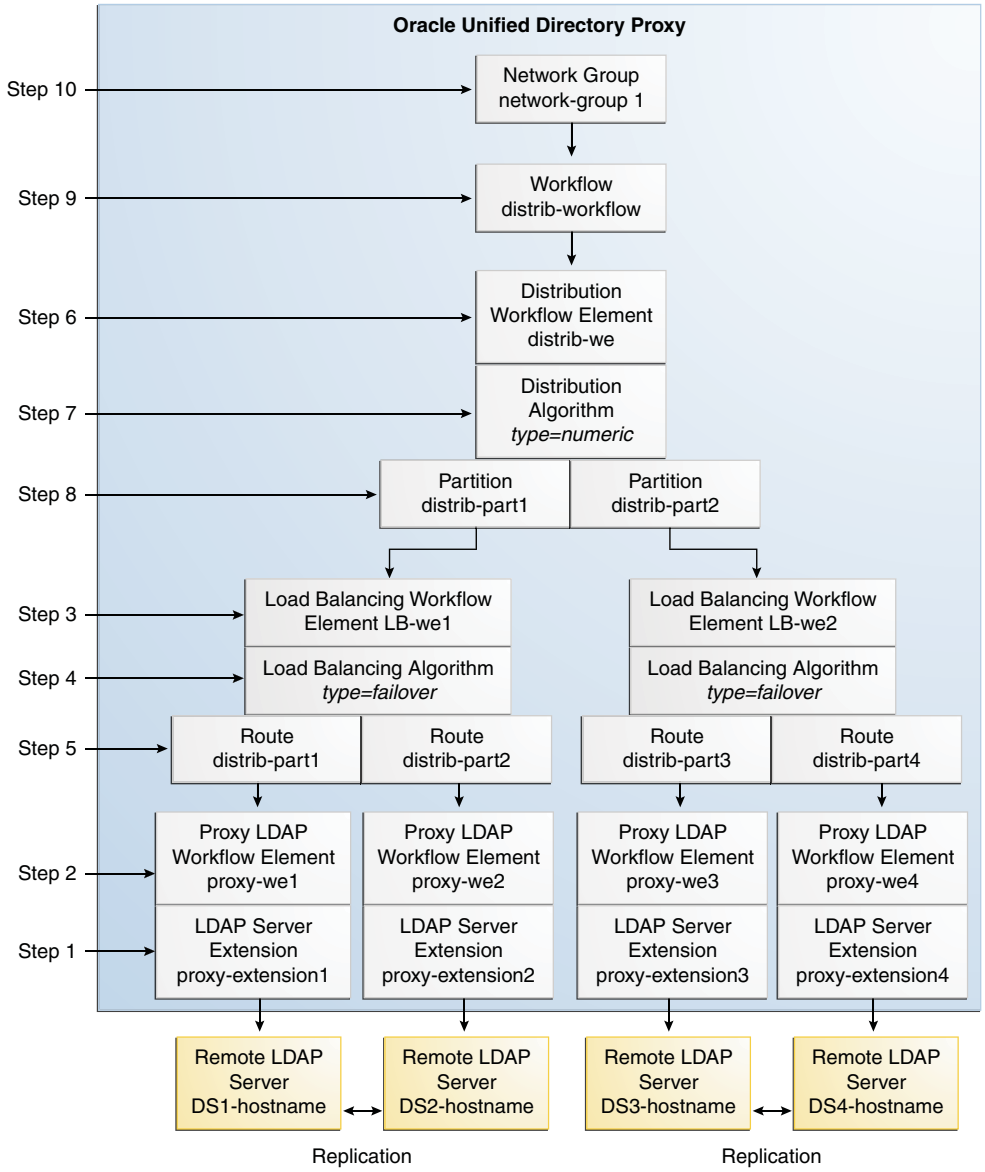
```
$ dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file \
create-network-group \
--group-name network-group1 \
--set enabled:true \
--set workflow:distrib-workflow \
--set priority:1
```

## 16.3 Configuring Distribution and Load Balancing

This use case combines distribution with load balancing. As for all distribution deployments, you can add a global index, however, this is not included here. For information about creating a global index, see Section 15.1.7, "Configuring Global Indexes By Using the Command Line."

The following figure illustrates the objects that must be created to configure a proxy server for distribution with load balancing. The objects must be created in the order indicated.

***Figure 16–3 Configuring Distribution and Load Balancing***




The following example presents a deployment with distribution over two partitions, with each partition load balanced onto two replicated LDAP servers. The distribution algorithm used to partition the data is numeric.

All the commands in this procedure specify the proxy hostname (`-h`), the proxy admin port (`-p`), the bind DN for the initial root user (`-D`) and the file containing the proxy password (`-j`). You must also indicate the authentication; if none is indicated and the client and the server are running in the same instance, the local authentication configuration is used.

## 16.3.1 To Configure Distribution with Load Balancing

1. Create the proxy LDAP server extensions.

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X \
create-extension \
--extension-name proxy_extension1 \
```

```
--type ldap-server \
--set enabled:true \
--set remote-ldap-server-address:DS1_hostname \
--set remote-ldap-server-port:2389
```

The LDAP server extension is a link to the remote LDAP server. For this use case, you will need four remote LDAP server instances. Go through this step once for each remote LDAP server, making sure to use a different LDAP hostname and port.

2. Create a proxy workflow element for each LDAP server extension.

```
$ dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file \
create-workflow-element \
--element-name proxy-we1 \
--type proxy-ldap\
--set enabled:true \
--set client-cred-mode:use-client-identity \
--set ldap-server-extension:proxy_extension1
```

For this use case, you will need four remote LDAP server instances. Go through this step once for each remote. The LDAP server extension name should be the same as those created in step 1.

The property `client-cred-mode` indicates the type of authentication used between the proxy and remote LDAP server. The client credential mode can be: `use-client-identity` or `use-specific-identity`.

3. Create a load balancing workflow element.

```
$ dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file \
create-workflow-element \
--element-name LB-we1 \
--type load-balancing \
--set enabled:true
```

You only need one load balancing workflow element to route requests to either of the two remote LDAP servers. In this use case, since you are using two load balancers, you will need to create two load balancing workflow elements.

4. Define the load balancing algorithm.

```
$ dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file \
create-load-balancing-algorithm \
--element-name LB-we1 \
--type failover
```

The type of load balancing algorithm can be `proportional`, `optimal`, `saturation`, `searchfilter`, or `failover`. The properties of the load balancing algorithm (weight, threshold, or priority) are defined with the load balancing routes, in the next step. For this use case, you will need two load balancing algorithms.

5. Define the load balancing routes for each proxy.

```
$ dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file \
create-load-balancing-route \
--element-name LB-we1 \
--route-name LB-route1 \
--type failover \
--set workflow-element:proxy-we1 \
--set add-priority:1 \
```

```
--set bind-priority:1 \
--set compare-priority:1 \
--set delete-priority:1 \
--set extended-priority:1 \
--set modify-priority:1 \
--set modifydn-priority:1 \
--set search-priority:1
```

For this use case, you will need four load balancing routes. Set two routes per load balancing workflow element (created in the previous step); for example, one route with priority 1 for all operations and the other route with priority 2 for all operations.

---

**Note:** The properties in the example above set the priority for failover load balancing. If you use proportional or saturation load balancing, the properties will differ. For more information on the setting different load balancing types, see Section 15.1.3.5, "Modifying Load Balancing Properties."

---

6. Set up distribution by creating a distribution workflow element.

```
$ dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file \
create-workflow-element \
--element-name distrib-we \
--type distribution \
--set base-dn:dc=example,dc=com \
--set enabled:true
```

For this use case, you will need only one distribution workflow element, which will point to the distribution algorithm.

7. Set the distribution algorithm.

```
$ dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file \
create-distribution-algorithm \
--element-name distrib-we \
--type numeric \
--set distribution-attribute:uid
```

The type of distribution algorithm can be `capacity`, `numeric`, `lexico`, or `dnpattern`. The boundaries are defined when you create the distribution partitions, in the next step.

8. Define the distribution partitions.

```
$ dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file \
create-distribution-partition \
--element-name distrib-we \
--partition-name distrib-part1\
--type numeric \
--set lower-bound:0 \
--set upper-bound:1000 \
--set partition-id:1 \
--set workflow-element:LB-we1
```

For this use case, you will need to create two partitions. Make sure that the partition ID and the partition name are unique for each workflow element, and that each partition uses a different load balancing workflow element. You must

specify the same type when defining the routes as you did when defining the load balancing algorithm.

> **Note:** The upper boundary indicated is exclusive. This means that if you indicate 1000 as the upper boundary, the partition will only include values from 0 to 999, inclusive.

**To create a global index**

Depending on the type of distribution algorithm defined, you need to create a global index. If you created a capacity algorithm, then you must create a global index.

For lexico, numeric, and dnpattern, a global index is optional.

Perform the following steps to create a global index.

**a.** Create a global index catalog:

```
$ gicadm -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file \
create-catalog \
--catalogName gi-catalog
```

**b.** Add a global index which indexes the dn attribute to the catalog.

```
$ gicadm -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file \
add-index \
--catalogName gi-catalog \
--attributeName dn
```

**c.** Associate the global index catalog to the distribution.

```
$ gicadm -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file \
associate \
--catalogName gi-catalog \
--distributionWorkflowElement distrib-we
```

**9.** Create a workflow.

This workflow associates the distribution workflow element with the base DN.

```
$ dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file \
create-workflow \
--workflow-name workflow \
--set enabled:true \
--set base-dn:dc=example,dc=com \
--set workflow-element:distrib-we
```

**10.** Create the network group.

The network group handles all the requests between the client and the proxy.

```
$ dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file \
create-network-group \
--group-name network-group1 \
--set enabled:true \
--set workflow:workflow \
--set priority:1
```

## 16.4 Configuring Failover Between Data Centers

Use the following commands to set up a failover deployment between two data centers, as presented in Section 3.4, "Configuration 3: Failover Between Data Centers."

```
#Create a proxy LDAP extension for each remote LDAP server
dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-extension \
  --type ldap-server \
  --extension-name proxy-extension1 \
  --set enabled:true \
  --set remote-ldap-server-address:DS1_hostname \
  --set remote-ldap-server-port:3189

dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-extension \
  --type ldap-server \
  --extension-name proxy-extension2 \
  --set enabled:true \
  --set remote-ldap-server-address:DS2_hostname \
  --set remote-ldap-server-port:3289

dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-extension \
  --type ldap-server \
  --extension-name proxy-extension3 \
  --set enabled:true \
  --set remote-ldap-server-address:DS3_hostname \
  --set remote-ldap-server-port:3389

dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-extension \
  --type ldap-server \
  --extension-name proxy-extension4 \
  --set enabled:true \
  --set remote-ldap-server-address:DS4_hostname \
  --set remote-ldap-server-port:3489

#Create a proxy workflow element for each LDAP server extension
dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-workflow-element \
  --element-name proxy-we1 \
  --type proxy-ldap \
  --set enabled:true \
  --set client-cred-mode:use-client-identity \
  --set ldap-server-extension:proxy-extension1

dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-workflow-element \
  --element-name proxy-we2 \
  --type proxy-ldap \
  --set enabled:true \
  --set client-cred-mode:use-client-identity \
  --set ldap-server-extension:proxy-extension2

dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-workflow-element \
  --element-name proxy-we3 \
  --type proxy-ldap \
  --set enabled:true \
  --set client-cred-mode:use-client-identity \
```

```
  --set ldap-server-extension:proxy-extension3

dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-workflow-element \
  --element-name proxy-we4 \
  --type proxy-ldap \
  --set enabled:true \
  --set client-cred-mode:use-client-identity \
  --set ldap-server-extension:proxy-extension4

# Create a load balancing workflow element for each data center
dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-workflow-element \
  --element-name LB-we1 \
  --type load-balancing \
  --set enabled:true

dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-workflow-element \
  --element-name LB-we2 \
  --type load-balancing \
  --set enabled:true

# Define the load balancing algorithm for each data center
dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-load-balancing-algorithm \
  --element-name LB-we1 \
  --type proportional

dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-load-balancing-algorithm \
  --element-name LB-we2 \
  --type proportional

# Define the load balancing routes for each proxy
dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-load-balancing-route \
  --element-name LB-we1 \
  --route-name LB-route1 \
  --type proportional \
  --set workflow-element:proxy-we1

dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-load-balancing-route \
  --element-name LB-we1 \
  --route-name LB-route2 \
  --type proportional \
  --set workflow-element:proxy-we2

dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-load-balancing-route \
  --element-name LB-we2 \
  --route-name LB-route3 \
  --type proportional \
  --set workflow-element:proxy-we3

dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-load-balancing-route \
  --element-name LB-we2 \
  --route-name LB-route4 \
```

```
  --type proportional \
  --set workflow-element:proxy-we4

# Set failover between the two data centers
dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-workflow-element \
  --element-name FO-we \
  --type load-balancing \
  --set enabled:true

dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-load-balancing-algorithm \
  --element-name FO-we \
  --type failover

dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-load-balancing-route \
  --element-name FO-we \
  --route-name FO-route1 \
  --type failover \
  --set workflow-element:LB-we1 \
  --set add-priority:1 \
--set bind-priority:1 \
--set compare-priority:1 \
--set delete-priority:1 \
--set extended-priority:1 \
--set modify-priority:1 \
--set modifydn-priority:1 \
--set search-priority:1 \

dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-load-balancing-route \
  --element-name FO-we \
  --route-name FO-route2 \
  --type failover \
  --set workflow-element:LB-we2 \
  --set add-priority:2 \
--set bind-priority:2 \
--set compare-priority:2 \
--set delete-priority:2 \
--set extended-priority:2 \
--set modify-priority:2 \
--set modifydn-priority:2 \
--set search-priority:2 \

# Create workflow
dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-workflow \
  --workflow-name FO-workflow \
  --set enabled:true \
  --set base-dn:dc=example,dc=com \
  --set workflow-element:FO-we

# Create network group
dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-network-group \
  --group-name network-group1 \
  --set enabled:true \
  --set workflow:FO-workflow \
  --set priority:1
```

## 16.5 Configuring Distribution with Failover Between Data Centers

Use the following commands to set up a failover deployment between two data centers, as presented in Section 3.6, "Configuration 5: Distribution with Failover Between Data Centers."

```
#Create the first failover route
#Create a proxy LDAP extension for each remote LDAP server
dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-extension \
  --type ldap-server \
  --extension-name proxy-extension-1a \
  --set enabled:true \
  --set remote-ldap-server-address:DS1a_hostname \
  --set remote-ldap-server-port:3189

dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-extension \
  --type ldap-server \
  --extension-name proxy-extension-2a \
  --set enabled:true \
  --set remote-ldap-server-address:DS2a_hostname \
  --set remote-ldap-server-port:3289

dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-extension \
  --type ldap-server \
  --extension-name proxy-extension-1b \
  --set enabled:true \
  --set remote-ldap-server-address:DS1b_hostname \
  --set remote-ldap-server-port:3389

dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-extension \
  --type ldap-server \
  --extension-name proxy-extension-2b \
  --set enabled:true \
  --set remote-ldap-server-address:DS2b_hostname \
  --set remote-ldap-server-port:3489

#Create a proxy workflow element for each LDAP server extension
dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-workflow-element \
  --element-name proxy-we-1a \
  --type proxy-ldap \
  --set enabled:true \
  --set client-cred-mode:use-client-identity \
  --set ldap-server-extension:proxy-extension-1a

dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-workflow-element \
  --element-name proxy-we-2a \
  --type proxy-ldap \
  --set enabled:true \
  --set client-cred-mode:use-client-identity \
  --set ldap-server-extension:proxy-extension-2a

dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
```

```
create-workflow-element \
  --element-name proxy-we-1b \
  --type proxy-ldap \
  --set enabled:true \
  --set client-cred-mode:use-client-identity \
  --set ldap-server-extension:proxy-extension-1b

dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-workflow-element \
  --element-name proxy-we-2b \
  --type proxy-ldap \
  --set enabled:true \
  --set client-cred-mode:use-client-identity \
  --set ldap-server-extension:proxy-extension-2b

# Create a load balancing workflow element for each data center
dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-workflow-element \
  --element-name LB-we-1a \
  --type load-balancing \
  --set enabled:true

dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-workflow-element \
  --element-name LB-we-1b \
  --type load-balancing \
  --set enabled:true

# Define the load balancing algorithm for each data center
dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-load-balancing-algorithm \
  --element-name LB-we-1a \
  --type proportional

dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-load-balancing-algorithm \
  --element-name LB-we-1b \
  --type proportional

# Define the load balancing routes for each proxy
dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-load-balancing-route \
  --element-name LB-we-1a \
  --route-name LB-route-1a \
  --type proportional \
  --set workflow-element:proxy-we-1a

dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-load-balancing-route \
  --element-name LB-we-1a \
  --route-name LB-route-2a \
  --type proportional \
  --set workflow-element:proxy-we-2a

dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-load-balancing-route \
  --element-name LB-we-1b \
  --route-name LB-route-1b \
  --type proportional \
  --set workflow-element:proxy-we-1b
```

```
dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-load-balancing-route \
  --element-name LB-we-1b \
  --route-name LB-route-2b \
  --type proportional \
  --set workflow-element:proxy-we-2b

# Set failover between the two data centers
dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-workflow-element \
  --element-name FO-we1 \
  --type load-balancing \
  --set enabled:true

dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-load-balancing-algorithm \
  --element-name FO-we1 \
  --type failover

dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-load-balancing-route \
  --element-name FO-we1 \
  --route-name FO-route-1a \
  --type failover \
  --set workflow-element:LB-we-1a \
  --set add-priority:1 \
  --set bind-priority:1 \
  --set compare-priority:1 \
  --set delete-priority:1 \
  --set extended-priority:1 \
  --set modify-priority:1 \
  --set modifydn-priority:1 \
  --set search-priority:1

dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-load-balancing-route \
  --element-name FO-we1 \
  --route-name FO-route-1b \
  --type failover \
  --set workflow-element:LB-we-1b \
  --set add-priority:2 \
  --set bind-priority:2 \
  --set compare-priority:2 \
  --set delete-priority:2 \
  --set extended-priority:2 \
  --set modify-priority:2 \
  --set modifydn-priority:2 \
  --set search-priority:2

#Create the second failover route
#Create a proxy LDAP extension for each remote LDAP server
dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-extension \
  --type ldap-server \
  --extension-name proxy-extension-3a \
  --set enabled:true \
  --set remote-ldap-server-address:DS3a_hostname \
  --set remote-ldap-server-port:3189
```

```
dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-extension \
  --type ldap-server \
  --extension-name proxy-extension-4a \
  --set enabled:true \
  --set remote-ldap-server-address:DS4a_hostname \
  --set remote-ldap-server-port:3289

dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-extension \
  --type ldap-server \
  --extension-name proxy-extension-3b \
  --set enabled:true \
  --set remote-ldap-server-address:DS3b_hostname \
  --set remote-ldap-server-port:3389

dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-extension \
  --type ldap-server \
  --extension-name proxy-extension-4b \
  --set enabled:true \
  --set remote-ldap-server-address:DS4b_hostname \
  --set remote-ldap-server-port:3489

#Create a proxy workflow element for each LDAP server extension
dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-workflow-element \
  --element-name proxy-we-3a \
  --type proxy-ldap \
  --set enabled:true \
  --set client-cred-mode:use-client-identity \
  --set ldap-server-extension:proxy-extension-3a

dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-workflow-element \
  --element-name proxy-we-4a \
  --type proxy-ldap \
  --set enabled:true \
  --set client-cred-mode:use-client-identity \
  --set ldap-server-extension:proxy-extension-4a

dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-workflow-element \
  --element-name proxy-we-3b \
  --type proxy-ldap \
  --set enabled:true \
  --set client-cred-mode:use-client-identity \
  --set ldap-server-extension:proxy-extension-3b

dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-workflow-element \
  --element-name proxy-we-4b \
  --type proxy-ldap \
  --set enabled:true \
  --set client-cred-mode:use-client-identity \
  --set ldap-server-extension:proxy-extension-4b

# Create a load balancing workflow element for each data center
dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-workflow-element \
```

```
  --element-name LB-we-2a \
  --type load-balancing \
  --set enabled:true

dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-workflow-element \
  --element-name LB-we-2b \
  --type load-balancing \
  --set enabled:true

# Define the load balancing algorithm for each data center
dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-load-balancing-algorithm \
  --element-name LB-we-2a \
  --type proportional

dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-load-balancing-algorithm \
  --element-name LB-we-2b \
  --type proportional

# Define the load balancing routes for each proxy
dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-load-balancing-route \
  --element-name LB-we-2a \
  --route-name LB-route-3a \
  --type proportional \
  --set workflow-element:proxy-we-3a

dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-load-balancing-route \
  --element-name LB-we-2a \
  --route-name LB-route-4a \
  --type proportional \
  --set workflow-element:proxy-we-4a

dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-load-balancing-route \
  --element-name LB-we-2b \
  --route-name LB-route-3b \
  --type proportional \
  --set workflow-element:proxy-we-3b

dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-load-balancing-route \
  --element-name LB-we-2b \
  --route-name LB-route-4b \
  --type proportional \
  --set workflow-element:proxy-we-4b

# Set failover between the two data centers
dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-workflow-element \
  --element-name FO-we2 \
  --type load-balancing \
  --set enabled:true

dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-load-balancing-algorithm \
  --element-name FO-we2 \
```

```
  --type failover

dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-load-balancing-route \
  --element-name FO-we2 \
  --route-name FO-route-2a \
  --type failover \
  --set workflow-element:LB-we-2a \
  --set add-priority:1 \
  --set bind-priority:1 \
  --set compare-priority:1 \
  --set delete-priority:1 \
  --set extended-priority:1 \
  --set modify-priority:1 \
  --set modifydn-priority:1 \
  --set search-priority:1

dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-load-balancing-route \
  --element-name FO-we2 \
  --route-name FO-route-2b \
  --type failover \
  --set workflow-element:LB-we-2b \
  --set add-priority:2 \
  --set bind-priority:2 \
  --set compare-priority:2 \
  --set delete-priority:2 \
  --set extended-priority:2 \
  --set modify-priority:2 \
  --set modifydn-priority:2 \
  --set search-priority:2

# Create distribution to the two failover routes
dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-workflow-element \
  --element-name distrib-we \
  --type distribution \
  --set base-dn:dc=example,dc=com \
  --set enabled:true

dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-distribution-algorithm \
  --element-name distrib-we \
  --type numeric \
  --set distribution-attribute:uid

dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-distribution-partition \
  --element-name distrib-we \
  --partition-name distrib-part1\
  --type numeric \
  --set lower-bound:0 \
  --set upper-bound:1000 \
  --set partition-id:1 \
  --set workflow-element:FO-we1

dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-distribution-partition \
  --element-name distrib-we \
  --partition-name distrib-part2\
```

```
  --type numeric \
  --set lower-bound:1000 \
  --set upper-bound:2000 \
  --set partition-id:2 \
  --set workflow-element:FO-we2


# Create workflow
dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-workflow \
  --workflow-name Distrib-workflow \
  --set enabled:true \
  --set base-dn:dc=example,dc=com \
  --set workflow-element:distrib-we

# Create network group
dsconfig -p 4444 -h localhost -D"cn=Directory Manager" -j pwd-file -X -n \
create-network-group \
  --group-name network-group1 \
  --set enabled:true \
  --set workflow:Distrib-workflow \
  --set priority:1
```

# 17

# Managing Directory Data

This chapter describes how to add, modify, remove, and search data in the directory server. The chapter includes information about how to make searches more efficient, by indexing data, how to ensure that entries are unique, and how to use advanced data features such as virtual attributes.

This chapter covers the following topics:



## 17.1 Importing and Exporting Data

The directory server provides several mechanisms to move data into and out of a specific back end. This chapter outlines the various options and then describes the import and export mechanisms in more detail.

This section covers the following topics:

- Section 17.1.4, "Creating MakeLDIF Template Files"

---

**Note:** When you import user entries, note that Oracle Unified Directory cannot verify that pre-encrypted passwords match the password policy. Pre-encrypted passwords are therefore rejected with the following error:

```
LDAP: error code 53 - Pre-encoded passwords are not
allowed for the password attribute userPassword.
```

To allow pre-encrypted passwords when you import user entries using `ldapmodify` or `import-ldif`, change the default password policy by setting the advanced property `allow-pre-encoded-passwords` to `true`. For more information, see Section 24.2.2, "To Modify the Default Password Policy".

---

### 17.1.1 Populating a Stand-Alone Directory Server With Data

To populate a stand-alone directory server with data, use one of the following methods:

- Import the data from an LDAP Data Interchange Format (LDIF) file while you are setting up the server, either by using the `setup` utility in GUI mode or by using the `setup` utility in interactive command-line mode. This is the most convenient method of initializing a stand-alone server or the first server in a replicated topology.
- Start with an empty suffix and add entries by using the `ldapmodify` command, for example:

  ```
  $ ldapmodify -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
    -a -f /usr/local/add_entry.ldif
  ```

- Import data from an LDIF file, using the `import-ldif` command. For example:

  ```
  $ import-ldif -b dc=example,dc=com -n userRoot -l /var/tmp/Example.ldif
  ```

  This method is much more efficient for the addition of bulk entries. The `import-ldif` command imports data from an LDIF file either by replacing any existing data in the suffix or by appending data to a base DN. Similarly, the `export-ldif` command exports entries from a database to an LDIF file, which can then be imported to another server. Both tools support file compression, SASL extension, and client/server authentication using SSL and startTLS.

- Copy the binary database from another server. This method is also called *binary copy*.

  ```
  $ cp instance-path/db/example.db destination-path/db
  ```

- Restore the database from a backup using the `restore` command, for example:

  ```
  $ restore -d /home/backup/userRoot
  ```

---

**Note:** Performing a binary database copy or restoring a database from a backup requires the source server and the destination server to have the same database remote LDAP structures and indexes.

---

## 17.1.2 Importing Data Using `import-ldif`

The `import-ldif` command is used to populate a directory server back end with data read from an LDIF file or with data generated based on a Section 17.1.4, "Creating MakeLDIF Template Files." In most cases, `import-ldif` is significantly faster than adding entries using `ldapmodify`.

The `import-ldif` command supports both LDIF files and compressed files (`.zip`).

Note the following aspects of an import operation:

- A complete import to an entire Oracle Berkeley DB Java Edition (JE) back end will have better performance than a partial import to a branch of the JE back end. All imported LDIF files must use UTF-8 character-set encoding.
- Importing *suffixes* is a resource-intensive operation. If you import LDIF files that include a large number of suffixes, your system might have insufficient heap to complete the import operation. Before importing such LDIF files, you should therefore increase the heap as much as possible. For more information, see Chapter 30, "Tuning Performance" and Section 17.2, "Importing Large Data Sets."
- You do not need root privileges to import an LDIF file, but you must authenticate as a user with root permissions, such as `cn=Directory Manager`.

The following sections describe how to import data using the `import-ldif` command:

- Section 17.1.2.1, "`import-ldif` Operation Modes"
- Section 17.1.2.2, "To Import Data in Offline Mode"
- Section 17.1.2.3, "To Replace Existing Data During an Offline Import"
- Section 17.1.2.4, "To Append Imported Data to Existing Data"
- Section 17.1.2.5, "To Import Fractional Files"
- Section 17.1.2.6, "To Import Fractional Files by Using Filters"
- Section 17.1.2.7, "To Include or Exclude Attributes During Import"
- Section 17.1.2.8, "To Import a Compressed LDIF File"
- Section 17.1.2.9, "To Record Rejected or Skipped Entries During Import"
- Section 17.1.2.10, "To Import Data From a MakeLDIF Template"
- Section 17.1.2.11, "To Run an Import in Online Mode"
- Section 17.1.2.12, "To Schedule an Import"

### 17.1.2.1 `import-ldif` Operation Modes

The `import-ldif` command has two modes of operation: online and offline.

- **Online mode**. In online mode, `import-ldif` contacts a running directory server instance and registers an import task. The command accesses the task back end over SSL via the administration connector. For more information, see Section 14.3, "Managing Administration Traffic to the Server." Online mode runs automatically when any connection options (such as `--hostname`, `--port`, `--bindDN`, and `--bindPasswordFile`) are specified.

  Note that, even for an online import, the backend is unavailable during the import. In a replicated topology, the overall service remains available through the referral on update feature. For more information, see Section 17.13.1, "Referrals in a Replicated Topology".

In general, if you expect to do online imports, you should increase the heap when you start the server. For more information, see Chapter 30, "Tuning Performance."

- **Offline mode**. When no connection options are specified, the command runs in offline mode. In offline mode, `import-ldif` accesses the database directly rather than through a directory server instance. In this case, the directory server must be stopped.

#### 17.1.2.2 To Import Data in Offline Mode

This procedure imports a remote LDAP database with new entries specified in an import LDIF file. The command runs in *offline* mode, which requires the server to be shut down prior to import.

1. Stop the server if it is running.

```
$ stop-ds
```

2. Import the LDIF file, as shown in the following example:

```
$ import-ldif -b dc=example,dc=com -n userRoot -l Example.ldif
```

This command specifies the base DN for the branch of the data that should be included in the import (`-b`), the back-end ID into which the data is imported (`-n`), and the LDIF file used for the import (`-l`).

#### 17.1.2.3 To Replace Existing Data During an Offline Import

The following procedure replaces an existing back-end with new entries specified in an import file.

1. Stop the server if it is running.

```
$ stop-ds
```

2. Import the LDIF file, replacing the existing data. For example:

```
$ import-ldif --includeBranch dc=example,dc=com --backendID userRoot \
  --replaceExisting --ldifFile Example.ldif
```

#### 17.1.2.4 To Append Imported Data to Existing Data

The following procedure appends the entries in an import file to the existing entries in the back end.

1. Stop the server if it is running.

```
$ stop-ds
```

2. Import the LDIF file, appending the new data to the existing data. For example:

```
$ import-ldif --backendID userRoot --append --ldifFile new.ldif
```

#### 17.1.2.5 To Import Fractional Files

The `import-ldif` command provides options to import a portion of an import file by specifying the base DN to include or exclude during the process.

This example imports all entries below the base DN, `dc=example,dc=com`, and excludes all entries below `ou=People,dc=example,dc=com`.

1. Stop the server if it is running.

```
$ stop-ds
```

2. Import a portion of the LDIF file. For example:

```
$ import-ldif --includeBranch dc=example,dc=com \
  --excludeBranch ou=People,dc=example,dc=com --backendID userRoot \
  --replaceExisting --ldifFile Example.ldif
```

#### 17.1.2.6 To Import Fractional Files by Using Filters

The `import-ldif` command provides options to import part of an import file by using filters for data inclusion or exclusion. Make sure that you fully understand how this mechanism works before you use it.

In this example, the contents of an LDIF file are imported, except those entries that match the search filter `l=Auckland` (that is, `location=Auckland`).

The `--includeFilter` option works in a similar manner to `--excludeFilter`, except that it includes all entries that match the search filter during import

1. Stop the server if it is running.

```
$ stop-ds
```

2. Import a portion of the file by using an exclude filter. For example:

```
$ import-ldif --excludeFilter "(l=Auckland)" --backendID userRoot \
  --replaceExisting --ldifFile Example.ldif
```

#### 17.1.2.7 To Include or Exclude Attributes During Import

The `import-ldif` command provides options to include and exclude attributes during import by using the `--includeAttribute` and `--excludeAttribute` options, respectively. Make sure that you fully understand how this mechanism works before you use it.

1. Stop the server if it is running.

```
$ stop-ds
```

2. View the entries of the import file before you start the import.

   The directory server provides useful utilities to search, modify, compare, or delete import files without connecting to the server. You can use the `ldifsearch` command to display an entry in your import file. For example, to display the entry for Sam Carter, use the following command:

```
$ ldifsearch -b dc=example,dc=com --ldifFile Example.ldif "(cn=Sam Carter)"
dn: uid=scarter,ou=People,dc=example,dc=com
objectClass: person
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: top
givenname: Sam
uid: scarter
cn: Sam Carter
telephonenumber: +1 408 555 4798
sn: Carter
userpassword: sprain
roomnumber: 4612
mail: scarter@example.com
l: Sunnyvale
ou: Accounting
ou: People
facsimiletelephonenumber: +1 408 555 9751
```

In this entry, notice the presence of the `roomnumber` attribute below the `telephonenumber` attribute.

3. Import the file, excluding the `roomnumber` attribute for all entries.

```
$ import-ldif --excludeAttribute "roomnumber" --backendID userRoot \
  --replaceExisting --ldifFile Example.ldif
```

4. Start the server.

```
$ start-ds
```

5. Perform an `ldapsearch` to verify the import.

   The following example shows that the `roomnumber` attribute is now absent from Sam Carter's entry.

```
$ ldapsearch --port 1389 --baseDN dc=example,dc=com --bindDN "cn=Directory
Manager" \
  --bindPassword password "(cn=Sam Carter)"
dn: uid=scarter,ou=People,dc=example,dc=com \
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: top
givenName: Sam
uid: scarter
cn: Sam Carter
sn: Carter
telephoneNumber: +1 408 555 4798
ou: Accounting
ou: People
l: Sunnyvale
mail: scarter@example.com
facsimileTelephoneNumber: +1 408 555 9751
```

#### 17.1.2.8 To Import a Compressed LDIF File

The `import-ldif` utility supports compressed LDIF files.

1. Stop the server if it is running.

```
$ stop-ds
```

2. Import the compressed LDIF file.

```
$ import-ldif --includeBranch dc=example,dc=com
  --excludeBranch "ou=People,dc=example,dc=com" --ldifFile Example.ldif \
  --backendID userRoot --replaceExisting --isCompressed
```

#### 17.1.2.9 To Record Rejected or Skipped Entries During Import

The `import-ldif` command provides a means to write to an output file for any entries that are rejected or skipped during the import process. This enables easy debugging of an LDIF file. Rejected entries occur when the directory server rejects the added entries due to schema violations. Skipped entries occur when entries cannot be placed under the specified base DN.

1. Stop the server if it is running.

```
$ stop-ds
```

2. Import the file, using the `--rejectFile` and `--skipFile` options.

You can also use the `--overWrite` option to replace any previous items in the two files. Without the option, the directory server appends new rejected and skipped entries to the existing files.

```
$ import-ldif --backendID userRoot --append --ldifFile new.ldif
  --overwrite --rejectFile rejected.ldif --skipFile skipped.ldif
```

3. View the contents of the `rejectFile` and `skipFile` to determine which entries were rejected or skipped during the import. For example:

```
$ more rejected.ldif
# Entry ou=Contractors,dc=example,dc=com read from LDIF starting at line 1
is not valid because it violates the server's schema configuration:
Entry ou=Contractors,dc=example,dc=com violates the Directory Server schema
configuration because it includes attribute changeType which is not allowed.
changetype: add objectclasses defined in that entry objectclass: top
objectclass: organizationalUnit ou: Contractors ou: Product Testing
ou: Product Dev ou: Accounting ...

$ more skipped.ldif
# Skipping entry ou=People,dc=example,dc=com because the DN is not one that
should be
  included based on the include and exclude branches objectclass: top
  objectclass: organizationalunit ou: People
  aci: (target ="ldap:///ou=People,dc=example,dc=com")(targetattr
="userpassword ||
  telephonenumber || facsimiletelephonenumber")(version 3.0;acl "Allow self
entry
  modification"; allow (write)(userdn = "ldap:///self");)
  aci: (target ="ldap:///ou=People,dc=example,dc=com")(targetattr h3.="cn || sn
||
  uid") (targetfilter ="(ou=Accounting)")(version 3.0;acl "Accounting Managers
Group
  Permissions"; allow (write)
  (groupdn = "ldap:///cn=Accounting Managers,ou=groups,dc=example,dc=com");)
  aci: (target ="ldap:///ou=People,dc=example,dc=com")(targetattr h3.="cn || sn
||
  uid") (targetfilter ="(ou=Human Resources)")(version 3.0;acl "HR Group
Permissions";
  allow write)(groupdn = "ldap:///cn=HR
Managers,ou=groups,dc=example,dc=com");) aci:
  (target ="ldap:///ou=People,dc=example,dc=com")(targetattr h3.="cn ||sn ||
uid")
  (targetfilter ="(ou=Product Testing)")(version 3.0;acl "QA Group
Permissions"; allow
  (write)(groupdn = "ldap:///cn=QA Managers,ou=groups,dc=example,dc=com");)
  aci: (target ="ldap:///ou=People,dc=example,dc=com")(targetattr h3.="cn || sn
||
  uid") (targetfilter ="(ou=Product Development)")(version 3.0;acl "Engineering
Group
  Permissions"; allow (write)(groupdn =
  "ldap:///cn=PD Managers,ou=groups,dc=example,dc=com");) ...
```

### 17.1.2.10 To Import Data From a MakeLDIF Template

The directory server includes the Java utility, `makeLDIF`, that can be used to generate sample data for import. The `makeLDIF` utility requires a template file. You can create your own template file, or you can use the template file located in `INSTANCE_DIR/OUD/config/MakeLDIF/example.template`, editing it as

required. For more information, see Section 17.1.4, "Creating MakeLDIF Template Files" and Appendix A.3.11, "make-ldif."

1. Stop the server if it is running.

```
$ stop-ds
```

2. Import the data, using a template file.

   The sample template generates 10,003 sample entries in the specified back end.

```
$ import-ldif --backendID userRoot --templateFile example.template \
  --randomSeed 0
```

#### 17.1.2.11 To Run an Import in Online Mode

The `import-ldif` utility can also be run with the server online. In online mode, the command accesses the task back end over SSL via the administration connector. To run the command in online mode you must specify the relevant connection options, including how the SSL certificate will be trusted. This example uses the `-X` option to trust all certificates. For more information, see Section 14.3, "Managing Administration Traffic to the Server."

Run the `import-ldif` command with the appropriate connection options.

```
$ import-ldif -h localhost -port 4444 -D "cn=Directory Manager" -j pwd-file -X \
  -l /ldif-files/example.ldif
```

#### 17.1.2.12 To Schedule an Import

The `import-ldif` utility provides a `--start` option for scheduling the import at some future date. You can view this scheduled task by using the `manage-tasks` utility. The command accesses the task back end over SSL via the administration connector. To schedule an import task, you must specify the relevant connection options, including how the SSL certificate will be trusted. This example uses the `-X` option to trust all certificates.

Run the `import-ldif` command with the `--start` option.

```
$ import-ldif -h localhost -port 4444 -D "cn=Directory Manager" -j pwd-file -X \
  -l /ldif-files/example.ldif --start 20080124121500
```

For more information, see Section 14.4, "Configuring Commands As Tasks."

### 17.1.3 Exporting Data Using `export-ldif`

The `export-ldif` command is used to export data from a directory server back end. The command is useful for the following tasks:

- Backing up directory data
- Exporting data to another application
- Repopulating a database after a change to the directory topology
- Reinitializing master servers in a replicated topology

> **Note:** The `export-ldif` command cannot be used to export data from the following back ends: `monitor`, `ads-truststore`, `backup`, and `config-file-handler`.

The following sections describe how to export data using the `export-ldif` command:

- Section 17.1.3.1, "`export-ldif` Operation Modes"
- Section 17.1.3.2, "To Export Data to LDIF"
- Section 17.1.3.3, "To Export Partial Data"
- Section 17.1.3.4, "To Export Part of a Back End by Using Filters"
- Section 17.1.3.5, "To Include or Exclude Attributes During Export"
- Section 17.1.3.6, "To Export to LDIF and Then Compress the File"
- Section 17.1.3.7, "To Run an Export in Online Mode"
- Section 17.1.3.8, "To Schedule an Export"

#### 17.1.3.1 `export-ldif` Operation Modes

The `export-ldif` command has two modes of operation: online and offline.

- **Online mode**. In online mode, `export-ldif` contacts a running directory server instance and registers an export task. This mode runs automatically when the LDAP connection options (`--hostname`, `--port`, `--bindDN`, and `--bindPasswordFile`) are used. The command accesses the task back end over SSL via the administration connector. For more information, see Section 14.3, "Managing Administration Traffic to the Server."
- **Offline mode**. When no connection options are specified, the command runs in offline mode. In offline mode, `export-ldif` accesses the database directly rather than through a directory server instance. In this case, the directory server must be stopped.

For more information, see Appendix A.3.5, "export-ldif".

#### 17.1.3.2 To Export Data to LDIF

1. Stop the server if it is running.

   ```
   $ stop-ds
   ```

2. Export the back end to a specified LDIF file.

   ```
   $ export-ldif --includeBranch "dc=example,dc=com" --backendID userRoot \
     --ldifFile example.ldif
   ```

#### 17.1.3.3 To Export Partial Data

The `export-ldif` command provides options to export a part of a back end by specifying the base DN and its children for inclusion or exclusion during processing.

1. Stop the server if it is running.

   ```
   $ stop-ds
   ```

2. Export a portion of the back end.

   In this example, only the entries under `ou=People,dc=example,dc=com` are exported.

   ```
   $ export-ldif --includeBranch ou=People,dc=example,dc=com --backendID userRoot
   \
     --ldifFile example-people.ldif
   ```

3. Use the `ldifsearch` command to verify the exported file.

   The `ldifsearch` command verifies entries in an LDIF file without connecting to the directory server. You can use it in a manner similar to the `ldapsearch` command. For example:

```
$ ldifsearch -b dc=example,dc=com --ldifFile export.ldif "(objectclass=*)"
dn: ou=People,dc=example,dc=com
objectClass: organizationalunit
objectClass: top
ou: People
dn: uid=scarter,ou=People,dc=example,dc=com
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: top
givenName: Sam
uid: scarter
cn: Sam Carter
sn: Carter
telephoneNumber: +1 408 555 4798
userPassword: {SSHA}Ocpp2P4sImz2MziL69AUG9+khdIhFpmU4B5mvA==
roomNumber: 4612
ou: Accounting
ou: People
l: Sunnyvale
mail: scarter@example.com
facsimileTelephoneNumber: +1 408 555 9751 ...
```

#### 17.1.3.4 To Export Part of a Back End by Using Filters

The `export-ldif` command provides options to export part of a back end by using a search filter. The directory server includes or excludes all entries that match the filter. Make sure that you fully understand how this mechanism works before you use it.

In this example, only those entries that match the search filter `l=Cupertino` (that is, `location=Cupertino`) are exported. The `--excludeFilter` option works in a similar manner to `--includeFilter`, except that it excludes all entries that match the filter during export.

1. Stop the server if it is running.

```
$ stop-ds
```

2. Export a portion of the back end by using the `--includeFilter` option.

```
$ export-ldif --includeFilter "(l=Cupertino)" --backendID userRoot \
  --ldifFile export.ldif
```

#### 17.1.3.5 To Include or Exclude Attributes During Export

The `export-ldif` utility provides options to include and exclude attributes during export by using the `--includeAttribute` and `--excludeAttribute` options, respectively. Make sure that you fully understand how this mechanism works before you use it.

1. With the server running, view a sample entry, by using the `ldapsearch` command. For example:

```
$ ldapsearch --baseDN dc=example,dc=com "(cn=Sam Carter)"
dn: uid=scarter,ou=People,dc=example,dc=com
objectClass: person
objectClass: inetOrgPerson
```

```
objectClass: organizationalPerson
objectClass: top
givenname: Sam
uid: scarter
cn: Sam Carter
telephonenumber: +1 408 555 4798
sn: Carter
userpassword: sprain
roomnumber: 4612
mail: scarter@example.com
l: Sunnyvale
ou: Accounting
ou: People
facsimiletelephonenumber: +1 408 555 9751
```

2. Stop the server.

```
$ stop-ds
```

3. Export the back end, using the `--includeAttribute` option to specify the attributes that should be included in the export.

   You can use the `--includeAttribute` option multiple times for each attribute that should be included. In this example, only the top level attributes are exported.

```
$ export-ldif --backendID userRoot --includeAttribute dn --includeAttribute dc
\
  --includeAttibute cn --includeAttribute sn --includeAttribute givenname \
  --includeAttribute objectclass --includeAttribute ou --includeAttribute uid \
  --ldifFile export.ldif
```

4. Use the `ldifsearch` command to verify the export file.

   If an error occurs, the server continues processing the command.

```
$ ldifsearch --baseDN dc=example,dc=com --ldifFile export.ldif
"(objectclass=*)"
dn: dc=example,dc=com
objectClass: domain
objectClass: top
dc: example
dn: ou=Groups,dc=example,dc=com
objectClass: organizationalunit
objectClass: top
ou: Groups
dn: cn=Directory Administrators,ou=Groups,dc=example,dc=com
objectClass: groupofuniquenames
objectClass: top
cn: Directory Administrators
ou: Groups
dn: ou=People,dc=example,dc=com
objectClass: organizationalunit
objectClass: top
ou: People ...
```

#### 17.1.3.6 To Export to LDIF and Then Compress the File

The `export-ldif` command allows you to compress the output LDIF file.

1. Stop the server if it is running.

```
$ stop-ds
```

2. Export to LDIF and then compress the file.

```
$ export-ldif --backendID userRoot --ldifFile export.ldif --compress
```

#### 17.1.3.7 To Run an Export in Online Mode

The `export-ldif` command can also be run with the server online. In online mode, the command accesses the task back end over SSL via the administration connector. For more information, see Section 14.3, "Managing Administration Traffic to the Server". To run the command in online mode you must specify the relevant connection options, including how the SSL certificate will be trusted. This example uses the `-X` option to trust all certificates.

Run the `export-ldif` command with the LDAP connection options. For example:

```
$ export-ldif -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X \
  --includeBranch "dc=example,dc=com" --backendID userRoot --ldifFile export.ldif
```

#### 17.1.3.8 To Schedule an Export

The `export-ldif` utility provides a `--start` option for scheduling the export at some future date. You can view this scheduled task by using the `manage-tasks` utility. The command accesses the task back end over SSL via the administration connector. For more information, see Section 14.3, "Managing Administration Traffic to the Server." To schedule an export task, you must specify the relevant connection options, including how the SSL certificate will be trusted. This example uses the `-X` option to trust all certificates.

The server must be running to schedule an export.

Run the `export-ldif` command with the `--start` option and the LDAP connection parameters.

The `--start` option takes as its value a date and time in the format `yyyymmddhhmmss`. For example:

```
$ export-ldif -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X \
  --includeBranch "dc=example,dc=com" --backendID userRoot \
  --ldifFile export.ldif --start 20080124121500
```

### 17.1.4 Creating MakeLDIF Template Files

The `make-ldif` command can use template files to define the way in which LDIF files are to be generated. This approach allows for flexibility without the need to alter any code to produce the desired result. The topics in this section describe how to use the `make-ldif` command to create customized LDIF files.

#### 17.1.4.1 The Template File Format

Template files can contain up to four sections, that must be provided in the following order:

1. Section 17.1.4.1.1, "Custom Tag Includes"
2. Section 17.1.4.1.2, "Global Replacement Variables"
3. Section 17.1.4.1.3, "Branch Definitions"
4. Section 17.1.4.1.4, "Template Definitions"

**17.1.4.1.1 Custom Tag Includes** Custom tag includes provide a mechanism for loading custom tags and making them available for use when processing `make-ldif` templates. This should be done using the `include` directive, as follows:

```
include com.example.opends.makeldif.MyCustomTag
```

The specified class must be in the class path, and it must be a subclass of the `org.opends.server.tools.makeldif.Tag` class. For information about developing custom tags, see Section 17.1.4.3, "Defining Custom Tags."

All of the standard replacement tags that are provided with `make-ldif` are automatically available for use and therefore do not require an explicit `include` directive.

**17.1.4.1.2 Global Replacement Variables** The first section that should be present in the template file is the section that defines the global replacement variables. Global replacement variables are used to define strings of text that can be referenced later in the template file and are automatically replaced as each line is read into memory (much like a C preprocessor replaces macros in code with their defined values). For example, the following replacement variable definition creates a global replacement variable named `suffix` with a value of `dc=example,dc=com`:

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

When a global replacement variable is defined, any case in which that variable name appears in square brackets (for example, `[suffix]`), causes the token to be replaced with the value that has been defined for that replacement variable.

When all the replacement variable definitions have been read (as signified by the first blank line following one or more replacement variable definitions), all remaining lines that are read from the template file are processed on a line-by-line basis. Any occurrences of a replacement variable name in square brackets are replaced with the value of that variable. Because that replacement is done as the template file is read into memory, replacement variables can occur in any point, including branch and template definitions, and even inside tags.

If there are global replacement variables defined in the template file, they must appear at the top of the file and there should not be any spaces between them. However, replacement variables are not required. If there are no replacement variables, the template file must start with the branch definitions.

**17.1.4.1.3 Branch Definitions** Branch definitions are used in make-ldif template files to define the basic structure to use for the generated LDIF. They specify the entry or entries that should appear at the top of the hierarchy, and the number and types of entries that should appear below them.

The most basic form of a branch definition is as follows:

```
branch: dc=example,dc=com
```

This example specifies that the following entry is to be created with a DN of `dc=example,dc=com`:

```
dn: dc=example,dc=com
objectClass: top
objectClass: domain
dc: example
```

The basic structure of the entry is defined by the RDN attribute of `dc` specified in the DN of the branch definition. The `make-ldif` command automatically associates the `dc` RDN attribute with the `domain` object class. The `make-ldif` command has similar definitions for other common RDN attributes in branch entries:

o
Creates an entry with the organization object class.

ou
Creates an entry with the organizationalUnit object class.

c
Creates an entry with the country object class.

You can also use any other kind of RDN attribute for a branch entry. For branch entries with an RDN attribute other than the ones specified above, the entry is created with the untypedObject and extensibleObject object classes.

The branch definition provided above does not cause any additional entries to be created below that branch entry. To do this, you must specify one or more subordinateTemplate lines. For example:

```
branch: ou=People,dc=example,dc=com
subordinateTemplate: person:100
```

This causes the ou=People,dc=example,dc=com entry to be created, and then 1000 other entries created below it modeled after the person template. The person template should be defined later in the template file. For more information, see Section 17.1.4.1.4, "Template Definitions."

Branch entries are not limited to just one subordinateTemplate definition. You can specify multiple subordinateTemplate definitions by including them on separate lines of the branch definition. The following example creates 1000 entries based on the person template and an additional 100 entries based on the certificatePerson template:

```
branch: ou=People,dc=example,dc=com
subordinateTemplate: person:10000
subordinateTemplate: certificatePerson:100
```

In all of the examples described previously, the branch entries themselves contain only the DN, the RDN attribute, and the object classes associated with the RDN attribute. You can include any other attributes in the branch entry by including them in the branch definition in the template file. For example, the branch definition:

```
branch: dc=example,dc=com
description: This is the description for dc=example,dc=com
```

creates the entry:

```
dn: dc=example,dc=com
objectClass: top
objectClass: domain
dc: example
description: This is the description for dc=example,dc=com
```

This additional text can be static, can contain any defined global replacement variables, or can contain a subset of the replacement tags that can be used in template definitions. For an overview of the tags available and information about which tags can be used in branch definitions, see Section 17.1.4.2.1, "Standard Replacement Tags."

#### 17.1.4.1.4 Template Definitions

The heart of the make-ldif template file structure is the set of template definitions. Templates define the structure of the entries that are generated. They specify the set of attributes that should be included in the entries and the types of values that those

attributes should contain. The specification of values is handled through tags that are parsed by `make-ldif` and replaced with the appropriate values for those tags.

A sample template definition might look as follows:

```
template: person
rdnAttr: uid
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
givenName: <first>
sn: <last>
cn: {givenName} {sn}
initials: {givenName:1}<random:chars:ABCDEFGHIJKLMNOPQRSTUVWXYZ:1>{sn:1}
employeeNumber: <sequential:0>
uid: user.{employeeNumber}
mail: {uid}@[maildomain]
userPassword: password
telephoneNumber: <random:telephone>
homePhone: <random:telephone>
pager: <random:telephone>
mobile: <random:telephone>
street: <random:numeric:5> <file:streets> Street
l: <file:cities>
st: <file:states>
postalCode: <random:numeric:5>
postalAddress: {cn}${street}${l}, {st}  {postalCode}
description: This is the description for {cn}.
```

This example illustrates some of the flexibility that `make-ldif` provides when generating LDIF data. The tags that can be included in a template definition are described in the topics that follow (see Section 17.1.4.2.1, "Standard Replacement Tags" and Section 17.1.4.2.2, "Attribute Value Reference Tags").

At the top of the template definition are two lines that provide information about the template itself and are not included in the entries created from this template. The first line specifies the name of the template. This is the name that is referenced in the `subordinateTemplate` lines of the branch definition. The second line specifies the name of the attribute that should be used as the RDN attribute for the entry. The RDN attribute must be assigned a value in the body of the template definition, and the way in which the value is assigned must ensure that the value will be unique among all other entries created with the same template below the same parent.

---

**Note:** It is possible to specify multi-valued RDNs by separating the attribute names with a plus sign, as shown in the following example:

```
rdnAttr: uid+employeeNumber
```

---

If multi-valued RDNs are used, all of the RDN attributes must be defined values in the template body and the combination of the RDN values for each entry must be unique. However, it is possible for one or more of the attributes in the RDN to be non-unique as long as the combination is never duplicated.

In addition to the `template` and `rdnAttr` lines, you can include one or more `subordinateTemplate` lines. This enables you to include dynamically-generated entries below other entries that have been dynamically generated (for example, if each user entry has one or more entries below it), and to allow for complex hierarchies.

Although there is no limit placed on this level of nesting, you must ensure that no recursive loops are created by having a `subordinateTemplate` that either directly or indirectly will create additional entries using the same template.

Template definitions also support the concept of inheritance through the use of the `extends` keyword. For example, entries generated from the following template definition include all of the attributes defined in the `person` template as well as `userCertificate;binary` with the specified format:

```
template: certificatePerson
rdnAttr: uid
extends: person
userCertificate;binary:: <random:base64:1000>
```

Multiple inheritance is allowed (by including multiple lines with the `extends` keyword), but as with the `subordinateTemplate` keyword it is important not to create a recursive loop in which a template file could either directly or indirectly inherit from itself.

### 17.1.4.2 make-ldif Template File Tags

To ensure that `make-ldif` can generate LDIF files that can be used to simulate a wide variety of deployments, a large number of tags have been defined for use in templates. This section describes the standard set of tags that can be used in a `make-ldif` template file. You can also create custom tags, as described in Section 17.1.4.3, "Defining Custom Tags."

This section contains the following topics:

- Section 17.1.4.2.1, "Standard Replacement Tags"
- Section 17.1.4.2.2, "Attribute Value Reference Tags"
- Section 17.1.4.2.3, "Tag Evaluation Order"

#### 17.1.4.2.1 Standard Replacement Tags

The `make-ldif` standard replacement tags are special elements that are enclosed in angle brackets (beginning with a less-than sign (<) and ending with a greater-than sign (>) that are dynamically replaced with generated values. Some standard replacement tags do not require any arguments (for example, `<first>`). Others do take arguments, in which case the tag name comes first followed by a colon and the argument list with a colon between each argument (for example, `<random:numeric:5>`). The tag name is treated in a case-insensitive manner, although the arguments are generally case sensitive.

The following types of standard replacement tags are currently included as part of `make-ldif`:

**The DN tag**

The DN standard replacement tag is replaced with the DN of the current entry. If that DN is not yet available (for example, because the RDN attribute has not yet been assigned a value in the entry being generated), it is replaced with an empty string. In general, you should ensure that all RDN attributes are assigned values earlier in the template before this tag is used.

The DN tag can be used without any arguments (for example, `<DN>`), in which case it is replaced with the full DN of the entry. The tag can also take a single integer argument, which specifies the maximum number of components to include in the output. For example, the tag `<DN:1>` will only include the left most DN component (often called the RDN) for the entry. So if the entry being generated will have a DN of

uid=john.doe,ou=People,dc=example,dc=com, the tag <DN:1> will be replaced with uid=john.doe. If the argument value is negative rather than positive, then it takes the absolute value of the given argument value and takes that number of components from the end of the DN. For example, using a DN of uid=john.doe,ou=People,dc=example,dc=com the tag <DN:-1> is replaced with dc=com.

This tag can be used in both branch and template definitions.

#### The File tag

The File standard replacement tag is replaced with a line from a specified file. It requires either one or two arguments. The first argument is the path to the data file, and can be either an absolute path or the name of a file (with no path information) that is contained in the config/MakeLDIF directory. If there is a second argument, it must have a value of either sequential or random, which indicates whether the lines in the file should be taken in sequential order or chosen at random. If the second argument is not provided, the values are selected at random. For example, the tags <file:cities> and <file:cities:random> both cause the tag to be replaced with a randomly-selected line from the cities file, but the tag <file:cities:sequential> causes the city names to be taken in sequential order. If sequential ordering is used and all values are exhausted, it will wrap back around to the first line of the file.

The make-ldif command includes a number of standard data files that can be used in generated data. These files are included in the config/MakeLDIF directory and therefore only the filename is required. The files include:

cities — contains a list of common city names

first.names — contains a list of common first names

last.names — contains a list of common last names

states — contains a list of all two-character US state abbreviations

streets — contains a list of common street names

This tag can be used in both branch and template definitions.

#### The First tag

The First standard replacement tag is replaced with a first name taken from the config/MakeLDIF/first.names file. Note that there is a special relationship between the <first> and <last> tags such that the combination of the first and last names is always unique. When every possible combination from the first and last name files has been exhausted, make-ldif appends an integer value onto the last name to ensure that the value always remains unique.

The <first> tag does not take any arguments. It can be used only in template definitions. It is not allowed for use in branch definitions.

#### The GUID tag

The GUID standard replacement tag is replaced with a randomly generated GUID (globally-unique identifier) value. All GUID values generated are guaranteed to be unique. The values generated consist of 32 hexadecimal digits in dash-delimited groups of 8, 4, 4, 4, and 12 digits, respectively (for example, 12345678-90ab-cdef-1234-567890abcdef).

The <guid> tag does not take any arguments. It can be used in both branch and template definitions.

#### The IfAbsent tag

The IfAbsent standard replacement tag does not generate any value of its own, and is therefore always be replaced with an empty string. However, its value is that it can prevent an attribute from appearing in the entry altogether based on whether a specified attribute or attribute value exists.

For example, consider the following template:

```
template: example
rdnAttr: cn
objectClass: top
objectClass: untypedObject
objectClass: extensibleObject
cn: <guid>
displayName: <presence:50>{cn}
description: <ifabsent:displayName>{cn}
```

In this case, the `description` attribute is only included in the generated entry if the `displayName` attribute is not included (that is, the resulting entry will contain either `displayName` or `description` but not both).

The `IfAbsent` tag requires either one or two arguments. The first argument is the name of the target attribute. If there is a second argument, it specifies a particular value for the target attribute. If a value is provided, the `IfAbsent` tag takes action if that value is included in the generated entry.

This tag can be used in both branch and template definitions.

#### The IfPresent tag

The `IfPresent` standard replacement tag does not generate any value of its own, and is therefore always replaced with an empty string. However, its value is that it can prevent an attribute from appearing in the entry altogether based on whether a specified attribute or attribute value exists.

For example, consider the following template:

```
template: example
rdnAttr: cn
objectClass: top
objectClass: untypedObject
objectClass: extensibleObject
cn: <guid>
displayName: <presence:50>{cn}
description: <ifpresent:displayName>{cn}
```

In this case, the `description` attribute will only be included in the generated entry if the `displayName` attribute is also included (that is, the resulting entry will either contain neither attribute or it will contain both attributes).

The `IfPresent` tag requires either one or two arguments. The first argument is the name of the target attribute. If there is a second argument, it specifies a particular value for the target attribute. If a value is provided, the IfPresent tag will only take action if that value is included in the generated entry.

This tag can be used in both branch and template definitions.

#### The Last tag

The `Last` standard replacement tag is replaced with a last name taken from the `config/MakeLDIF/last.names` file. Note that there is a special relationship between the `<first>` and `<last>` tags such that the combination of the first and last names will always be unique. When every possible combination from the first and last

name file has been exhausted, `make-ldif` will append an integer value onto the last name to ensure that the value always remains unique.

The `<last>` tag does not take any arguments. It can only be used in template definitions. It is not allowed for use in branch definitions.

#### The List tag

The List standard replacement tag is replaced with a string selected from a provided list of values. The values to use should be provided as arguments to the List tag (at least one argument must be provided). Optionally, each value can be followed with a semicolon and an integer value that specifies the relative weight for that value. If a value does not include a weight, the weight for that item is assumed to be one. The weight is used to control how frequently the associated value is chosen compared with all of the other values in the list.

For example, to select from a list of the colors red, green, and blue in which all listed colors have equal weights, you can use:

```
<list:red:green:blue>
```

If the color red is to appear twice as frequently as either of the other colors, you can use:

```
<list:red;2:green;1:blue;1>
```

Note that in this case, the `1` following the green and blue elements are not technically needed since the weight of any item that does not explicitly include a weight is one, but it is provided in the example above for clarity.

This tag can be used in both branch and template definitions.

#### The ParentDN tag

The `ParentDN` standard replacement tag is replaced with the DN of the parent entry of the entry being generated. This should always be available.

This tag does not take any arguments. It can only be used in template definitions. It cannot be used in branch definitions.

#### The Presence tag

The `Presence` standard replacement tag does not generate any value of its own, and is therefore always replaced with an empty string. However, its value is that it can be used to cause the associated attribute to appear in the entry a specified percentage of the time.

For example, consider the following template:

```
template: example
rdnAttr: cn
objectClass: top
objectClass: untypedObject
objectClass: extensibleObject
cn: <guid>
displayName: <presence:50>{cn}
```

In this case, the `displayName` attribute will only be present in about 50% of the entries generated.

The `Presence` tag requires exactly one argument, which is an integer value between 0 and 100, indicating the percentage of entries that should have the associated attribute.

This tag can be used in both branch and template definitions.

#### The Random tag

The `Random` standard replacement tag is replaced with a randomly-generated value. A number of different types of values can be generated. This tag accepts a variable number of arguments, but the first argument always specifies the type of value to generate. That type may be one of the following values:

- **alpha.** This causes the tag to be replaced with a specified number of lowercase ASCII alphabetic characters (that is, the character set `abcdefghijklmnopqrstuvwxyz`). This requires exactly one more argument, which is an integer specifying the number of characters to include in the generated value. For example, `<random:alpha:5>` generates a string of five randomly-selected alphabetic characters.
- **numeric.** This causes the tag to be replaced with one or more numeric digits. There can be either one or two additional arguments. If there is one additional argument, it specifies the number of numeric digits to include in the value (for example, `<random:numeric:5>` will generate a string of five numeric digits). If there are two additional arguments, they will specify the upper and lower bounds for a randomly-generated number (for example, `<random:numeric:5:10>` will generate a random integer between 5 and 10, inclusive).
- **alphanumeric.** This causes the tag to be replaced with a specified number of lowercase ASCII alphabetic characters (that is, the character set `abcdefghijklmnopqrstuvwxyz`) and/or numeric digits (that is, the character set `0123456789`). This requires exactly one more argument, which is an integer specifying the number of characters to include in the generated value. For example, `<random:alphanumeric:5>` will generate a string of five randomly-selected alphanumeric characters.
- **chars.** This causes the tag to be replaced with characters from a user-defined character set. This can take either two or three additional arguments. The first additional argument is the characters for the user-defined character set. If there is a single argument after the character set, it specifies the number of characters to take from that set (for example, `<random:chars:abcd:3>` will cause three characters to be chosen in which each of those characters is either a, b, c, or d). If there are two arguments after the character set, they must be integer values and the number of characters generated will be an integer between this range (for example, `<random:chars:abcd:3:5>` will cause between 3 and 5 characters to be included in the value, where each character is either a, b, c, or d).
- **hex.** This causes the tag to be replaced with a specified number of hexadecimal characters (that is, the character set `0123456789abcdef`). This requires exactly one more argument, which is an integer specifying the number of characters to include in the generated value. For example, `<random:hex:5>` will generate a string of five randomly-selected hexadecimal characters.
- **base64.** This causes the tag to be replaced with a specified number of characters allowed in the base64 character set (`ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz01234567890+/`). This requires exactly one more argument, which is an integer specifying the number of characters to include in the generated value. For example, `<random:base64:5>` will generate a string of five randomly-selected hexadecimal characters.
- **month.**This causes the tag to be replaced with the name of a month of the year. If there are no additional arguments, the full name of the month is included (for example, `<random:month>` might return a value of `October`). If there is a single additional argument, it must be an integer value that specifies the maximum

number of characters to include from the name of the month (for example, `<random:month:3>` might generate a value of `Oct`).

- **telephone.**This causes the tag to be replaced with a randomly-generated telephone number in the format 123-456-7890. It does not take any additional arguments (that is, it should always be used like `<random:telephone>`).

This tag can be used in both branch and template definitions.

#### The RDN tag

The `RDN` standard replacement tag is replaced with the RDN (that is, the leftmost DN component) of the current entry. If the RDN is not yet available (for example, because the RDN attribute has not yet been assigned a value in the entry being generated), it will be replaced with an empty string. In general, you should ensure that all RDN attributes are assigned values earlier in the template before this tag is used. The behavior of this tag is identical to that of the DN tag when used with a single argument whose value is one (that is, `<dn:1>`).

The `RDN` tag does not take any arguments. It can be used in both branch and template definitions.

#### The Sequential tag

The `Sequential` standard replacement tag is replaced with an integer value. Each entry is given a sequentially-incrementing value (for example, the first entry is given a value of zero, the next entry a value of one, and so on).

This tag can take zero, one, or two arguments:

- If there are no arguments (that is, the tag is `<sequential>`), the first value will be zero, and the value will be reset to zero for each new branch.
- If there is a single argument, it must be an integer that specifies the initial value to use (for example, a tag of `<sequential:1000>` will start generating values at 1000 instead of 0). The value will be reset to the specified initial value for each new branch.
- If there are two arguments, the first must be an integer that specifies the initial value, and the second should be a Boolean value of either `true` or `false` indicating whether to reset the counter each time a new branch is started.

This tag can be used in both branch and template definitions.

#### The _DN tag

The `_DN` (note the leading underscore character) standard replacement tag is replaced with the DN of the entry being generated, but with an underscore used instead of a comma between DN components. Apart from using underscores instead of commas, this works exactly like the DN tag. As such, it can also take an optional integer argument that specifies the number of components from the left (or from the right if the value is negative) should be included.

This tag can be used in both branch and template definitions.

#### The _ParentDN tag

The `_ParentDN` (note the leading underscore character) standard replacement tag is replaced with the DN of the parent entry of the entry being generated, but with an underscore used instead of a comma between DN components. This should always be available.

This tag does not take any arguments. It can only be used in template definitions. It cannot be used in branch definitions.

#### 17.1.4.2.2 Attribute Value Reference Tags

Attribute value reference tags can be used to replace the tag with the value of a specified attribute from the same entry. They are used by enclosing the name of the desired attribute in curly braces. For example, `{cn}` will be replaced with the value of the `cn` attribute, if it has already been given a value in the target entry. If the target attribute has not yet been given a value in the entry, the tag will be replaced with an empty string.

For example, consider the following excerpt from a template:

```
givenName: <first>
sn: <last>
uid: {givenName}.{sn}
cn: {givenName} {sn}
mail: {uid}@example.com
```

If the value chosen for the first name is `John` and the last name is `Doe`, then the resulting LDIF output would be:

```
givenName: John
sn: Doe
uid: John.Doe
cn: John Doe
mail: John.Doe@example.com
```

It is also possible to place a colon after the name of the attribute followed by a positive integer value specifying the maximum number of characters to include from the target attribute. For example, the template excerpt:

```
givenName: <first>
sn: <last>
initials: {givenName:1}{sn:1}
```

would cause the following LDIF to be generated:

```
givenName: John
sn: Doe
initials: JD
```

If the specified length is longer than the value of the named attribute, the entire value is used with no padding added. Otherwise, the specified number of characters are taken from the value.

#### 17.1.4.2.3 Tag Evaluation Order

All tags in the `make-ldif` syntax are currently given equal priority. As such, they are evaluated in the order that they appear in the template definition, from top to bottom, and from left to right within a given line. It is not possible to embed one tag within another.

### 17.1.4.3 Defining Custom Tags

The make-ldif utility has been designed in an extensible manner so that new tags can be defined and used in template files.

All tags must be subclasses of the `org.opends.server.tools.makeldif.Tag` abstract class. Custom tag definitions must include the following methods:

`public String getName()`
This retrieves the name that should be used to reference the tag. The value that it returns must be unique among all other tags in use by the server.

`public boolean allowedInBranch()`
This indicates whether the tag will be allowed in branch definitions. If it returns a value of `true`, then the tag may be used in both branch and template definitions. If it returns a value of `false`, then the tag may be used in template definitions but not branch definitions.

`public void initializeForBranch(TemplateFile templateFile, Branch branch, String[] arguments, int lineNumber, List<String> warnings)`
This performs any initialization that may be required if the tag is to be used in a branch definition. This does not need to be implemented if `allowedInBranch()` returns `false`.

`public void initializeForTemplate(TemplateFile templateFile, Template template, String[] arguments, int lineNumber, List<String> warnings)`
This performs any initialization that may be required of the tag is to be used in a template definition.

`public void initializeForParent(TemplateEntry parentEntry)`
This performs any initialization that may be required before starting to generate entries below a new parent. This does not need to be implemented if no special initialization is required.

`public TagResult generateValue(TemplateEntry templateEntry, TemplateValue templateValue)`
This generates the value that will be used to replace the associated tag when generating entries.

All of the tags available in `make-ldif` are included in the `org.opends.server.tools.makeldif` package. They may be used for reference to understand what is involved in implementing a custom tag.

> **Note:** If you define a custom tag, ensure that it is available for use in any template file that might need it. This is done using the `include` statement, that should appear at the top of the template file. For more information, see Section 17.1.4.1.1, "Custom Tag Includes."

## 17.2 Importing Large Data Sets

The topics in this section provide tips on improving performance when importing large data sets to the directory server. By default, the server imports data with a fixed set of parameters. You can change the default behavior in two ways:

- Specify certain options when you run the `import-ldif` command.

  For more information, see Section 17.2.1, "Setting the Import Options."

- Use the `dsjavaproperties` command to set the appropriate Java arguments before running the `import-ldif` command.

  For more information, see Section 17.2.2, "Tuning the JVM and Java Arguments."

### 17.2.1 Setting the Import Options

The following options of the `import-ldif` command are useful when you are importing particularly large databases:

- `--skipDNValidation`

  This option significantly speeds up a large import because no DN validation or database loading is performed during the first phase of the import. The DNs in the LDIF file are treated as regular indexes and are written to a scratch index file that is loaded in phase two of the import.

  During the second phase of the import, limited DN parental checking is performed. During this evaluation, the DNs in the LDIF file are examined to make sure that each DN has a correct parent DN. When a DN is detected without a parent, a dummy entry is written to the reject file.

  If the `--skipDNValidation` option is specified, no duplicate DN checking is performed.

  The server does not remove bad entry IDs from the index database during phase two of the import. It is therefore essential that the LDIF import file is correct if the `--skipDNValidation` option is specified. Correct LDIF files are generally those that are generated by using the `make-ldif` command, LDIF files exported from an LDAP server, or LDIF files created by scripts that are historically known to generate correct LDIF files.

- `--threadCount`

  This option speeds up a large import by enabling you to specify that more threads are dedicated to the import process. By default, two threads per CPU are used for an import operation.

  Increasing the `--thread-count` also increases the buffer space that is required in phase one of the LDIF import.

- `--tmpDirectory`

  In the first phase of the import, the server parses the LDIF file, sorts the index records, and writes the records to temporary files. By default, the temporary index files are written to `intall-dir/import-tmp`. If you are importing particularly large index files, you might want to specify another location that has more disk space.

  The amount of space required for the temporary index files depends on the following factors:

  - The number of entries in the LDIF file.
  - The size of the entries in the LDIF file.

    Entries with large numbers of attributes that require indexing will require more space in the temporary directory location, and in the database directory.

  - The number of indexes that are configured.

    The more indexes that are configured, the more disk space is required in the temporary directory location, and in the database directory. Substring indexes require more temporary disk space to process than other types of indexes.

  - Increasing the `index-entry-limit` for all indexes, or for individual indexes, requires more disk space.

This is especially true for substring indexes. If you are importing an LDIF file with a large number of entries, you should turn off all substring indexing to prevent a number of the index records will hitting the `index-entry-limit`.

### 17.2.2 Tuning the JVM and Java Arguments

Tuning the JVM heap is essential to the performance of the `import-ldif` command. Although the `import-ldif` command attempts to limit the amount of JVM heap that it requires, you should allocate as large a JVM heap as possible to `import-ldif` if you are importing a large number of entries.

The following JVM tuning considerations have specific impact on the `import-ldif` operation:

- Performing an online import uses the JVM settings that were specified when the server was started. If you plan to import a large LDIF file by using the online import, you should provide extra JVM heap when the server is started. In general, if you need to import a large LDIF file, the best option is to perform an offline import.
- The 32-bit JVM generally performs better for smaller LDIF files and for most larger LDIF files.

  You should always try this JVM first, with as large a heap as can be spared. A minimum heap of 2 Gbytes is recommended.
- You might require a 64-bit JVM with a large JVM heap (greater than 4 Gbytes) for extremely large LDIF files, depending on the size of the entries and the indexes configured.

  The 64-bit JVM does not generally perform as well as the 32-bit JVM.
- The default JVM ergonomics might be too small for some JVMs and can seriously impact performance.

  Take note of the default ergonomic values for your JVM (these values differ by vendor and by operating system).
- If you are using replication, you should budget additional JVM heap, particularly if you plan to do a full initialization of the other replicas in the topology after an online import.
- Enable parallel garbage collection for large imports.
- Use the Concurrent Mark Sweep (CMS) garbage collector. This option allows the JVM to minimize the response time of LDAP operations, but it can have a small impact on the overall performance (throughput) of the server.

When you have calculated the memory requirement, perform the following steps:

1. Edit the `instance-dir/OUD/config/java.properties` file and set the following values:

   ```
   overwrite-env-java-args=true
   import-ldif.offline.java-args=-Xms2560M -Xmx2560M -XX:+UseParallelGC
   -XX:+UseConcMarkSweepGC
   ```

2. Run the `dsjavaproperties` command:

   ```
   $ bin/dsjavaproperties
   ```

**Note:** Running the `dsjavaproperties` command, or setting the `OPENDS_JAVA_ARGS` environment variable, only has a performance impact if the import is offline. If the server is already running and you perform an online import, changing the Java arguments has no impact on the import performance because the import is performed by the server JVM.

## 17.3 Backing Up and Restoring Data

Oracle Unified Directory provides an extensible framework that supports a variety of repository types. The directory server uses the Berkeley DB Java Edition (JE) as its primary back end. The JE back end provides some advantages over other databases as it provides a high-performance, scalable transactional B-tree database with full support for ACID semantics for small to very large data sets. It can also store its entries in encoded form and provide indexes for fast, efficient data retrieval.

This section covers the following topics:

- Section 17.3.1, "Overview of the Backup and Restore Process"
- Section 17.3.2, "Backing Up Data"
- Section 17.3.3, "Backing Up the Server Configuration"
- Section 17.3.4, "Backing Up for Disaster Recovery"
- Section 17.3.5, "Backing up and Restoring Data Using File System Snapshots"
- Section 17.3.6, "Restoring Data"
- Section 17.3.7, "Restoring Replicated Directory Servers"

### 17.3.1 Overview of the Backup and Restore Process

To maintain the directory data on the JE back end, Oracle Unified Directory provides efficient backup and restore utilities that support full and incremental backups. A *full backup* saves the directory data files in the environment as a compressed archive file. An *incremental backup* saves and compresses just those files that have been written since the previous backup, together with a list of names of files that are unchanged since the previous backup. Oracle Unified Directory stores its backup information in a *backup back end* for easy restores

Directory server backups can be made on the local disks or on remote disks, for example, on network-attached storage (NAS). If you run a backup locally, you should then copy and store the backup on a different machine or file system for security purposes.

Before you start backing up and restoring data, consider the following:

- You must design a workable backup and restore strategy for your directory services system. For example, you can run an incremental backup daily and perform a full backup at least once a week. Test your backup process and your ability to restore regularly. For data restores, many companies restore a directory server from a replicated server, which ensures that the most update copy of the directory data is used. Backup tapes are still needed if the directory data is damaged (for example, missing entries) and the corrupted data has been replicated to other servers.
- Ensure that you have a disaster recovery plan in place. Disaster recovery is necessary when catastrophic events, data corruption, or data tampering occurs.

Companies devise their own plans or out source the work to third party specialists. See Section 17.3.4, "Backing Up for Disaster Recovery" for more information.

- Ensure that you have a place to store your back ups. Store the archived data, configuration directory, schema subdirectory, and installation directory used for your server together in a single location. All these items are required when you restore the server.

## 17.3.2 Backing Up Data

The directory server provides an efficient command-line utility (`backup`) to back up databases. The `backup` command can be run immediately or scheduled as a task. If the backup is scheduled, the command contacts the server over SSL, using the administration connector, and registers a backup task. If no connection options are specified, the command runs immediately.

The following procedures show the use of the `backup` command in various backup scenarios.

### 17.3.2.1 To Back Up All Back Ends

You can back up all back ends end by using the `--backUpAll` option.

The following command is run on a standalone directory server and specifies that all databases should be backed up, compresses the backup file, and saves the file to a specified location.

```
$ backup --backUpAll --compress --backupDirectory /tmp/backup
```

The backup directory contains subdirectories for each back end:

```
$ ls /tmp/backup
./ ../ config/ schema/ tasks/ userRoot/
```

The `backup` utility writes the backup to the specified directory and creates a `backup.info` file that provides details about the backup. The directory server assigns a backup ID based on the current date and time. To create your own ID, use the `--backupID` option:

```
$ ls /tmp/backup/config
./ backup.info
../ config-backup-20070827153501Z
```

The `backup.info` file contains detailed information about the current backup.

```
$ more /tmp/backup/config/backup.info
backend_dn=ds-cfg-backend-id=config,cn=Backends,cn=config

backup_id=20070827153501Z
backup_date=20070827153511Z
incremental=false
compressed=true
encrypted=false
property.archive_file=config-backup-20070827153501Z
```

### 17.3.2.2 To Back Up All Back Ends with Encryption and Signed Hashes

The `backup` utility provides encryption and signed hash support for secure backups. The use of the encryption and signed hash options requires a connection to an online server instance, so the appropriate connection options must be specified.

Run the backup command.

The following command backs up all back ends, compresses them, generates a hash, signs the hash, and encrypts the data.

```
$ backup -h localhost -p 4444 -D "cn=directory manager" -j pwd-file --backUpAll \
  -X --compress --hash --signHash --encrypt --backupID 123 \
  --backupDirectory /tmp/backup
```

#### 17.3.2.3 To Perform an Incremental Backup on All Back Ends

Incremental backups save only those changes that have occurred since the last backup (full or incremental). The main advantage of an incremental backup is the faster time to back up a system when compared to that of full backups. The disadvantage of an incremental backup is that each incremental backup must be restored, which requires more time and care than that of a full restore.

To perform an incremental backup, run the backup command with the `--incremental` option, as follows:

```
$ backup --backUpAll --incremental --compress --backupDirectory /tmp/backup
```

#### 17.3.2.4 To Back Up a Specific Back End

You can back up a single back end by using the `--backendID` option, which specifies the back end to save.

1. List the back ends that are configured on the server, by running the `list-backends` command. For example:

   ```
   $ list-backends

   Backend ID     Base DN
   -------------- -----------------
   adminRoot      cn=admin data
   ads-truststore cn=trust-store
   backup         cn=backups
   config         cn=config
   monitor        cn=monitor
   schema         cn=schema
   tasks          cn=tasks
   userRoot       dc=example,dc=com
   ```

2. Run the backup command with the `--backendID` option.

   For example, to back up the `userRoot` back end, run the following command:

   ```
   $ backup --backendID userRoot --backupDirectory /tmp/backup
   ```

   If you back up a single back end and replication is configured, any changes that you make to that back end are stored in the change log on the replication server. When you restore that back end, the replication server detects that the back end is not up to date and replays the changes made after the backup. This behavior occurs even if there is only one directory server in the replicated topology, because the changes are stored on the replication server.

   If you do not want this behavior, back up all back ends in a replicated environment. This ensures that the data, and the replication server are backed up. In this case when a restore is done, the directory server and the replication server are restored to their state before the back up, and no memory of subsequent changes remains.

#### 17.3.2.5 To Perform an Incremental Backup on a Specific Back End

1. List the back ends that are configured on the server, by running the `list-backends` command. For example:

```
$ list-backends

Backend ID     Base DN
-------------- -----------------
adminRoot      cn=admin data
ads-truststore cn=trust-store
backup         cn=backups
config         cn=config
monitor        cn=monitor
schema         cn=schema
tasks          cn=tasks
userRoot       dc=example,dc=com
```

2. Run the backup command with the `--incremental` option.

```
$ backup --incremental --backendID userRoot --backupDirectory /tmp/backup
```

#### 17.3.2.6 To Schedule a Backup as a Task

The directory server provides a task back end for processing administrative tasks, such as backups and restores. You can specify the start time for a backup or restore by using the `-t` or `--start` option. If one of these options is provided, the utility exits immediately after scheduling the task. To schedule a task for immediate execution and have the utility exit immediately after scheduling the task, specify `0` as the value for the start time. If the `-t` or `--start` option is omitted, the utility schedules the task for immediate execution and tracks the task's progress, printing log messages as they are available and exiting when the task has completed.

Access to the task back end is provided over SSL via the administration connector. If you schedule the backup as a task, you must therefore specify how the SSL certificate will be trusted. This example schedules a backup for execution at a future time. The `-X` option specifies that all certificates presented by the server are trusted. For more information, see Section 14.3, "Managing Administration Traffic to the Server."

1. Run the `backup` command with the following options:

```
$ backup --port 4444 --bindDN "cn=Directory Manager" \
  --bindPasswordFile pwd-file -X \
  --backUpAll --backupDirectory /tmp/backups --start 20080601121500 \
  --completionNotify admin@example.com --errorNotify admin@example.com
```

2. View information about the scheduled task by using the `manage-tasks` command. For example:

```
$ manage-tasks --port 4444 --bindDN "cn=Directory Manager" \
  --bindPasswordFile pwd-file -X --info 2008040210324704 --no-prompt
```

### 17.3.3 Backing Up the Server Configuration

All configuration settings for a directory server instance are stored in the `config.ldif` file, which is located in the `config` directory. The directory server automatically saves the `config.ldif` file to ensure that changes are properly accounted for in the configuration. The file is saved at two specific times:

- **At startup**. If the current configuration does not match the archived configuration, the server saves the `config.ldif` file.

- **At modification time**. Whenever a directory administrator makes changes to the configuration by using the `dsconfig` utility with the server online, the directory server saves the `config.ldif` file prior to the change.

You can access archived configuration files from the `INSTANCE_DIR/OUD/config/archived-configs` directory. This directory lists each saved configuration file, compresses it as a `.gz` file, and saves the configuration as `config-timestamp.gz`. For example, you can see archived `config.ldif` files as follows:

```
$ ls config/archived-configs
09/02/2010 03:43 PM 9,045 config-20100819055359Z.gz
```

## 17.3.4 Backing Up for Disaster Recovery

Directory and system administrators should have a disaster recovery plan in place in the event of a natural, human-induced, or catastrophic disaster. If your directory service is distributed over multiple individual servers, back up all the servers individually or back up all the directory data from a central location.

Alternatively, consider replication as a backup and restore strategy. Replication provides faster restores and more update data from another replicated server. For more information, see Section 17.3.7, "Restoring Replicated Directory Servers."

### 17.3.4.1 To Back Up the Directory Server For Disaster Recovery

1. Make a backup of all back ends by using the `--backUpAll` option, for example:

   ```
   $ backup --backUpAll --backupDirectory /tmp/backup
   ```

2. Copy the configuration directory, `INSTANCE_DIR /OUD/config`.

   Make sure that the `schema` subdirectory is present within the `INSTANCE_DIR /OUD/config` directory.

3. Copy the files in `INSTANCE_DIR/OUD/logs`.

4. Make a copy of the installation directory.

5. Store the archived data, configuration directory, schema subdirectory, log files and installation directory together in a single location.

   All items are required when restoring the server.

## 17.3.5 Backing up and Restoring Data Using File System Snapshots

For certain deployments, file system snapshot technologies offer a viable alternative to the traditional backup. On Solaris systems, ZFS enables file system snapshots that are space efficient, very quick to create, and portable between systems. By dedicating a Directory Server per data center, or two if your entire service runs in one data center, you deploy an effective, redundant solution for restoring data as part of your disaster recovery plan.

This section contains the following topics:

- Section 17.3.5.1, "To Take a ZFS Snapshot On a Dedicated Backup Server"
- Section 17.3.5.2, "To Restore a Directory Server From a ZFS Snapshot"

### 17.3.5.1 To Take a ZFS Snapshot On a Dedicated Backup Server

1. Because the Directory Server is dedicated to backup, configure the server as a read-only replica if you have not already done so.

```
$ dsconfig -h host -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  set-global-configuration-prop --set writability-mode:internal-only
```

When you restore a server from the snapshot of the read-only replica, the restored server accepts only replication traffic until you enable writability, after the server has caught up with other replicas in the topology.

2. Take the ZFS snapshot.

   For example, if the Directory Server files are stored in the file system corresponding to `zpool/DS_FS`, the command is:

   ```
   $ zfs snapshot zpool/DS_FS@{todays_date}
   ```

3. Back up the snapshot to other storage.

   ```
   $ zfs send zpool/DS_FS@{today_date} > /backups/DS_FS.{today_date}.zfs
   ```

   Do not keep snapshots longer than the replication purge delay, because when you restore from a snapshot, the replication mechanism has to be able to replay all the missing changes on the replica.

#### 17.3.5.2 To Restore a Directory Server From a ZFS Snapshot

1. Import the backup `zpool`.

   Create a ZFS file system to access the backup pool, using `/backups` as the mount point.

2. Stop the Directory Server that is being restored.

3. Initialize the ZFS file system from `/backups`.

   ```
   $ dd if=/backups/DS_FS.{date_to_restore}.zfs bs=32k | zfs receive -F
   zpool/DS_FS
   ```

4. Adapt the configuration as necessary to use the host name and port numbers of the Directory Server to restore.

5. Start the Directory Server.

6. Monitor replication until you observe that the Directory Server is in sync with other replicas in the topology.

7. Set the `writability-mode` to `enabled`, allowing the Directory Server to process write operations from clients.

   ```
   $ dsconfig -h restored-host -p 4444 -D "cn=Directory Manager" -j pwd-file -X \
     -n set-global-configuration-prop --set writability-mode:enabled
   ```

### 17.3.6 Restoring Data

You can restore data by using the `restore` utility. The `restore` utility allows you to restore only one back end at a time. The directory server must be stopped prior to a restore, unless you are scheduling a restore task, or you are restoring data that has been signed or hashed.

This section contains the following topics:

- Section 17.3.6.4, "To Restore the Configuration File"
- Section 17.3.6.5, "To Restore a Directory Server During Disaster Recovery"

### 17.3.6.1 To Restore a Back End

1. Stop the server, if it is running.
2. Display the backup information by running the `restore` command with the `--listBackups` option. For example:

```
$ restore --listBackups --backupDirectory backup/userRoot
Backup ID: 20080827153501Z
Backup Date: 27/Aug/2008:10:35:11 -0500
Is Incremental: false
Is Compressed: true
Is Encrypted: false
Has Unsigned Hash: false
Has Signed Hash: false
Dependent Upon: none
```

3. Restore the back end.

```
$ restore --backupDirectory backup/userRoot
```

4. Repeat the restore for the other back ends.

### 17.3.6.2 To Restore a Back End From Incremental Backups

Typically, system administrators run a weekly full backup with daily incremental backups. Be aware that it takes longer to restore your system from incremental backups.

1. Restore the last full backup on your system by using the `restore` command.

   Each back end must be restored individually.

2. Restore each incremental backup by using the `restore` command.

   Restore each incremental backup starting from the last full backup.

### 17.3.6.3 To Schedule a Restore as a Task

The directory server provides a task back end for processing administrative tasks, such as backups and restores. You can specify the start time for a restore by using the `-t` or `--start` option. If one of these options is provided, the utility exits immediately after scheduling the task. To schedule a task for immediate execution and have the utility exit immediately after scheduling the task, specify `0` as the value for the start time. If the `-t` or `--start` option is omitted, the utility schedules the task for immediate execution and tracks the task's progress, printing log messages as they are available and exiting when the task has completed.

Access to the task back end is provided over SSL, using the administration connector. If you schedule the restore as a task, you must therefore specify how the SSL certificate will be trusted.

1. Ensure that the server is stopped prior to the scheduled restore time.
2. Schedule the restore by using the `-t` or `--start` option of the `restore` command.

   The following command restores the `userRoot` back end at a scheduled start time by using the `--start` option. The restore sends a completion and error

notification to admin@example.com. The -X option specifies that all certificates presented by the server are trusted.

```
$ restore -p 4444 -D "cn=Directory Manager" -j pwd-file -X \
  -d /backup/userRoot --start 20080125121500 --completionNotify
admin@example.com \
  --errorNotify admin@example.com
```

3. You can view this scheduled task by using the manage-tasks utility.

   For more information, see Section 14.4, "Configuring Commands As Tasks."

#### 17.3.6.4 To Restore the Configuration File

You might need to restore the configuration file to transfer the configuration to another server, for disaster recovery purposes, or for other events. In general, if a server is online, the current configuration file is equivalent to the latest archived configuration file. However, you can choose to restore the config.ldif file from a previous date.

1. Stop the server if it is running.
2. Locate the required configuration file on the system. For example:

```
$ ls INSTANCE_DIR/OUD/config/archived-configs
./
../
config-20110817192057Z.gz
config-20110827153200Z.gz
config-20110817192052Z.gz
config-20110827153214Z-2.gz
```

3. Manually decompress the archived configuration file, using a decompression utility such as gunzip.
4. Copy the file to the config directory, replacing the current config.ldif file.

```
$ cp config-20110817192057Z INSTANCE_DIR/OUD/config/config.ldif
```

#### 17.3.6.5 To Restore a Directory Server During Disaster Recovery

1. Install the same version of the directory server that was previously installed on the host.
2. Create a server instance by using the setup command.
3. Copy the saved config directory to INSTANCE_DIR/OUD/config.

   The config.ldif file should reside in this directory. The saved schema subdirectory should be located in INSTANCE_DIR/OUD/config/schema.
4. Check that the configuration for the restored server is correct.
5. Restore the individual back ends by using the restore command.

### 17.3.7 Restoring Replicated Directory Servers

Performing binary restores in replicated environments requires special care depending on your replicated topology. If possible, update your back end by using the replication mechanisms in your system instead of restoring it from a backup. Replication has distinct advantages over traditional tape backups. Data restores are much faster than tape restores, and the data is more up to date. However, tapes are still needed in the event that the replicated data is corrupt and has been propagated to other servers.

When restoring a replicated server, ensure that the configuration file `INSTANCE_DIR/OUD/config/config.ldif` is the same as when the backup was made. Restore the `config.ldif` file prior to restoring the server back ends.

You cannot restore an old backup to a master server because it might be out of date. Rather allow the replication mechanism to bring a master up to date with the other master servers by setting that master to read-only. When the master has been synchronized, you can reset it to read-write.

If you need to restore a replicated server, reinitialize the server from one of the other replicated servers by importing an LDIF file.

For very large databases (millions of entries), make a binary copy of one server and restore it on the other replicated server.

If you have a fairly recent backup (one that is not older than the maximum age of the change log contents on any of the other replicated servers), you can use that version to restore your data. When the old backup is restored, the other servers will update that server with recent updates made since the backup was saved.

## 17.3.8 Deleting Backup Data

If you run regular backups, the backup files might start to consume too much disk space. You must remove the old backup files manually to create space for new ones.

### 17.3.8.1 To Delete Backup Files

When you delete backup files manually, make sure that you do not break any dependencies between backup sets.

1. List the existing backups in your backup directory.

   For example, to list the backups in the default backup directory, run the following command:

   ```
   UNIX: $ ls INSTANCE_DIR/OUD/bak
          backup-userRoot-20110929184101Z backup-userRoot-20111029184509Z
          backup.info backup.info.save

   WINDOWS: C:\> dir INSTANCE_DIR\OUD\bak
            backup-userRoot-20110929184101Z backup-userRoot-20111029184509Z
            backup.info backup.info.save
   ```

2. Delete the backup file from the backup directory.

   For example, to remove the oldest backup of the `userRoot` database in the preceding step, run the following command:

   ```
   UNIX: $ rm INSTANCE_DIR/OUD/bak/backup-userRoot-20110929184101Z

   WINDOWS C:\> del INSTANCE_DIR\OUD\bak\backup-userRoot-20110929184101Z
   ```

3. Remove the associated backup information from the `backup.info` file.

   You can display the contents of the `backup.info`, as follows (on UNIX systems):

   ```
   $ more INSTANCE_DIR/OUD/bak/backup.info
   backend_dn=ds-cfg-backend-id=userRoot,cn=Backends,cn=config

        backup_id=20110929184101Z
        backup_date=20110929184104Z
        incremental=false
        compressed=false
   ```

```
encrypted=false
property.last_logfile_name=00000000.jdb
property.last_logfile_size=160773
property.archive_file=backup-userRoot-20110929184101Z

backup_id=20111029184509Z
backup_date=20111029184512Z
incremental=false
compressed=false
encrypted=false
property.last_logfile_name=00000000.jdb
property.last_logfile_size=160773
property.archive_file=backup-userRoot-20111029184509Z
```

For Windows systems, use an appropriate text editor.

## 17.4 Searching Directory Data

The directory server provides a suite of LDAPv3-compliant command-line tools, including a sophisticated look-up operation in the form of a search function and filters. You can also use Oracle Directory Services Manager to search directory data. This section explains how to use the `ldapsearch` command-line utility and Oracle Directory Services Manager to locate entries in the directory.

This section contains the following topics:

- Section 17.4.1, "Overview of the `ldapsearch` Command"
- Section 17.4.2, "`ldapsearch` Location and Format"
- Section 17.4.3, "Understanding Search Criteria"
- Section 17.4.4, "`ldapsearch` Examples"
- Section 17.4.5, "Searching Data With Oracle Directory Services Manager"

### 17.4.1 Overview of the `ldapsearch` Command

The `ldapsearch` command allows you to enter a search request where you specify the host name, port, bind DN and password plus search criteria to locate entries in the directory. When an LDAP client makes a search request to the directory server, it opens a connection to the directory server over TCP/IP. The client then performs a *bind* operation to the directory server by attempting to match a given entry, which effectively authenticates the client. Most users have the option to bind as a particular user, such as a Directory Administrator or themselves, or to not bind as any user, in which case the directory server assumes that the user is bound as an *anonymous* user.

Because all access to directory data is based on how a connection is bound, the directory server checks the client's privileges to see if the client can run a particular search operation. After the directory server checks the user's access rights, the client passes a search request consisting of a set of search criteria and options to the directory server.

The directory server searches all entries that match the search criteria and options. It then returns the entries, the DN, and all attributes for each entry, in the form of LDIF text to standard output. If an error occurs, the directory server displays an error message indicating the error. Finally, the client closes the connection when the search operation has completed.

## 17.4.2 `ldapsearch` Location and Format

The `ldapsearch` utility is found in the following location:

```
(UNIX, Linux) INSTANCE_DIR/OUD/bin
(Windows) INSTANCE_DIR\OUD\bat
```

The utility has the following format:

```
ldapsearch optional-options search-filter optional-list-of-attributes
```

where:

- *optional-options* are command-line options that must appear before the search filter.
- *search-filter* is an LDAP search filter either specified on the command-line or in a file.
- *optional-list-of-attributes* is a list of attributes separated by a space. The list of attributes must appear after the search filter.

### 17.4.2.1 Common `ldapsearch` Options

The `ldapsearch` command has many options to search entries in the directory. Options are allowed in either their short form (for example, `-b baseDN`) or their long form (for example, `--baseDN`). The most common command options to use with `ldapsearch` are as follows:

`-h, --hostname address`
Specifies the host name or IP address of the directory server on which the search should be run. It can be an IP address or a resolvable name. If this is not provided, a default value of `localhost` is used.

`-p, --port port`
Specifies the directory server port. It should be an integer value between `1` and `65535`, inclusive. If this is not provided, a default port of `389` is used.

`-b, --baseDN baseDN`
Specifies the base DN to use for the search operation. If a file containing multiple filters is provided using the `--filename` option, this base DN is used for all of the searches. This is a required option.

`-s, --searchScope scope`
Sets the scope for the search operation. Its value must be one of the following:

- `base`. Searches only the entry specified by the `--baseDN` or `-b` option.
- `one`. Searches only the entry specified by the `--baseDN` or `-b` option and its immediate children.
- `sub` or `subordinate`. Searches the entire subtree whose base is the entry specified by the `--baseDN` or `-b` option. This is the default option when no `--searchScope` option is provided.

`-D, --bindDN bindDN`
Specifies the DN to use when binding to the directory server through simple authentication. This option is not required when using SASL authentication or anonymous binding.

`-w, --bindPassword bindPassword`
Specifies the password to use when binding to the directory server. This option is used for simple authentication, as well as for password-based SASL mechanisms like

CRAM-MD5, DIGEST-MD5, and PLAIN. It is not required if anonymous binding is used. This option must not be used in conjunction with the `--bindPasswordFile` option. To prompt for the password, type `-w -`.

`-l, --timeLimit numSeconds`
Sets the maximum length of time in seconds that the directory server should spend processing any search request. If this is not provided, no time limit is imposed by the client. Note that the directory server may enforce a lower time limit than the one requested by the client.

`-z, --sizeLimit numEntries`
Sets the maximum number of matching entries that the directory server should return to the client. If this is not provided, no maximum size is imposed by the client. Note that the directory server may enforce a lower size limit than the one requested by the client.

`-S, --sortOrder sortOrder`
Sorts the results before returning them to the client. The sort order is a comma-delimited list of sort keys, where each sort key consists of the following elements:

- `+/- (plus or minus sign)`. Indicates that the sort should be in ascending (+) or descending (-) order. If this value is omitted, the sort uses ascending order by default.
- `Attribute name`. The name of the attribute to sort the data. This element is required.
- `Name` or `OID Matching Rule`. An optional colon followed by the name or OID of the matching rule used to perform the sort. If this is not provided, the default ordering matching rule for the specified attribute type is used.

  For example, the sort order string `sn,givenName` sorts the entries in ascending order first by `sn` and then by `givenName`. Alternately, using `-modifyTimestamp`, the directory server sorts the `modifyTimestamp` attributes with the most recent values first.

### 17.4.3 Understanding Search Criteria

The `ldapsearch` command requires three sets of information to specify where and what to search in the directory information tree:

- **Base DN**. By specifying the base DN, you are defining the topmost distinguished name (DN) or starting point in the directory to conduct the search. All searches begin at or below the base DN, depending on the scope, and move down the tree, never upwards. Examples of base DNs are: `dc=example,dc=com` and `ou=People,dc=example,dc=com`.
- **Scope**. The scope determines which set of entries at or below the base DN should be evaluated by the search filter. The search scope and base DN together indicate "where" to look for entries in the directory.
- **Search filter**. The search filter specifies the conditions that the entries must meet to be returned to the client.

This section describes the different filter options, and contains the following topics:

- Section 17.4.3.1, "Specifying Filter Types and Operators"
- Section 17.4.3.2, "Using Compound Search Filters"

- Section 17.4.3.3, "Using UTF-8 Encoding in Search Filters"
- Section 17.4.3.4, "Using Special Characters in Search Filters"

### 17.4.3.1 Specifying Filter Types and Operators

The directory server provides seven types of search filters, defined in the LDAP protocol. With each search filter type, you use operators that test the relationships between two entities, *attribute* and *value*.

The following table shows how search filters are used to return specific entries in a search query.

| Search Filter | Operator | Description |
|---|---|---|
| Presence | `attr=*` | Return all entries that have any value associated with the specified attribute. The filter uses the wildcard character to denote zero or more characters in the string. For example, the following filter is common and returns all entries that have an object class with any value, which every entry has: `(objectclass=\*)`.<br><br>**Note:** the LDAP protocol specifies that filters should have the form "(filter)", which includes parentheses surrounded by quotation marks. Although most directory servers accept filters without the parentheses and quotation marks, it is good practice to include them. |
| Equality | `attr=value` | Return entries containing attributes equal to a specified value. For example: `(sn=Bergin)` returns all entries that have a surname (`sn`) attribute with the value of `Bergin`.<br><br>**Note:** The `sn` value is case insensitive, so entries associated with `sn=bergin` or `sn=Bergin` will be returned. |

| Search Filter | Operator | Description |
|---|---|---|
| Substring | `attr=<initial-string><any substring><final-string>` | Return entries with attributes containing a specified substring or partial substring. The filter uses the wildcard character to denote zero or more characters in the string.<br>■ Run an initial substring search that looks for all attribute values that have the characters `Ber` at the start of the string: `(sn=Ber\*)`<br>■ Specify the middle substring of an attribute value. For example: `(sn=\*erg\*)`<br>■ Specify the end of a substring of an attribute value. For example: `(sn=\*gin)`. Or you can specify some combination of substrings<br>■ Specify the initial and middle substring: `(sn=ber\*gi\*)`<br>■ Specify the initial and ending substrings: `(sn=be\*in)`<br>■ Specify the middle and end substrings: `(sn=\*er\*in)`<br>**Note:** Substring filters do not use true wild cards such as in system listings or regular expressions. Thus, the following filter would be invalid because of too many criteria: `(sn=\*B\*rg\*n)`. |
| Greater than or equal to | `attr>=value` | Return entries containing attributes that are greater than or equal to the specified value. For example, `(sn>=Bergin)` returns all entries that have an attribute greater than or equal to the value, `Bergin`, based on the matching rules for attributes (see Understanding Matching Rules). |
| Less than or equal to | `attr<=value` | Return entries containing attributes that are less than or equal to the specified value. For example, `(sn<=Bergin)` returns all entries that have an attribute less than or equal to the value, `Bergin`, based on the matching rules for attributes. |
| Approximate | `attr~=value` | Return entries containing the specified attribute with a value that is approximately equal to the value specified in the search filter. For example: `(sn~=Bergan)` could return the entry associated with `(sn=Bergin)` or `(sn=Bergan)`. The Approximate search filter works only with English language strings. It does not work with non-ASCII-based strings, such as Ja or Zn. |

| Search Filter | Operator | Description |
|---|---|---|
| Extensible match | `attr= attr [":dn"] [":" matchingrule] ":=" value Or:[:dn] ":" matchingrule ":=" value` | Return the results entries when an attribute equals the value with the specified matching rule. LDAP version 3 enables you to build match operators and rules for a particular attribute. Matching rules define how to compare attribute values with a particular syntax. In other words, an extensible search filter enables you to add a matching rule to a search filter. For example, the following search filter compares entries containing the surname attribute with value equal to "`Jensen`" by using the matching rule designated by OID 2.5.13.5: `(sn:2.5.13.5:=Jensen)`. Another example illustrates the use of the `":dn"` notation to indicate that the OID `2.5.13.5` should be used when making comparisons, and that the attributes of an entry should be considered part of the entry when evaluating the match: `(sn:dn:2.5.13.5:=Jensen)` |

### 17.4.3.2 Using Compound Search Filters

Multiple search filter components can be combined and evaluated by using the operator:

`(`*`Boolean-Operator`*`(`*`filter`*`)(`*`filter`*`)(`*`filter`*`))`

Boolean operators can be combined and nested together to form complex expressions:

`(`*`Boolean-Operator`*`(`*`filter`*`)(`*`Boolean-operator`*`(`*`filter`*`)(`*`filter`*`)))`

The following table describes the Boolean operators.

| Search Filter | Operator | Description |
|---|---|---|
| AND | `(&(filter)(filter))` | All specified filters must be true for the statement to be true. For example, `(&(sn=Carter)(l=Cupertino))` returns all entries that have the surname attribute equal to "Carter" and the location attribute equal to `Cupertino` if any. |
| OR | `(\|(filter)(filter))` | At least one specified filter must be true for the statement to be true. For example, `(\|(sn=Carter)(l=Cupertino))` returns all entries that have the surname attribute equal to `Carter` or the location attribute equal to `Cupertino` if any. |
| NOT | `(!(filter)(filter))` | The specified filter must not be true for the statement to be true. For example, `(!(sn=Bergin))` returns all entries that do *not* have a surname attribute equal to the string `Smith`. The filter also returns all entries that do not have the `sn` attribute. |

### 17.4.3.3 Using UTF-8 Encoding in Search Filters

UTF8 is a byte-order, variable-length character code for Unicode and a subset of ASCII. You use UTF-8 for multiple-language support by replacing each character of a non 7-bit ASCII character with a byte of a UTF-8 encoding. Typically, you must escape the UTF-8 encoding with a backslash.

For example, the character é has a UTF-8 representation of `c3a9` and è has a UTF-8 representation `c3a8`. A UTF-8 encoding is represented with an escaped backslash. So, é is represented as `\\c3\\a9` and è is represented as `\\c3\\a8`. To represent `cn=Hélène Laurent`, you would use the following encoding:

```
(cn=H\\c3\\a9l\\c3\\a8ne Laurent)
```

#### 17.4.3.4 Using Special Characters in Search Filters

You must specify special characters (for example, a space, backslash, asterisk, comma, period, or others) by using the escape backslash.

- Asterisk. Represent an asterisk (`*`) as `\\2a`. For example, `Five*Star` would be represented as `"(cn=Five\\2aStar)"`.
- Backslash. Represent a backslash (`\`) as `\\5c`. For example, `c:\\file` would be represented as `"(cn=c:\\5c\\5cfile)"`.
- Parentheses. Represent parentheses `()` as `\\28` and `\\29`, respectively. For example, `John Doe (II)` would be represented as `"(cn=John Doe \\28II\\29)"`.
- Null. Represent null as `\\00`. For example, 0001 would be represented as `"(bin=\\00\\00\\00\\01)"`.
- Comma. Represent a comma (`,`) by escaping it as `\\,`. For example, `"(cn=Mkt\\,Peru,dc=example,dc=com)"`.
- Space. Generally, use quotation marks around strings that contain a space. For example, `(cn="HR Managers,ou=Groups,dc=example,dc=com")`.

### 17.4.4 `ldapsearch` Examples

The following examples show the use of the `ldapsearch` command with various search options. These examples all assume that your current working directory is `INSTANCE_DIR/OUD/bin` (`INSTANCE_DIR\OUD\bat` on Windows systems).

The following points pertain to all the examples in this section:

- If the example does not specify a scope (with the `--searchScope` or `-s` option), `ldapsearch` assumes that the scope is `subordinate` or `sub`, which returns the full subtree of the base DN.
- If no attributes are specified, the command returns all attributes and their values.
- If no `--bindDN` and `--bindPassword` are specified, the search uses an anonymous bind.
- If no `--hostname` is specified, the default (`localhost`) is used.

**Note:** Many UNIX and Linux operating systems provide an installed version of common LDAP-client tools, such as `ldapsearch`, `ldapmodify`, and `ldapdelete` in the `/usr/bin` directory. You should use the `ldapsearch` provided with the directory server to search the directory server. You can check which version of `ldapsearch` you are using by typing the following command:

```
$ which ldapsearch
```

If you are using the `ldapsearch` in `/usr/bin`, put `INSTANCE_DIR/OUD/bin` at the beginning of your `$PATH`.

This section contains the following topics:



### 17.4.4.1 To Return All Entries

You can return all entries below a specified branch DN using the presence search filter `(objectclass=*)`. The search filter looks for all entries that have one or more object classes with any value. Because all entries have several object class definitions, the filter guarantees that all entries will be returned.

Run the `ldapsearch` command with the filter `(objectclass=*)`.

```
$ ldapsearch --hostname localhost --port 1389 --baseDN "dc=example,dc=com" \
  "(objectclass=*)"
dn: dc=example,dc=com
objectClass: domain
objectClass: top
dc: example

dn: ou=Groups,dc=example,dc=com
objectClass: organizationalunit
objectClass: top
ou: Groups
```

```
dn: cn=Directory Administrators,ou=Groups,dc=example,dc=com
objectClass: groupofuniquenames
objectClass: top
ou: Groups
cn: Directory Administrators
uniquemember: uid=kvaughan, ou=People, dc=example,dc=com
uniquemember: uid=rdaugherty, ou=People, dc=example,dc=com
uniquemember: uid=hmiller, ou=People, dc=example,dc=com
...
```

#### 17.4.4.2 To Search For a Specific User

You can use an equality filter to locate a specific user in the directory. This example locates an employee with the common name of "Frank Albers".

Run the `ldapsearch` command with the filter `"(cn=Frank Albers)"`.

```
$ ldapsearch --port 1389 --baseDN dc=example,dc=com  "(cn=Frank Albers)"

dn: uid=falbers,ou=People,dc=example,dc=com
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: top
givenName: Frank
uid: falbers
cn: Frank Albers
sn: Albers
telephoneNumber: +1 408 555 3094
userPassword: {SSHA}nDTQJ9DDiMUrBwR0WNKq0tgS4iB2A9QJFgpZiA==
roomNumber: 1439
ou: Accounting
ou: People
l: Sunnyvale
mail: falbers@example.com
facsimileTelephoneNumber: +1 408 555 9751
```

#### 17.4.4.3 To Search for Specific User Attributes

You can use an equality filter to locate an entry's attribute(s) in the directory. Specify one or more attributes by placing them after the search filter. This example locates the `telephoneNumber` and `mail` attributes from the user entry for `Frank Albers`.

Run the `ldapsearch` command with the filter `"(cn=Frank Albers)"` and the corresponding attributes.

```
$ ldapsearch --port 1389 --baseDN dc=example,dc=com \
 "(cn=Frank Albers)" telephoneNumber mail
dn: uid=falbers,ou=People,dc=example,dc=com
telephoneNumber: +1 408 555 3094
mail: falbers@example.com
```

#### 17.4.4.4 To Perform a Search With Base Scope

Together with the search base DN, the scope determines what part of the directory information tree (DIT) is examined. A base scope examines only the level specified by the base DN (and none of its child entries). You specify a base scope by using the `--searchScope base` option or its short form equivalent `-s base`.

Run the `ldapsearch` command with the `--searchScope base` option.

```
$ ldapsearch --hostname localhost --port 1389 --baseDN "dc=example,dc=com" \
  --searchScope base "(objectclass=*)"
dn: dc=example,dc=com
objectClass: domain
objectClass: top
dc: example
```

#### 17.4.4.5 To Perform a Search With One-Level Scope

A one-level scope examines only the level immediately below the base DN. You specify a one-level scope by using the `--searchScope one` option or its short form equivalent `-s one`. This example displays the entries immediately below the base DN.

Run the `ldapsearch` command with the `--searchScope one` option.

```
$ ldapsearch --hostname localhost --port 1389 --baseDN "dc=example,dc=com" \
  --searchScope one "(objectclass=*)"
dn: ou=Groups,dc=example,dc=com
objectClass: top
objectClass: organizationalunit
ou: Groups
dn: ou=People,dc=example,dc=com
objectClass: top
objectClass: organizationalunit
ou: People
dn: ou=Special Users,dc=example,dc=com
objectClass: top
objectClass: organizationalUnit
ou: Special Users
description: Special Administrative Accounts
dn: ou=Company Servers,dc=example,dc=com
objectClass: top
objectClass: organizationalUnit
ou: Company Servers
description: Standard branch for Company Server registration
```

#### 17.4.4.6 To Perform a Search With Subtree Scope

The subtree scope examines the subtree below the base DN and includes the base DN level. You specify a subtree scope using the `--searchScope sub` option, or its short form equivalent `-s sub`. If you do not specify the `--searchScope`, `ldapsearch` assumes a subtree scope.

Run the `ldapsearch` command with the `--searchScope sub` option.

```
$ ldapsearch --hostname localhost --port 1389 \
  --baseDN "cn=Directory Administrators,ou=Groups,dc=example,dc=com" \
  --searchScope sub "(objectclass=*)"
dn: cn=HR Managers,ou=groups,dc=example,dc=com
objectClass: groupOfUniqueNames
objectClass: top
ou: groups
description: People who can manage HR entries
cn: HR Managers
uniqueMember: uid=kvaughan, ou=People, dc=example,dc=com
uniqueMember: uid=cschmith, ou=People, dc=example,dc=com
```

#### 17.4.4.7 To Return Attribute Names Only

The `ldapsearch` command provides a convenient option to check if an attribute is present in the directory. Use the `--typesOnly` option or its short form equivalent `-A` to instruct the directory server to display the attribute names but not their values.

Run the `ldapsearch` command with the `--typesOnly` option.

```
$ ldapsearch --hostname localhost --port 1389 \
  --baseDN "dc=example,dc=com" --typesOnly "(objectclass=*)"
dn: dc=example,dc=com
objectClass
dc
dn: ou=Groups,dc=example,dc=com
objectClass
ou ...
```

#### 17.4.4.8 To Return User Attributes Only

You can use `ldapsearch` to return only user attributes for entries that match the search filter, by including an asterisk `*`. User attributes (as opposed to operational attributes) store user information in the directory. If you do not specify the asterisk, the user attributes are returned by default. You must escape the asterisk appropriately for your shell.

Run the `ldapsearch` command, specifying `'*'` after the search filter.

```
$ ldapsearch --hostname localhost --port 1389 --baseDN "dc=example,dc=com" \
  "(objectclass=*)" '*'
dn: cn=Aggie Aguirre,ou=People,dc=example,dc=com
objectClass: person
objectClass: inetorgperson
objectClass: organizationalperson
objectClass: top
postalAddress: Aggie Aguirre$15172 Jackson Street$Salt Lake City, MI  49843
postalCode: 49843
uid: user.99
description: This is the description for Aggie Aguirre.
employeeNumber: 99
initials: AGA
givenName: Aggie
pager: +1 514 297 1830
mobile: +1 030 300 0720
cn: Aggie Aguirre
telephoneNumber: +1 730 027 2062
sn: Aguirre
street: 15172 Jackson Street
homePhone: +1 229 128 3072
mail: user.99@maildomain.net
l: Salt Lake City
st: MI
```

#### 17.4.4.9 To Return Base DNs Only

You can use `ldapsearch` to return only the base DNs for entries that match the search filter by including a `1.1` string after the search filter.

Run the `ldapsearch` command, specifying `1.1` after the search filter.

```
$ ldapsearch --hostname localhost --port 1389 --baseDN "dc=example,dc=com" \
  "(objectclass=*)" 1.1
version: 1
dn: cn=Richard Arnold,ou=people,dc=example,dc=com
```

```
dn: cn=Kevin Booysen,ou=people,dc=example,dc=com

dn: cn=Steven Morris,ou=people,dc=example,dc=com

dn: cn=Leila Shakir,ou=people,dc=example,dc=com

dn: cn=Emily Smith,ou=people,dc=example,dc=com
...
```

#### 17.4.4.10 To Search For Specific Object Classes

You can search all entries where the attributes are referenced by a specific object class by prepending a `@` character to the object class name. For example, to view all entries that have an object class of `groupOfUniqueNames`, include `@groupOfUniqueNames` after the search filter.

Run the `ldapsearch` command, specifying `@` and the object class after the search filter.

```
$ ldapsearch --hostname localhost --port 1389 \
  --baseDN "ou=Groups,dc=example,dc=com" "(objectclass=*)" @groupOfUniqueNames
dn: ou=Groups,dc=example,dc=com
ou: Groups
objectClass: organizationalunit
objectClass: top
dn: cn=Directory Administrators,ou=Groups,dc=example,dc=com
ou: Groups
objectClass: groupofuniquenames
objectClass: top
cn: Directory Administrators
uniqueMember: uid=kvaughan, ou=People, dc=example,dc=com
uniqueMember: uid=rdaugherty, ou=People, dc=example,dc=com
uniqueMember: uid=hmiller, ou=People, dc=example,dc=com ...
```

#### 17.4.4.11 To Return a Count of All Entries in the Directory

The `ldapsearch` command provides the `--countEntries` to return the total number of entries in the directory. The directory server returns all entries that match the search filter and displays the total number on the last line. This example determines the number of employee entries whose location is Cincinnati.

Run the `ldapsearch` command with the `--countEntries` option.

```
$ ldapsearch --hostname localhost --port 1389 --bindDN "cn=Directory Manager" \
  --bindPassword password --baseDN dc=example,dc=com --countEntries "l=Cincinnati"
dn: cn=Adi Adamski,ou=People,dc=example,dc=com
...
l: Cincinnati
st: OH

dn: Aggi Aguinsky,ou=People,dc=example,dc=com
objectClass: person
...
l: Cincinnati
st: OH

# Total number of matching entries: 2
```

#### 17.4.4.12 To Perform a Search With a Compound Filter

Compound search filters involve multiple tests using the boolean operators AND (`&`), OR (`|`), or NOT (`!`). You can combine and nest boolean operators and filters together to form complex expressions. The following example searches for all entries for employees named `Jensen` who work in `Cupertino`. The command returns two results.

Run the `ldapsearch` command with a compound search filter.

```
$ ldapsearch --hostname localhost --port 1389 --bindDN "cn=Directory Manager" \
  --bindPassword password --baseDN dc=example,dc=com "(&(sn=jensen)(l=Cupertino))"
dn: uid=bjensen,ou=People,dc=example,dc=com
objectClass: person
objectClass: inetOrgPerson
objectClass: top
objectClass: organizationalPerson
ou: Product Development
ou: People
sn: Jensen
...
l: Cupertino
st: CA

dn: uid=rjensen,ou=People,dc=example,dc=com
objectClass: person
objectClass: inetOrgPerson
objectClass: top
objectClass: organizationalPerson
ou: Accounting
ou: People
sn: Jensen
...
l: Cupertino
st: CA
```

#### 17.4.4.13 To Perform a Search Using a Filter File

You can place complex or multiple filters in a file by using the `--filename` option. If the file contains multiple filters, the file should be structured with one filter per line. Searches are performed using the same connection to the directory server in the order in which they appear in the filter file. If the `--filename` option is used, any trailing options are treated as separate attributes. Otherwise, the first trailing option must be the search filter.

This example searches all entries for employees named `Jensen` who work in `Cupertino` and who do not work in the Accounting department.

1. Create the filter file.

   For this example, create a file called `myfilter.txt` with the following content: `(&(sn=jensen)(l=Cupertino)(!(ou=Accounting)))`

2. Run the `ldapsearch` command, specifying the file name as a filter.

   ```
   $ ldapsearch --hostname localhost --port 1389 --bindDN "cn=Directory Manager" \
   --bindPassword password --baseDN dc=example,dc=com --filename myfilter.txt
   dn: uid=bjensen,ou=People,dc=example,dc=com
   objectClass: person
   objectClass: inetOrgPerson
   objectClass: top
   objectClass: organizationalPerson
   ou: Product Development
   ```

```
ou: People
sn: Jensen
l: Cupertino
cn: Barbara Jensen
cn: Babs Jensen
telephoneNumber: +1 408 555 1862
givenName: Barbara
uid: bjensen
mail: bjensen@example.com
```

#### 17.4.4.14 To Limit the Number of Entries Returned in a Search

You can limit the number of entries that are returned by using the `-z` or `--sizeLimit` option. If the number of entries exceeds the number that is specified, the search returns the specified number of entries, then returns an error stating that the size limit was exceeded. The following example requests a maximum of 5 entries.

Run the `ldapsearch` command with the `--sizeLimit` option.

```
$ ldapsearch --hostname localhost --port 1389 -b "dc=example,dc=com" \
  --sizeLimit 5 "objectclass=*" 1.1
dn: dc=example,dc=com

dn: ou=People,dc=example,dc=com

dn: uid=user.0,ou=People,dc=example,dc=com

dn: uid=user.1,ou=People,dc=example,dc=com

dn: uid=user.2,ou=People,dc=example,dc=com

SEARCH operation failed
Result Code:  4 (Size Limit Exceeded)
Additional Information:  This search operation has sent the maximum of 5 entries
to the client
```

### 17.4.5 Searching Data With Oracle Directory Services Manager

The Advanced Search tab of each server instance in ODSM enables you to perform complex searches on directory data, as described in the following section.

#### 17.4.5.1 Perform a Complex LDAP Search

To perform a complex LDAP search by using the ODSM advanced search facility, complete the following steps:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager."
2. Select the **Advanced Search** tab.
3. Select the appropriate network group from the **Network Group** list.
4. In the **Base Search DN** field, enter the DN that will be the starting point of the search.

   To select an entry as Base Search DN, click **Select**.

   In the **Entry Picker** window, select **Tree View** to navigate the directory tree and locate the entry, or **Search View** to search for the entry.

5. Select the scope of the search from the **Scope** list. The LDAP search scope indicates the set of entries at or below the search base DN that will be considered potential matches for a search operation. The scope can be one of:
   - **Base.** This specifies that the search operation should only be performed against the entry specified as the search base DN. No entries below it will be considered.
   - **One Level.** This specifies that the search operation should only be performed against entries that are immediate subordinates of the entry specified as the search base DN. The base entry itself is not included, nor are any entries below the immediate subordinates of the search base entry.
   - **Subtree.** This specifies that the search operation should be performed against the entry specified as the search base and all of its subordinates to any depth.
6. In the **Filter** field, enter a valid LDAP search filter.

   Alternatively, click **Filter Builder** and enter the required information for ODSM to build the LDAP search filter.

   For more information about LDAP search filters, see Section 17.4.3.1, "Specifying Filter Types and Operators."
7. From the **Search Results Size** list, select how you want ODSM to limit the number of entries that are returned by the search.
   - **Set Limit** enables you to specify the precise number of entries that are returned.
   - **Use Virtual List View** enables you to use a virtual list view index in the search. For more information, see Section 17.5.3.16, "Searching Using the Virtual List View Control."
   - **Use Paging** enables you to specify that only a subset of the results should be returned at a time, and allows you to indicate the number of results on each page. For more information, see Section 17.5.3.15, "Searching Using the Simple Paged Results Control."

## 17.5 Using Advanced Search Features

The directory server supports LDAPv3-compliant search functionality by using the `ldapsearch` command. You can use special attributes, security options, and LDAP controls with the search process, based on your system configuration. For additional information, see Section 17.4, "Searching Directory Data," Appendix A.1.2, "Using a Properties File With Server Commands," and Appendix A.4.5, "ldapsearch."

This section contains the following topics:

- Section 17.5.1, "Searching for Special Entries and Attributes"
- Section 17.5.2, "Searching Over SSL"
- Section 17.5.3, "Searching Using Controls"
- Section 17.5.3.16, "Searching Using the Virtual List View Control"
- Section 17.5.4, "Searching in Verbose Mode and With a Properties File"
- Section 17.5.5, "Searching Internationalized Entries"

## 17.5.1 Searching for Special Entries and Attributes

This section describes how to search for operational attributes and how to search the Root DSE entry, and contains the following topics:

- Section 17.5.1.1, "To Search for Operational Attributes"
- Section 17.5.1.2, "To Search the Root DSE Entry"
- Section 17.5.1.3, "To Search for ACI Attributes"
- Section 17.5.1.4, "To Search the Schema Entry"
- Section 17.5.1.5, "To Search the Configuration Entry"
- Section 17.5.1.6, "To Search the Monitoring Entry"

### 17.5.1.1 To Search for Operational Attributes

Operational attributes are used for storing information needed for processing by the directory server itself or for holding any other data maintained by the directory server that was not explicitly provided by clients. Operational attributes are not included in entries returned from search operations unless they are explicitly included in the list of search attributes. You can request the directory server to return operational attributes by adding + (the plus sign) in your `ldapsearch` command.

Run the `ldapsearch` command with the + character.

You must escape the character using a means appropriate to your shell.

```
$ ldapsearch -h localhost -p 1389 -D "cn=Directory Manager" \
  -j pwd-file -b "dc=example,dc=com" "(objectclass=*)" "+"
...
dn: cn=PD Managers,ou=groups,dc=example,dc=com
numSubordinates: 0
hasSubordinates: false
subschemaSubentry: cn=schema
entryDN: cn=pd managers,ou=groups,dc=example,dc=com
entryUUID: 38666d52-7a53-332e-902f-e34dd4aaa7a0
...
```

### 17.5.1.2 To Search the Root DSE Entry

The Root DSE is a special entry that provides information about the server's name, version, naming contexts, and supported features. Because many of the attributes are operational, you must specify + (the plus sign) to display the attributes of the Root DSE entry.

Run the `ldapsearch` command with a baseDN of `""`.

Specify the scope as `base` and include the + character to display operational attributes.

```
$ ldapsearch -h localhost -p 1389 -D "cn=Directory Manager" \
  -j pwd-file -b "" --searchScope base "(objectclass=*)" "+"
dn:
supportedExtension: 1.3.6.1.4.1.4203.1.11.3
supportedExtension: 1.3.6.1.4.1.4203.1.11.1
supportedExtension: 1.3.6.1.4.1.26027.1.6.2
supportedExtension: 1.3.6.1.4.1.26027.1.6.1
supportedExtension: 1.3.6.1.1.8
supportedExtension: 1.3.6.1.4.1.1466.20037
...
```

#### 17.5.1.3 To Search for ACI Attributes

The directory server stores access control instructions (ACIs) as one or more values of the `aci` attribute on an entry to allow or deny access to the directory database. The `aci` attribute is a multi-valued operational attribute that can be read and modified by directory users and that should itself be protected by ACIs. Administrative users are usually given full access to the `aci` attribute and can view its values by running an `ldapsearch` command.

Run the `ldapsearch` command as follows:

```
$ ldapsearch -h localhost -p 1389 -D "cn=Directory Manager" \
  -j pwd-file -b dc=example,dc=com --searchScope base "(aci=*)" aci
dn: dc=example,dc=com
aci: (target ="ldap:///dc=example,dc=com")(targetattr h3.="userPassword")
  (version 3.0;acl "Anonymous read-search access";allow (read, search, compare)
  (userdn = "ldap:///anyone");)
aci: (target="ldap:///dc=example,dc=com") (targetattr = "*")
  (version 3.0; acl "allow all Admin group"; allow(all)
  groupdn = "ldap:///cn=Directory Administrators,ou=Groups,dc=example,dc=com";)
```

#### 17.5.1.4 To Search the Schema Entry

The directory server holds schema information in the schema entry (`cn=schema`) for the object classes and attributes defined on your instance.

Run the `ldapsearch` command on the `cn=schema` base DN.

Because the attributes in the schema are operational attributes, you must include `"+"` at the end of your search.

```
$ ldapsearch -h localhost -p 1389 -D "cn=Directory Manager" \
  -j pwd-file -b cn=schema --searchScope base "(objectclass=*)" "+"
dn: cn=schema
nameForms: ( 1.3.6.1.1.10.15.1 NAME 'uddiBusinessEntityNameForm' OC uddiBusiness
 Entity MUST ( uddiBusinessKey ) X-ORIGIN 'RFC 4403' )
nameForms: ( 1.3.6.1.1.10.15.2 NAME 'uddiContactNameForm' OC uddiContact MUST
 (uddiUUID ) X-ORIGIN 'RFC 4403' )
nameForms: ( 1.3.6.1.1.10.15.3 NAME 'uddiAddressNameForm' OC uddiAddress MUST
 (uddiUUID ) X-ORIGIN 'RFC 4403' )
...
attributeTypes: ( 1.3.6.1.1.1.1.12 NAME 'memberUid' EQUALITY caseExactIA5Match
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 X-ORIGIN 'draft-howard-rfc2307bis' )
attributeTypes: ( 1.3.6.1.1.1.1.13 NAME 'memberNisNetgroup' EQUALITY caseExactIA
 5Match SUBSTR caseExactIA5SubstringsMatch SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
 X-ORIGIN 'draft-howard-rfc2307bis' )
attributeTypes: ( 1.3.6.1.1.1.1.14 NAME 'nisNetgroupTriple' DESC 'Netgroup
 triple' EQUALITY caseIgnoreIA5Match SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 X-ORIGIN
 'draft-howard-rfc2307bis' )
...
```

#### 17.5.1.5 To Search the Configuration Entry

The directory server stores its configuration under the `cn=config` entry. Direct access to this entry over LDAP is not advised. The configuration is accessible and modifiable by using the `dsconfig` command. The `dsconfig` command connects to the directory server over SSL via the administration connector. For more information, see Section 14.3, "Managing Administration Traffic to the Server."

To search the configuration entry using `dsconfig` in interactive mode, run the command as follows:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file
```

For more information about accessing the server configuration by using `dsconfig`, see Section 14.1, "Managing the Server Configuration With `dsconfig`."

#### 17.5.1.6 To Search the Monitoring Entry

The directory server monitor entry `cn=monitor` provides statistical information about the server performance, state, and version. You can access this information by using the `ldapsearch` command.

Although you can access `cn=monitor` using any configured LDAP connection handler, it is recommended that you use the administration connector for all access to administrative suffixes. Using the administration connector ensures that monitoring data is not polluted and that server administration takes precedence over user traffic. To use the administration connector, specify the administration port, and include the `--useSSL` option. For more information, see Section 14.3, "Managing Administration Traffic to the Server."

Run the `ldapsearch` command on the base DN `cn=monitor`.

```
$ ldapsearch -h localhost -p 4444 --useSSL -D "cn=Directory Manager" \
  -j pwd-file -b cn=monitor "(objectclass=*)"
dn: cn=monitor
startTime: 20120119135658Z
objectClass: extensibleObject
objectClass: top
objectClass: ds-monitor-entry
cn: monitor
vendorName: Oracle Corporation
currentTime: 20120125145650Z
vendorVersion: Oracle Unified Directory 11.1.2.0.0
maxConnections: 3
productName: Oracle Unified Directory
currentConnections: 1
totalConnections: 22
upTime: 6 days 0 hours 59 minutes 52 seconds
...
```

### 17.5.2 Searching Over SSL

If you have configured the directory server to accept SSL connections by using a self-signed certificate or certificate, you can search using client authentication. The following procedures show how to search the directory over SSL using various authentication mechanisms.

This section contains the following topics:

- Section 17.5.2.1, "To Search Over SSL With Blind Trust"
- Section 17.5.2.2, "To Search Over SSL Using a Trust Store"
- Section 17.5.2.3, "To Search Over SSL With No Trust Store"
- Section 17.5.2.4, "To Search Over SSL Using a Keystore"
- Section 17.5.2.5, "To Search Using StartTLS"
- Section 17.5.2.6, "To Search Using SASL With DIGEST-MD5 Client Authentication"
- Section 17.5.2.7, "To Search Using SASL With the GSSAPI Mechanism"
- Section 17.5.2.8, "To Search Using SASL With the PLAIN Mechanism"

#### 17.5.2.1 To Search Over SSL With Blind Trust

You can configure the client to automatically trust any certificate that the server presents to it. However, this method is not secure and is vulnerable to man-in-the-middle attacks. Generally, you should use this type of authentication for testing purposes only.

Run the `ldapsearch` command with the `--trustAll` option.

The following command searches the Root DSE.

```
$ ldapsearch -h localhost -p 1636 --useSSL --trustAll -b "" \
  --searchScope base "(objectClass=*)"
```

#### 17.5.2.2 To Search Over SSL Using a Trust Store

You can configure the client to use a certificate trust store, which contains information about the certificates it can trust. The client can check any server certificate to those listed in its trust store. If the client finds a match, a secure communication can take place with the server. If no match is found, the server cannot be trusted. You must ensure that the presented certificate is valid and add it to the trust store, which then allows secure communication.

Run the `ldapsearch` command with the `--trustStorePath` option.

The following command searches the Root DSE using a trust store.

```
$ ldapsearch -h localhost -p 1636 --useSSL \
  --trustStorePath /home/scarter/security/cert.db -b "" \
  --searchScope base "(objectClass=*)"
```

#### 17.5.2.3 To Search Over SSL With No Trust Store

If no trust store is specified, you are prompted as to whether the certificate that was presented to the client should be trusted.

Run the `ldapsearch` command without the `--trustStorePath` option.

The following command searches the Root DSE without using a trust store.

```
$ ldapsearch -h localhost -p 1636 --useSSL -b "" \
  --searchScope base "(objectclass=*)"
The server is using the following certificate:
Subject DN: CN=example.com, O=Example Corp, C=US
Issuer DN: CN=example.com, O=Example Corp, C=US
Validity: Fri Mar 02 16:48:17 CST 2007 through Thu May 31 17:48:17 CDT 2007
Do you wish to trust this certificate and continue connecting to the server?
Please enter "yes" or "no": yes

dn: objectClass: ds-rootDSE
objectClass: top
```

#### 17.5.2.4 To Search Over SSL Using a Keystore

If the client is required to present its own certificate to the directory server, that client must know which certificate keystore to use. The client can determine the certificate keystore by specifying the `--keyStorePath` option with either the `--keyStorePassword` or `--keyStorePasswordFile`. This scenario typically occurs when the client performs a SASL EXTERNAL authentication or if the server always requires the client to present its own certificates.

Run the `ldapsearch` command with the `--keyStore...` options.

The following command searches the Root DSE using a trust store and a key store.

```
$ ldapsearch -h localhost -p 1636 --useSSL \
  --keyStorePath /home/scarter/security/key.db \
  --keyStorePasswordFile /home/keystore.pin \
  --trustStorePath /home/scarter/security/cert.db --useSASLExternal -b "" \
  --searchScope base "(objectClass=*)"
```

#### 17.5.2.5 To Search Using StartTLS

The process for using StartTLS with the `ldapsearch` utility is very similar to the process for using SSL. However, you must do the following:

- Use the port on which the server is listening for *unencrypted* LDAP requests
- Indicate that StartTLS should be used instead of SSL (that is, use the `--startTLS` option instead of the `--useSSL` option).

Run the `ldapsearch` command with the `--startTLS` option.

The following command searches the Root DSE using startTLS.

```
$ ldapsearch -h localhost -p 1389 --startTLS \
  -b "" --searchScope base "(objectClass=*)"
```

#### 17.5.2.6 To Search Using SASL With DIGEST-MD5 Client Authentication

The directory server supports a number of Simple Authentication and Security Layer (SASL) mechanisms. DIGEST-MD5 is one form of SASL authentication to the server that does not expose the clear-text password.

Run the `ldapsearch` command with the appropriate `--saslOption` options.

The `authid` option specifies the identity of the user that is authenticating to the server. The option can be in the form of a `dn` (for example, `dn:uid=scarter,dc=example,dc-com`) or a user name (for example, `authid=u:sam.carter`). The attribute can be used to indicate that the search operation should be performed under the authority of another user after authentication. The `realm` specifies the fully qualified name of the server host machine and is optional.

This example searches the Root DSE.

```
$ ldapsearch -h localhost -p 1636 --useSSL \
  --trustStorePath /home/cert.db --certNickName "my-cert" -w - \
  --saslOption mech=DIGEST-MD5 --saslOption realm="example.com" \
  --saslOption authid="dn:uid=scarter,dc=example,dc=com" -b "" "(objectclass=*)"
```

#### 17.5.2.7 To Search Using SASL With the GSSAPI Mechanism

The GSSAPI mechanism performs authentication in a Kerberos environment and requires that the client system be configured to participate in such an environment.

Run the `ldapsearch` command to search as a user who already has a valid Kerberos session.

The `authid` attribute specifies the authentication ID that should be used to identify the user.

This example searches the Root DSE.

```
$ ldapsearch -h localhost -p 1389 --saslOption mech=GSSAPI \
  --saslOption authid="dn:uid=scarter,dc=example,dc=com" \
  --searchScope "" -b "" "(objectclass=*)"
```

#### 17.5.2.8 To Search Using SASL With the PLAIN Mechanism

The PLAIN mechanism performs authentication in a manner similar to LDAP simple authentication except that the user is identified in the form of an authorization ID rather than a full DN.

Run the `ldapsearch` command to search as a user who already has a valid Kerberos session.

The `authid` attribute specifies the authentication ID that should be used to identify the user.

This example searches the Root DSE.

```
$ ldapsearch -h localhost -p 1389 \
  --saslOption mech=PLAIN --saslOption authid="dn:uid=scarter,dc=example,dc=com" \
  --searchScope "" -b "" "(objectclass=*)"
```

### 17.5.3 Searching Using Controls

LDAP controls extend the functionality of LDAP commands, such as `ldapsearch`, to carry out additional operations on top of the search. Each control is defined as an object identifier (OID) that uniquely identifies the control, a criticality flag, and any associated values. If the client sets the criticality flag when sending the control to the directory server, the directory server must either perform the operation with the control or not process it. If the flag is not set by the client, the directory server is free to ignore the control if it cannot process it.

You can use multiple controls in a single operation, such as the virtual list view with server-side sorting. The virtual list view control requires additional explanation and is therefore described in its own section.

This section contains the following topics:

#### 17.5.3.1 Viewing the Available Controls

You can view the current list of controls for your directory server by searching the Root DSE entry for the `supportedControl` attribute.

Run the `ldapsearch` command on the Root DSE entry.

```
$ ldapsearch -h localhost -p 1389 -b "" --searchScope base "(objectclass=*)" \
  supportedControl
dn:
supportedControl: 1.2.826.0.1.3344810.2.3
supportedControl: 1.2.840.113556.1.4.1413
supportedControl: 1.2.840.113556.1.4.319
supportedControl: 1.2.840.113556.1.4.473
supportedControl: 1.2.840.113556.1.4.805
supportedControl: 1.3.6.1.1.12
supportedControl: 1.3.6.1.1.13.1
supportedControl: 1.3.6.1.1.13.2
supportedControl: 1.3.6.1.4.1.26027.1.5.2
supportedControl: 1.3.6.1.4.1.26027.1.5.5
supportedControl: 1.3.6.1.4.1.26027.1.5.6
supportedControl: 1.3.6.1.4.1.26027.2.3.1
supportedControl: 1.3.6.1.4.1.26027.2.3.2
supportedControl: 1.3.6.1.4.1.42.2.27.8.5.1
supportedControl: 1.3.6.1.4.1.42.2.27.9.5.2
supportedControl: 1.3.6.1.4.1.42.2.27.9.5.8
supportedControl: 1.3.6.1.4.1.4203.1.10.1
supportedControl: 1.3.6.1.4.1.4203.1.10.2
supportedControl: 2.16.840.1.113730.3.4.12
supportedControl: 2.16.840.1.113730.3.4.16
supportedControl: 2.16.840.1.113730.3.4.17
supportedControl: 2.16.840.1.113730.3.4.18
supportedControl: 2.16.840.1.113730.3.4.19
supportedControl: 2.16.840.1.113730.3.4.2
supportedControl: 2.16.840.1.113730.3.4.3
supportedControl: 2.16.840.1.113730.3.4.4
supportedControl: 2.16.840.1.113730.3.4.5
supportedControl: 2.16.840.1.113730.3.4.9
supportedControl: 2.16.840.1.113894.1.8.21
supportedControl: 2.16.840.1.113894.1.8.31
```

The controls are returned as a list of OIDs. See the following table for a description of the control that corresponds to each OID. Note that not all of these controls can be used with the `ldapsearch` command.

| OID | Control |
|---|---|
| `1.2.826.0.1.3344810.2.3` | Matched Values Control |
| 1.2.840.113556.1.4.1413 | LDAP Ease Modify Restrictions Control |
| `1.2.840.113556.1.4.319` | Simple Paged Results Control |
| `1.2.840.113556.1.4.473` | Server-Side Sort Control |
| `1.2.840.113556.1.4.805` | Subtree Delete Control |
| `1.3.6.1.1.12` | LDAP Assertion Control |
| `1.3.6.1.1.13.1` | LDAP Pre-Read Control |
| `1.3.6.1.1.13.2` | LDAP Post-Read Control |
| `1.3.6.1.4.1.26027.1.5.2` | Replication Repair Control |

| OID | Control |
|---|---|
| 1.3.6.1.4.1.26027.1.5.5 | Network Group Selection Control |
| 1.3.6.1.4.1.26027.1.5.6 | Network Group Query Control |
| 1.3.6.1.4.1.26027.2.3.1 | Join Search Control |
| 1.3.6.1.4.1.26027.2.3.2 | Proximity Search Control |
| 1.3.6.1.4.1.42.2.27.8.5.1 | Password Policy control |
| 1.3.6.1.4.1.42.2.27.9.5.2 | Get Effective Rights Control |
| 1.3.6.1.4.1.42.2.27.9.5.8 | Account Usability Request Control |
| 1.3.6.1.4.1.4203.1.10.2 | LDAP No-Op Control |
| 1.3.6.1.4.1.4203.1.10.1 | LDAP Subentry Request Control |
| 2.16.840.1.113730.3.4.12 | Proxied Authorization v1 Control |
| 2.16.840.1.113730.3.4.16 | Authorization Identity Control |
| 2.16.840.1.113730.3.4.17 | Real Attributes Only Control |
| 2.16.840.1.113730.3.4.18 | Proxied Authorization v2 Control |
| 2.16.840.1.113730.3.4.19 | Virtual Attributes Only Control |
| 2.16.840.1.113730.3.4.2 | Manage DSA IT Control |
| 2.16.840.1.113730.3.4.3 | Persistent Search Control |
| 2.16.840.1.113730.3.4.4 | Netscape Password Expired LDAPv3 Control |
| 2.16.840.1.113730.3.4.5 | Netscape Password Expiring LDAPv3 Control |
| 2.16.840.1.113730.3.4.9 | Virtual List View Control |
| 2.16.840.1.113894.1.8.21 | Search Count Request Control |
| 2.16.840.1.113894.1.8.31 | ECID Execution Info control |

#### 17.5.3.2 Searching Using the Join Search Control

The Join Search Control retrieves related entry tree chains such as friends, managers, and so forth, in a single search operation. The Join Control can only target entry chains with established relationships that can (but do not have to) be cross referenced.

For example, the following entry is part of an established "friends" relationship hierarchy where each participating entry has links to other participating entries. In this case these links are formed by the `friend` attribute.

```
dn: uid=user.3,ou=People,dc=example,dc=com
objectClass: person
objectClass: organizationalperson
objectClass: inetorgperson
objectClass: top
uid: user.3
cn: Kenny McCormick
sn: McCormick
friend: uid=user.0,ou=People,dc=example,dc=com
friend: uid=user.1,ou=People,dc=example,dc=com
friend: uid=user.2,ou=People,dc=example,dc=com
...
```

In a search operation with the Join Control, the search parameters such as scope and filter apply to the join search, that is, to entries evaluated during the join. This means

that only matching results are returned. This functionality enables you to retrieve the entire linked relationship hierarchy, or a subset of it, in a single search operation, based on specific search criteria and scope.

You can specify the Proximity Search Control with the `ldapsearch` command by using the `--control` or `-J` option with the Proximity Search Control OID (1.3.6.1.4.1.26027.2.3.1) as follows:

```
OID:criticality:attribute
```

where *attribute* is the attribute on which the relationship between entries is based.

The following example requests the subset of user entries that are linked through the `friend` attribute.

```
$ ldapsearch -h localhost -p 1389 -D "cn=directory manager" -j pwd-file \
  --baseDN "uid=user.3,ou=People,dc=example,dc=com" \
  --searchScope sub \
  -J "1.3.6.1.4.1.26027.2.3.1:true:friend" \
  "(objectClass=person)"
```

In a join search, the search parameters have the following significance:

- `baseDN`

  The search base is used to specify the precise entry from which to start the join search.

- `searchScope`

  The search scope is used to specify distinct levels of join depth.

  - A search scope of `base` retrieves only direct relationships, for example, direct friends that are specified by the `friend` attribute in the sample entry.
  - A search scope of `one` goes one level deep, retrieving direct friends of direct friends of the sample entry.
  - A search scope of `sub` traverses the entire hierarchy chain no matter how many levels.
  - A search scope of `subordinate` has the same effect as `sub`, but does not include the base entry in search results.

- `filter`

  The search filter is used to evaluate candidate entries during the join for inclusion in the search results. The filter can be used to refine the search to include only specific entries. It works in exactly the same way as the filter for standard search operations but is applicable only to join search results.

### 17.5.3.3 Searching Using the Proximity Search Control

The Proximity Search Control provides base location data to the server in the search request. This enables the server to generate proximity virtual attribute values for all candidate entries that include location data. The value of the `location` attribute in an entry is the latitude-longitude GPS coordinates, in WGS84 standard format. User applications can periodically update the value of this attribute with the last known location of the user. For example, the following entry extract shows an entry whose location has been updated to the coordinates of Golden Gate Bridge:

```
dn: uid=user.1,ou=People,dc=example,dc=com
objectClass: geoObject
objectClass: person
```

```
objectClass: organizationalperson
objectClass: inetorgperson
objectClass: top
objectClass: geoObject
uid: user.1
cn: Bob Smith
sn: Smith
location: 37.81997, -122.47859
...
```

The server can calculate the location proximity of each entry to the base location provided in the Proximity Search Control.

A client application can therefore request a proximity value to be calculated and returned for each matching search result entry. The client application can use the proximity attribute in the search filter of the search operation itself and can therefore request matching search result entries based on their proximity to a given base location.

You can specify the Proximity Search Control with the `ldapsearch` command by using the `--control` or `-J` option with the Proximity Search Control OID (1.3.6.1.4.1.26027.2.3.2) as follows:

`OID:`*`criticality`*`:`*`location`*

where *location* represents the latitude-longitude GPS coordinates in WGS84 standard format.

The following example sets the base location to the coordinates of the Eiffel Tower (48.858844, 2.294351) and requests all user entries whose location is within 500 meters of the base location.

```
$ ldapsearch -h localhost -p 1389 -D "cn=directory manager" -j pwd-file \
  -b "dc=example,dc=com" --searchScope sub \
  -J "1.3.6.1.4.1.26027.2.3.2:true:48.858844,2.294351" \
  "(&(objectClass=person)(proximity<=500))"
```

### 17.5.3.4 Searching Using the Account Usability Request Control

The Account Usability Request Control determines if a user account can be used to authenticate to a server. If the user account is available, the control adds a message before any entry about whether the account is usable.

You can specify the Account Usability Request Control with `ldapsearch` in the following ways:

- **OID.** Use the `--control` or `-J` option with the Account Usability Request Control OID: `1.3.6.1.4.1.42.2.27.9.5.8` with no value.
- **Named constant.** Use a named constant, `accountusable` or `accountusability`, with the `--control` or `-J` option, instead of using the Account Usability Request Control OID. For example, use `-J accountusable` or `-J accountusability` with the `ldapsearch` command.

Use the `ldapsearch` command with the `--control` option or its short form `-J`.

```
$ ldapsearch -h localhost -p 1389 -b "dc=example,dc=com" \
  --searchScope sub -J "accountusability:true" "(objectclass=*)"
# Account Usability Response Control
# The account is usable
dn: dc=example,dc=com
objectClass: domain
objectClass: top
```

```
dc: example
...
```

#### 17.5.3.5 Searching Using the Authorization Identity Request Control

The Authorization Identity Request Control allows the client to obtain the authorization identity for the client connection during the LDAP bind request. The authorization ID returned by the server is displayed to the client as soon as authentication has completed. The line containing the authorization ID is prefixed with a # character, making it a comment if the output is to be interpreted as an LDIF.

You can specify the Authorization Identity Request Control with `ldapsearch` in a number of ways:

- **OID.** Use the `--control` or `-J` option with the Authorization Identity Request Control OID: `2.16.840.1.113730.3.4.16` with no value.
- **Named constant.** Use a named constant, `authzid` or `authorizationidentity` with the `-control` or `-J` option instead of using the Authorization Identity Request Control OID. For example, use `-J authzid` or `-J authorizationidentity` with the `ldapsearch` command.

Use the `ldapsearch` command with the `--reportAuthzID` option.

```
$ ldapsearch -h localhost -p 1389 -D "cn=Directory Manager" \
  -j pwd-file -b dc=example,dc=com --searchScope base \
  --reportAuthzID "(objectclass=*)"
# Bound with authorization ID dn:cn=Directory Manager,cn=Root DNs,cn=config
dn: dc=example,dc=com
objectClass: domain
objectClass: top
dc: example
```

#### 17.5.3.6 Searching Using the Get Effective Rights Control

The Get Effective Rights Control enables you to evaluate existing or new ACIs and to see the effective rights that they grant for a user on a specified entry.

The response to this control is to return the effective rights information about the entries and attributes in the search results. This extra information includes read and write permissions for each entry and for each attribute in each entry. The permissions can be requested for the bind DN used for the search or for an arbitrary DN, allowing administrators to test the permissions of directory users.

The `ldapsearch` command provides two ways to use the Get Effective Rights Control:

- Use `-J effectiverights` or the OID `-J "1.3.6.1.4.1.42.2.27.9.5.2"`. The request only takes an authorization ID (`authzid`). If you specify a NULL value for the authorization ID (`authzid`), the bind user is used as the `authzid`.
- Use `-g dn:"dn"`. The command option shows the effective rights of the user binding with the given DN. You can use this option together with the `-e` option to include the effective rights on the named attributes. You can use the option to determine if a user has permission to add an attribute that does not currently exist in an entry.

> **Note:** You cannot use the `-g` option with the `-J` option.

To view effective rights, you should specify the virtual attributes `aclRights` and `aclRightsInfo`, which are generated by the server in response to the effective rights request. Thus, you should not use these attributes in search commands of any kind.

1. Use the `ldapsearch` command to display the effective rights of all users.

```
$ ldapsearch -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
  -b dc=example,dc=com -J effectiverights "(objectclass=*)" aclRights

dn: dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0

dn: ou=Groups, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0

dn: ou=People, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0

dn: cn=Accounting Managers,ou=groups,dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0

dn: cn=HR Managers,ou=groups,dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0

...
```

2. Use the `ldapsearch` command to display the effective rights of a specific user.

   This example uses the `--getEffectiveRightsAuthzid` option. You can also use the `--control` or `-J` option, such as `-J geteffectiverights`.

```
$ ldapsearch -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
  -b dc=example,dc=com \
  --getEffectiveRightsAuthzid "dn:uid=scarter,ou=People,dc=example,dc=com" \
  "(uid=scarter)" aclRights
dn: uid=scarter,ou=People,dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:1,proxy:0
```

3. Use the `ldapsearch` command to display effective rights information for a specific user.

   The `aclRightsInfo` attribute provides more detailed logging information that explains how effective rights are granted or denied.

```
ldapsearch -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
  -b dc=example,dc=com \
  --getEffectiveRightsAuthzid "dn:uid=scarter,ou=People,dc=example,dc=com"\
  "(uid=scarter)" aclRightsInfo

dn: uid=scarter,ou=People,dc=example,dc=com
aclRightsInfo;logs;entryLevel;add: acl_summary(main): access not allowed(add)
on
entry/attr(uid=scarter,ou=People,dc=example,dc=com, NULL) to
 (uid=scarter,ou=People,dc=example,dc=com)
(not proxied) ( reason: no acis matched the subject )
aclRightsInfo;logs;entryLevel;proxy: acl_summary(main): access not
allowed(proxy ) on
entry/attr(uid=scarter,ou=People,dc=example,dc=com, NULL) to
 (uid=scarter, ou=People,dc=example,dc=com)
(not proxied) ( reason: no acis matched the subject )
aclRightsInfo;logs;entryLevel;write: acl_summary(main): access allowed(write)
on
```

```
entry/attr(uid=scarter,ou=People,dc=example,dc=com, NULL) to
 (uid=scarter,ou=People,dc=example,dc=com)
(not proxied) ( reason: evaluated allow , deciding_aci : Allow self entry
modification)
aclRightsInfo;logs;entryLevel;read: acl_summary(main): access allowed(read) on
entry/attr(uid=scarter,ou=People,dc=example,dc=com, NULL) to
 (uid=scarter,ou=People,dc=example,dc=com)
(not proxied) ( reason: evaluated allow , deciding_aci: Anonymous extended
 operation access)
aclRightsInfo;logs;entryLevel;delete: acl_summary(main): access not
allowed(delete) on
entry/attr(uid=scarter,ou=People,dc=example,dc=com, NULL) to
 (uid=scarter,ou=People,dc=example,dc=com)
(not proxied) ( reason: no acis matched the subject )
```

### 17.5.3.7 Searching Using the LDAP Assertion Control

The LDAP Assertion Control allows you to specify a condition that must evaluate to true for the searching operation to process. The value of the control should be in the form of an LDAP search filter. The server tests the base object before searching for entries that match the search scope and filter. If the assertion fails, no entries are returned.

This example determines first if the assertion is met, and returns the entry if it matches the search filter.

Run the `ldapsearch` command with the `--assertionFilter` option using the assertion `(objectclass=top)`.

```
$ ldapsearch -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
  -b "cn=HR Managers,ou=Groups,dc=example,dc=com" \ -s sub \
  --assertionFilter "(objectclass=top)" "(objectclass=*)"
dn: cn=HR Managers,ou=groups,dc=example,dc=com
objectClass: groupOfUniqueNames
objectClass: top
ou: groups
description: People who can manage HR entries
uniqueMember: uid=kvaughan, ou=People, dc=example,dc=com
uniqueMember: uid=cschmith, ou=People, dc=example,dc=com
cn: HR Managers
```

### 17.5.3.8 Searching Using the LDAP Subentry Control

The LDAP Subentry Control allows the client to request that the server return only entries with the `ldapSubEntry` object class during a search operation. LDAP subentries are *operational objects*, similar to operational attributes, that are returned only if they are explicitly requested. Typically, you can use the control when searching the schema.

You request the server to return subentries with `ldapsearch` in the following ways:

- Using the `--subEntries` option to specify the LDAP Subentry Control.
- Specifying base search scope to retrieve a specific subentry if its base DN is known.
- Using the equality filter, `(objectclass=ldapSubentry)`.

> **Note:** Using the equality filter is not part of the standard and is supported for backward compatibility only.

Run the `ldapsearch` command with the `--subEntries` option, as follows:

```
$ ldapsearch -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
  -b "cn=schema" --subEntries "(objectclass=*)"
```

### 17.5.3.9 Searching Using the Manage DSA IT Control

The Manage DSA IT Control allows the client to request that the server treat smart referrals as regular entries during the search. A *smart referral* is an entry that references another server or location in the directory information tree `DIT` and contains the `referral` object class with one or more attributes containing the LDAP URLs that specify the referral.

You can specify the Manage DSA IT Control with `ldapsearch` in a number of ways:

- **OID**. Use the `--control` or `-J` option with the Manage DSA IT Control OID: `2.16.840.1.113730.3.4.2` with no value.
- **Named constant**. Use the named constant, `managedsait` with the `--control` or `-J` option instead of the Manage DSA IT Control OID. For example, use `-J managedsait` with the `ldapsearch` command.

To use the Manage DSA IT control in a search, run the `ldapsearch` command with the `-J` option, as follows:

```
$ ldapsearch -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
  -b dc=example,dc=com -J managedsait "(uid=president)" ref
dn: uid=president,ou=People,dc=example,dc=com
ref: ldap://example.com:389/dc=example,dc=com??sub?(uid=bjensen)
```

> **Note:** Without the `-J managedsait` argument, the command returns the referred entry.

### 17.5.3.10 Searching Using the Matched Values Filter Control

The Matched Values Filter Control allows clients to request a subset of attribute values from an entry that evaluate to TRUE. This control allows the user to selectively read a subset of attribute values without retrieving all values, and then scan for the desired set locally.

Run the `ldapsearch` command with the `--matchedValuesFilter` option.

```
$ ldapsearch -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
  -b ou=groups,dc=example,dc=com --matchedValuesFilter
"(uniquemember=uid=kvaughan*)"
  "(objectclass=*)"
dn: ou=Groups,dc=example,dc=com
dn: cn=Directory Administrators,ou=Groups,dc=example,dc=com
uniqueMember: uid=kvaughan, ou=People, dc=example,dc=com
dn: cn=Accounting Managers,ou=groups,dc=example,dc=com
dn: cn=HR Managers,ou=groups,dc=example,dc=com
uniqueMember: uid=kvaughan, ou=People, dc=example,dc=com
dn: cn=QA Managers,ou=groups,dc=example,dc=com
dn: cn=PD Managers,ou=groups,dc=example,dc=com
```

### 17.5.3.11 Searching Using the Password Policy Control

The Password Policy Control allows a client to request information about the current password policy information for a user entry.

You can specify the Password Policy Control with `ldapsearch` in a number of ways:

- **OID**. Use the `--control` or `-J` option with the Password Policy Control OID: `1.3.6.1.4.1.42.2.27.8.5.1` with no value.
- **Named constant**. Use the named constants, `pwpolicy` or `passwordpolicy` with the `--control` or `-J` option instead of the Password Policy Control OID. For example, use `-J pwpolicy` or `-J passwordpolicy` with `ldapsearch`.
- **Option**. Use the `--usePasswordPolicyControl` option.

> **Note:** The `-J` or `--control` option is used to specify which controls to use in a *search* request. The `--usePasswordPolicyControl` option is used for *bind* requests.

Run the `ldapsearch` command with the `--usePasswordPolicyControl` option.

```
$ ldapsearch -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
  -b dc=example,dc=com -s base --usePasswordPolicyControl "(objectclass=*)"
```

### 17.5.3.12 Searching Using the Persistent Search Control

The Persistent Search Control allows a client to receive notification when entries in the directory are changed by an add, delete, or modify operation. When a change occurs, the server sends the updated entry to the client if the entry matches the search criteria that was used by the Entry Change Notification Control.

The `ldapsearch` command provides an option to run a persistent search (`-C`) that keeps the connection open and displays the entries that match the scope and filter whenever any changes (add, delete, modify, or all) occur. You can quit the search by pressing `Control-C`.

The value for this argument must be in the form:
`ps[[:''changetype''[[:''changesonly''[[:''entrychangecontrols'']]]]]`

The elements of this value include the following:

- `ps` — Required operator.
- `changetype` — Indicates the types of changes for which the client wants to receive notification. This element can be any of `add`, `del`, `mod`, or `moddn`, or it can be `all` to register for all change types. It can also be a comma-separated list to register for multiple specific change types. If this element is not provided, it defaults to including `all` change types.
- `changesonly` — If `True`, the client should only be notified of changes that occur to matching entries after the search is registered. If `False`, the server should also send all existing entries in the server that match the provided search criteria. If this element is not provided, then it will default to only returning entries for updates that have occurred since the search was registered.
- `entrychangecontrols` — If `True`, the server should include the Entry Change Notification Control in entries sent to the client as a result of changes. If `False`, the Entry Change Notification Control should not be included. If this element is not provided, then it will default to including the Entry Change Notification Controls.

1. Run the `ldapsearch` command as follows:

```
$ ldapsearch -h localhost -p 1389 -D "cn=admin,cn=Administrators,cn=config" \
  -j pwd-file -b dc=example,dc=com --persistentSearch ps:add:true:true \
  "(objectclass=*)"
```

**Note:** When you use this command, the server waits for any changes made using `add`, `delete`, `modify` or `all` to return values.

2. Open another terminal window and use `ldapmodify` to add a new entry.

```
$ ldapmodify -h localhost -p 1389 -b dc=example,dc=com \
  --defaultAdd --filename new_add.ldif
Processing ADD request for uid=Marcia Garza,ou=People,dc=example,dc=com
ADD operation successful for DN uid=Marcia Garza,ou=People,dc=example,dc=com
```

3. The original terminal window shows the change.

   To end the session, press Control-Z (Unix/Linux) or Control-C (Windows).

```
# Persistent search change type:  add
dn: uid=Marcia Garza,ou=People,dc=example,dc=com
objectClass: person
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: top
givenName: Marcia
uid: mgarza
uid: Marcia Garza
cn: Marcia Garza
sn: Garza
userpassword: {SSHA}SNfL1RUm5uvTnLK+G0K3oz+Peb1i5/+YsylfBg==
roomnumber: 5484
l: Santa Clara
ou: Accounting
ou: People
mail: mgarza@example.com
```

4. To terminate the session, press Control-D (Unix/Linux) or Control-C (Windows), and then type `Y` to quit.

```
Terminate batch job (Y/N)?
```

### 17.5.3.13 Searching Using the Proxied Authorization Control

The Proxied Authorization Control allows a client to impersonate another entry for a specific operation. This control can be useful in trusted applications that need to perform on behalf of many different users, so that the application does not need to re-authenticate for each operation.

Run the `ldapsearch` command with the `--proxyAs` option, as follows:

Here, `clientApp` must have the appropriate ACI permissions within the subtree to use the Proxied Authorization Control. If not granted, LDAP error 50 `insufficient access rights` will be returned to the client.

```
$ ldapsearch -h localhost -p 1389 \
  -D "uid=clientApp,ou=Applications,dc=example,dc=com" -j pwd-file \
  -s sub -b dc=example,dc=com \
  --proxyAs "dn:uid=acctgAdmin,ou=Administrators,ou=People,dc=example,dc=com" \
  "(uid=kvaughan)" mail
```

### 17.5.3.14 Searching Using the Server-Side Sort Control

The Server-Side Sort Control allows the client to request that the server sort the search results before sending them to the client. This is convenient when the server has indexes that can satisfy the sort order requested by the client faster than the client can.

You can sort the number of entries returned by using the `--sortOrder` option. If you do not specify + (a plus sign) for ascending or - (a minus sign) for descending, then the default option is to sort in ascending order.

1. Use the `ldapsearch` command to search all entries and to display the results in ascending order.

   Use the `--sortOrder` option sorted on the attributes `sn` and `givenName`.

   ```
   $ ldapsearch -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
     --s sub -b dc=example,dc=com --sortorder sn,givenName "(objectclass)"
   dn: uid=dakers,ou=People,dc=example,dc=com
   objectClass: person
   objectClass: organizationalPerson
   ...<search results>...
   ```

2. Use the `ldapsearch` command to search all entries and display the results in descending order.

   Use the `--sortorder` option sorted on the attribute `sn`.

   ```
   $ ldapsearch -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
     -s sub -b dc=example,dc=com --sortOrder -sn "(objectclass)"
   dn: uid=pworrell,ou=People,dc=example,dc=com
   objectClass: person
   objectClass: organizationalPerson
   ...<search results>...
   ```

### 17.5.3.15 Searching Using the Simple Paged Results Control

The Simple Paged Results Control allows a search operation to return only a subset of the results at a time. It can be used to iterate through the search results a page at a time. It is similar to the Virtual List View Control with the exception that it does not require the results to be sorted and can only be used to iterate sequentially through the search results.

Use the `ldapsearch` command with the `--simplePageSize` option.

The following command also uses the `--countEntries` option to mark each page.

```
$ ldapsearch --hostname localhost --port 1389 \
 --bindDN "cn=Directory Manager" --bindPassword password \
 --searchScope sub --baseDN dc=example,dc=com \
 --simplePageSize 2 --countEntries "(objectclass=*)"

dn: ou=Groups,dc=example,dc=com
objectClass: organizationalunit
objectClass: top
ou: Groups

dn: ou=People,dc=example,dc=com
objectClass: organizationalunit
objectClass: top
ou: People

# Total number of matching entries: 2
```

```
dn: ou=Special Users,dc=example,dc=com
objectClass: organizationalUnit
objectClass: top
description: Special Administrative Accounts
ou: Special Users

dn: ou=Company Servers,dc=example,dc=com
objectClass: organizationalUnit
objectClass: top
description: Standard branch for Company Server registration
ou: Company Servers

# Total number of matching entries: 2

dn: ou=Contractors,dc=example,dc=com
objectClass: organizationalUnit
objectClass: top
ou: Contractors
ou: Product Testing
ou: Product Development
ou: Accounting

# Total number of matching entries: 1
```

### 17.5.3.16 Searching Using the Virtual List View Control

The Virtual List View Control allows a client to request that the server send search results in small, manageable chunks within a specific range of entries. It also allows a client to move forward and backward through the results of a search operation if configured with a GUI browser or application, or jump directly to a particular entry.

> **Note:** The Virtual List View Control requires that the returned entries be sorted.

Together with the `--virtualListView` option or its short form `-G`, specify the following arguments:

- **before**. Specify the number of entries before the target to include in the results.

  If the `before` value is greater than or equal to the target offset, then the `before` value is adjusted so that the first entry returned is the beginning of the list.

- **after**. Specify the number of entries after the target to include in the results.

- **index**. Specify the offset of the target entry within the result set. An index of 1 always means the first entry. If `index` and `content_count` are equal, the last entry is selected.

  If the `index` value is negative, the server rejects the request.

  If the `index` value is `0`, it is adjusted to `1` so that returned values are displayed.

  If the `index` value is greater than the total number of matching values, it is adjusted to one greater than the content count.

  The value of `index` can also be an assertion value, so that the returned entry contains that value. If the returned entry is so near the end of the list that the value of `after` extends beyond the last entry, the value of `after` is adjusted to display the appropriate entries.

- **count**. Specify the expected size of the result set.

- **count=0**. The target entry is the entry at the specified *index* position, starting from 1 and relative to the entire list of sorted results. Use this argument if the client does not know the size of the result set.
- **count=1**. The target entry is the first entry in the list of sorted results.
- **count>1**. The target entry is the first entry in the portion of the list represented by the fraction *index/count*. To target the last result in the list, use an *index* argument greater than the *count* argument. Client applications can use interfaces that allow users to move around a long list by using a scroll bar. For example, for an index of 33 and a count of 100, the application can jump 33 percent of the way into the list.

For example, the arguments (0:4:1:0) indicate that you want to show 0 entries before and 4 entries after the target entry at index 1. If the client does not know the size of the set, the count is 0.

#### 17.5.3.16.1 To Search Using the Virtual List View Control

The sort order option (`-S`) must be used with the Virtual List View control. This example uses the Virtual List View Control options to specify the following:

- **Before=0.** Specifies that 0 entries before the target should be displayed.
- **After=2.** Specifies that 2 entries after the target should be displayed.
- **Index=1.** Specifies that the offset of the target entry within the result set should be returned.
- **Count=0.** Specifies that target entry at the index position should be returned, which is the first entry.

Thus, the server returns the first entry plus two entries after the target sorted in ascending order by the `givenName` attribute.

Use the `ldapsearch` command with the `--virtualListView` option.

```
$ ldapsearch -h localhost -p 1389 -D "cn=Directory Manager" -w bindPassword \
  -b dc=example,dc=com --searchScope sub --sortOrder givenName \
  --virtualListView "0:2:1:0" "(objectclass=*)"

dn: uid=awhite,ou=People,dc=example,dc=com
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: top
givenName: Alan
uid: awhite
cn: Alan White
sn: White
...

dn: uid=aworrell,ou=People,dc=example,dc=com
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: top
givenName: Alan
uid: aworrell
cn: Alan Worrell
sn: Worrell
...
```

```
dn: uid=alutz,ou=People,dc=example,dc=com
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: top
givenName: Alexander
uid: alutz
cn: Alexander Lutz
sn: Lutz
...

# VLV Target Offset: 1
# VLV Content Count: 172
```

#### 17.5.3.16.2 To Search Using Virtual List View With a Specific Target

The sort order (`-S`) option must also be used with Virtual List View. The example command uses the Virtual List View Control options to specify the following:

- **Before=0.** Specifies that 0 entries before the target should be displayed.
- **After=4.** Specifies that 4 entries after the target should be displayed.
- **Index=jensen.** Specifies that the string `jensen` within the result set be returned.
- **Count=not specified.** Use the default `count=0`, which is the first entry.

Thus, the server returns the first `sn` attribute that matches `jensen` plus four `sn` attributes after the target sorted in ascending order by the `sn` attribute.

Use the `ldapsearch` command with the `--virtualListView` option.

```
$ ldapsearch -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
  -b dc=example,dc=com --searchScope sub --sortOrder sn \
  --virtualListView "0:4:jensen" "(objectclass=*)" sn

dn: uid=kjensen,ou=People,dc=example,dc=com
sn: Jensen

dn: uid=bjensen,ou=People,dc=example,dc=com
sn: Jensen

dn: uid=gjensen,ou=People,dc=example,dc=com
sn: Jensen

dn: uid=jjensen,ou=People,dc=example,dc=com
sn: Jensen

dn: uid=ajensen,ou=People,dc=example,dc=com
sn: Jensen

# VLV Target Offset: 56
# VLV Content Count: 172
```

#### 17.5.3.16.3 To Search Using Virtual List View With a Known Total

The sort order (`-S`) option must also be used with Virtual List View. The example command uses the Virtual List View Control options to specify the following:

- **Before=0.** Specifies that 0 entries before the target should be displayed.
- **After=2.** Specifies that 2 entries after the target should be displayed.

- **Index=57.** Specifies that the index of 57 within the result set should be returned. This is roughly one-third of the list.
- **Count=172.** Use the total count.

Thus, the server returns the first `sn` attribute that is one-third within the list, plus two `sn` attributes sorted in ascending order by the `sn` attribute.

Use the `ldapsearch` command with the `--virtualListView` option.

```
$ ldapsearch -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
  -b dc=example,dc=com -s sub --sortOrder sn \
  --virtualListView "0:2:57:172" "(objectclass=*)" sn

dn: uid=bjensen,ou=People,dc=example,dc=com
sn: Jensen

dn: uid=gjensen,ou=People,dc=example,dc=com
sn: Jensen

dn: uid=jjensen,ou=People,dc=example,dc=com
sn: Jensen

# VLV Target Offset: 57
# VLV Content Count: 172
```

#### 17.5.3.16.4 Allowing Anonymous Access to the Virtual List View Control

By default, access to the virtual list view control is allowed for authenticated users only. To allow unauthenticated users to access the virtual list view control, the OID for the virtual list view control (2.16.840.1.113730.3.4.9) must be added to the "Anonymous control access" global ACI and removed from the "Authenticated users control access" global ACI.

```
ds-cfg-global-aci: (targetcontrol="2.16.840.1.113730.3.4.2 ||
2.16.840.1.113730.3.4.17 || 2.16.840.1.113730.3.4.19 ||
1.3.6.1.4.1.4203.1.10.2 || 1.3.6.1.4.1.42.2.27.8.5.1 ||
2.16.840.1.113730.3.4.16 || 2.16.840.1.113894.1.8.31") (version 3.0; acl
"Anonymous control access"; allow(read) userdn="ldap:///anyone";)
ds-cfg-global-aci: (targetcontrol="1.3.6.1.1.12 || 1.3.6.1.1.13.1 ||
1.3.6.1.1.13.2 || 1.2.840.113556.1.4.319 || 1.2.826.0.1.3344810.2.3 ||
2.16.840.1.113730.3.4.18 || 2.16.840.1.113730.3.4.9 || 1.2.840.113556.1.4.473
|| 1.3.6.1.4.1.42.2.27.9.5.9 || 1.2.840.113556.1.4.473") (version 3.0; acl
"Authenticated users control access"; allow(read) userdn="ldap:///all";)
```

The easiest way to modify these global ACIs is to use ODSM, as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager."
2. Select the **Security** tab.
3. Under the **Root** menu, select **Anonymous control access**.
4. In the **Targets** table on the right hand pane, select the **Target Control** field, and click **Edit**.
5. From the **Available Controls** list, select **Virtual List View Control (2.16.840.1.113730.3.4.9)**.
6. Click the right arrow to move the VLV control to the **Selected Controls** list.
7. Click **OK**.

8. Click **Apply** to save your changes.
9. Under the **Root** menu, select **Authenticated users control access**.
10. In the **Targets** table on the right hand pane, select the **Target Control** field, and click **Edit**.
11. From the **Selected Controls** list, select **Virtual List View Control (2.16.840.1.113730.3.4.9)**.
12. Click the left arrow to move the VLV control to the **Available Controls** list.
13. Click **OK**.
14. Click **Apply** to save your changes.

You can also use `dsconfig` to modify the global ACIs, but it is not possible to modify an ACI value with dsconfig. Instead, the ACIs must be deleted and recreated. For more information, see Section 22.1.1, "Default Global ACIs".

## 17.5.4 Searching in Verbose Mode and With a Properties File

This section describes how to search in verbose mode and how to search by using a properties file, and contains the following topics:

- Section 17.5.4.1, "To Search in Verbose Mode"
- Section 17.5.4.2, "To Search Using a Properties File"

### 17.5.4.1 To Search in Verbose Mode

Verbose mode displays the processing information that is transmitted between client and server. This mode is convenient for debugging purposes.

Use the `ldapsearch` command as follows:

```
$ ldapsearch -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
   -b dc=example,dc=com -s base --verbose "(objectclass=*)"
LDAP: C>S 01:43:46.140 (0ms) LDAPMessage(msgID=1, protocolOp=BindRequest
  (version =3, dn=cn=Directory Manager, password=password))
ASN1: C>S 01:43:46.140 (0ms) ASN.1 Sequence
  BER Type: 30
  Decoded Values:
  ASN1Integer(type=02, value=1)
  ASN1Sequence(type=60, values={ ASN1Integer(type=02, value=3),
    cn=Directory Manager, opends })
  Value:
  02 01 01 60 23 02 01 03   04 14 63 6E 3D 64 69 72     `#  cn=directory
  65 63 74 6F 72 79 20 6D   61 6E 61 67 65 72 80 08          manager
  70 61 73 73 77 6F 72 64                                    password
...
```

### 17.5.4.2 To Search Using a Properties File

The directory server supports the use of a properties file that holds default argument values used with the `ldapsearch` command. The properties file is convenient when working in different configuration environments, especially in scripted or embedded applications. For more information, see Appendix A.1.2, "Using a Properties File With Server Commands."

1. Create a properties file in any text editor, with the following content:

```
hostname=localhost
port=1389
```

```
bindDN=cn=Directory Manager
bindPasswordFile=pwd-file
baseDN=dc=example,dc=com
searchScope=sub
sortOrder=givenName
virtualListView=0:2:1:0
```

2. Save the file as `tools.properties`.
3. Use the `ldapsearch` with the `--propertiesFilePath` option.

```
$ ldapsearch --propertiesFilePath tools.properties "(objectclass=*)"
```

## 17.5.5 Searching Internationalized Entries

Oracle Unified Directory supports collation rules that match entries and can be used with the Section 17.5.3.14, "Searching Using the Server-Side Sort Control" to sort search results. The collation rule is specified in the search filter as a matching rule, delimited by colons, as shown here:

*locale* . *matchingRule*

where:

- *locale* is specified in one of the following ways
  - Locale OID
  - Locale character suffix (such as `ar`, `en`, or `fr-CA`).

    See Section 17.5.5.2, "Supported Collation Rules" at the end of this section for a list of supported locales, their OIDs, and tags.
- *matchingRule* can specified as either a numeric suffix or a character suffix appended to the *locale*, as listed in Table 17–1.

> **Note:** If the locale is specified by its OID, then the matching rule must be specified by its numeric suffix. In this case, the matching rule cannot be specified by the character suffix.

***Table 17–1 Matching Rule Suffixes***

| Matching Rule | Numeric Suffix | Character Suffix |
|---|---|---|
| Less than | .1 | .lt |
| Less than or equal to | .2 | .lte |
| Equality | .3 | .eq (default) |
| Greater than or equal to | .4 | .gte |
| Greater than | .5 | .gt |
| Substring | .6 | .sub |

`Equality` is the default matching rule. That is, when no matching rule suffix is specified, the collation rule uses `equality` matching rule. The two following examples are equivalent and specify the English collation rule and the equality matching rule, but the second example specifies the equality matching rule explicitly with the.`eq` suffix:

```
"cn:en:=sanchez"
```

```
"cn:en.eq:=sanchez"
```

The next example shows the same search filter, but specified using the locale's character suffix and the matching rule's numeric code:

```
"cn:en.3:=sanchez"
```

The following example shows the same search filter specified using the locale OID and the matching rule numeric suffix:

```
"cn:1.3.6.1.4.1.42.2.27.9.4.34.1.3:=sanchez"
```

The following examples specify the same search filter but with a Spanish collation rule.

```
"cn:es.eq:=sanchez"
"cn:1.3.6.1.4.1.42.2.27.9.4.49.1.3:=sanchez"
"cn:es.3:=sanchez"
```

The following examples specify a similar search filter that uses a greater-than matching rule with the Spanish collation rule.

```
"cn:es.gt:=sanchez"
"cn:1.3.6.1.4.1.42.2.27.9.4.49.1.5:=sanchez"
"cn:es.5:=sanchez"
```

This section contains the following topics:

- Section 17.5.5.1, "Examples"
- Section 17.5.5.2, "Supported Collation Rules"

### 17.5.5.1 Examples

***Example 17–1 Equality Search***

The following search uses a filter with the `en` (`en-US`) locale OID to perform an equality search to return any entry with a `cn` value of `sanchez`:

```
$ ldapsearch -D "cn=directory manager" -j pwd-file -b "o=test" \
  "cn:1.3.6.1.4.1.42.2.27.9.4.34.1:=sanchez"
```

The following filters return the same results:

- `"cn:en:=sanchez"`
- `"cn:en.3:=sanchez"`
- `"cn:en.eq:=sanchez"`
- `"cn:1.3.6.1.4.1.42.2.27.9.4.34.1.3:=sanchez"`

***Example 17–2 Less-Than Search***

The following search uses a filter with the `es` (`es-ES`) locale and performs a less-than search and returns the entry with a `departmentnumber` value of `abc119`:

```
$ ldapsearch -D "cn=directory manager" -j pwd-file -b "o=test" \
  "departmentnumber:1.3.6.1.4.1.42.2.27.9.4.49.1.1:=abc120"
```

The following filters return the same results:

- `"departmentnumber:es.1:=abc120"`

- `"departmentnumber:es.lt:=abc120"`

***Example 17–3 Less-Than-or-Equal-To Search***

The following search uses a filter with the `es` (`es-ES`) locale and performs a less-than-or-equal-to search that returns the entry with a `departmentnumber` value of `abc119`:

```
$ ldapsearch -D "cn=directory manager" -j pwd-file -b "o=test" \
  "departmentnumber:1.3.6.1.4.1.42.2.27.9.4.49.1.2:=abc119"
```

The following filters return the same results:

- `"departmentnumber:es.2:=abc119"`
- `"departmentnumber:es.lte:=abc119"`

***Example 17–4 Greater-Than-or-Equal-To Search***

The following search uses a filter with the `fr` (`fr-FR`) locale and performs a greater-than-or-equal-To search that returns an entry with a `departmentnumber` value of `abc119`

```
$ ldapsearch -D "cn=directory manager" -j pwd-file -b "o=test" \
  "departmentnumber:fr.4:=abc119"
```

The following filters return the same results:

- `"departmentnumber:1.3.6.1.4.1.42.2.27.9.4.76.1.4:=abc119"`
- `"departmentnumber:fr.gte:=abc119"`

***Example 17–5 Greater-Than Search***

The following search uses a filter with the `fr` (`fr-FR`) locale and performs a greater-than search:

```
$ ldapsearch -D "cn=directory manager" -j pwd-file -b "o=test" \
  "departmentnumber:fr.5:=abc119"
```

The above search should *not* return an entry with a `departmentnumber` value of `abc119`.

The following filters return the same results:

- `"departmentnumber:1.3.6.1.4.1.42.2.27.9.4.76.1.5:=abc119"`
- `"departmentnumber:fr.gt:=abc119"`

***Example 17–6 Substring Search***

The following search uses a filter with the `en` (`en-US`) locale and performs a substring search that returns an entry with an `sn` value of "Quebec":

```
$ ldapsearch -D "cn=directory manager" -j pwd-file -b "o=test" \
  "sn:en.6:=*u*bec"
```

The following filters return the same results:

- `"sn:1.3.6.1.4.1.42.2.27.9.4.34.1.6:=*u*bec"`
- `"sn:en.sub:=*u*bec"`

### 17.5.5.2 Supported Collation Rules

The following table lists the internationalization locales supported by Oracle Unified Directory, alphabetized by character suffix.

***Table 17–2 Supported Collation Rules***

| Locale | Character Suffix | OID |
|---|---|---|
| Arabic | ar | 1.3.6.1.4.1.42.2.27.9.4.3.1 |
| Arabic United Arab Emirates | ar-AE | 1.3.6.1.4.1.42.2.27.9.4.4.1 |
| Arabic Bahrain | ar-BH | 1.3.6.1.4.1.42.2.27.9.4.5.1 |
| Arabic Algeria | ar-DZ | 1.3.6.1.4.1.42.2.27.9.4.6.1 |
| Arabic Egypt | ar-EG | 1.3.6.1.4.1.42.2.27.9.4.7.1 |
| Arabic India | ar-IQ | 1.3.6.1.4.1.42.2.27.9.4.9.1 |
| Arabic Jordan | ar-JO | 1.3.6.1.4.1.42.2.27.9.4.10.1 |
| Arabic Kuwait | ar-KW | 1.3.6.1.4.1.42.2.27.9.4.11.1 |
| Arabic Lebanon | ar-LB | 1.3.6.1.4.1.42.2.27.9.4.12.1 |
| Arabic Lybia | ar-LY | 1.3.6.1.4.1.42.2.27.9.4.13.1 |
| Arabic Morocco | ar-MA | 1.3.6.1.4.1.42.2.27.9.4.14.1 |
| Arabic Oman | ar-OM | 1.3.6.1.4.1.42.2.27.9.4.15.1 |
| Arabic Qatar | ar-QA | 1.3.6.1.4.1.42.2.27.9.4.16.1 |
| Arabic Saudi Arabia | ar-SA | 1.3.6.1.4.1.42.2.27.9.4.17.1 |
| Arabic Sudan | ar-SD | 1.3.6.1.4.1.42.2.27.9.4.18.1 |
| Arabic Syria | ar-SY | 1.3.6.1.4.1.42.2.27.9.4.19.1 |
| Arabic Tunisia | ar-TN | 1.3.6.1.4.1.42.2.27.9.4.20.1 |
| Arabic Yemen | ar-YE | 1.3.6.1.4.1.42.2.27.9.4.21.1 |
| Byelorussian | be | 1.3.6.1.4.1.42.2.27.9.4.22.1 |
| Bulgaria | bg | 1.3.6.1.4.1.42.2.27.9.4.23.1 |
| Catalan | ca | 1.3.6.1.4.1.42.2.27.9.4.25.1 |
| Czech | cs | 1.3.6.1.4.1.42.2.27.9.4.26.1 |
| Danish | da | 1.3.6.1.4.1.42.2.27.9.4.27.1 |
| German | de | 1.3.6.1.4.1.142.2.27.9.4.28.1 |
| German Germany | de-DE | 1.3.6.1.4.1.142.2.27.9.4.28.1 |
| German Austria | de-AT | 1.3.6.1.4.1.42.2.27.9.4.29.1 |
| German Swiss | de-CH | 1.3.6.1.4.1.42.2.27.9.4.31.1 |
| German Luxembourg | de-LU | 1.3.6.1.4.1.42.2.27.9.4.32.1 |
| Greek | el | 1.3.6.1.4.1.42.2.27.9.4.33.1 |
| English | en | 1.3.6.1.4.1.42.2.27.9.4.34.1 |
| English US | en-US | 1.3.6.1.4.1.42.2.27.9.4.34.1 |
| English Australia | en-AU | 1.3.6.1.4.1.42.2.27.9.4.35.1 |
| English Canada | en-CA | 1.3.6.1.4.1.42.2.27.9.4.36.1 |
| English Great Britain | en-GB | 1.3.6.1.4.1.42.2.27.9.4.37.1 |

*Table 17–2 (Cont.) Supported Collation Rules*

| Locale | Character Suffix | OID |
|---|---|---|
| English Ireland | en-IE | 1.3.6.1.4.1.42.2.27.9.4.39.1 |
| English India | en-IN | 1.3.6.1.4.1.42.2.27.9.4.40.1 |
| English New Zealand | en-NZ | 1.3.6.1.4.1.42.2.27.9.4.42.1 |
| English South Africa | en-ZA | 1.3.6.1.4.1.42.2.27.9.4.46.1 |
| Spanish | es | 1.3.6.1.4.1.42.2.27.9.4.49.1 |
| Spanish Spain | es-ES | 1.3.6.1.4.1.42.2.27.9.4.49.1 |
| Spanish Argentina | es-AR | 1.3.6.1.4.1.42.2.27.9.4.50.1 |
| Spanish Bolivia | es-BO | 1.3.6.1.4.1.42.2.27.9.4.51.1 |
| Spanish Chile | es-CL | 1.3.6.1.4.1.42.2.27.9.4.52.1 |
| Spanish Colombia | es-CO | 1.3.6.1.4.1.42.2.27.9.4.53.1 |
| Spanish Costa Rica | es-CR | 1.3.6.1.4.1.42.2.27.9.4.54.1 |
| Spanish Dominican Republic | es-DO | 1.3.6.1.4.1.42.2.27.9.4.55.1 |
| Spanish Ecuador | es-EC | 1.3.6.1.4.1.42.2.27.9.4.56.1 |
| Spanish Guatemala | es-GT | 1.3.6.1.4.1.42.2.27.9.4.57.1 |
| Spanish Honduras | es-HN | 1.3.6.1.4.1.42.2.27.9.4.58.1 |
| Spanish Mexico | es-MX | 1.3.6.1.4.1.42.2.27.9.4.59.1 |
| Spanish Nicaragua | es-NI | 1.3.6.1.4.1.42.2.27.9.4.60.1 |
| Spanish Panama | es-PA | 1.3.6.1.4.1.42.2.27.9.4.61.1 |
| Spanish Peru | es-PE | 1.3.6.1.4.1.42.2.27.9.4.62.1 |
| Spanish Puerto Rico | es-PR | 1.3.6.1.4.1.42.2.27.9.4.63.1 |
| Spanish Paraguay | es-PY | 1.3.6.1.4.1.42.2.27.9.4.64.1 |
| Spanish Salvador | es-SV | 1.3.6.1.4.1.42.2.27.9.4.65.1 |
| Spanish Uraguay | es-UY | 1.3.6.1.4.1.42.2.27.9.4.67.1 |
| Spanish Venezuela | es-VE | 1.3.6.1.4.1.42.2.27.9.4.68.1 |
| Estonian | et | 1.3.6.1.4.1.42.2.27.9.4.69.1 |
| Finnish | fi | 1.3.6.1.4.1.42.2.27.9.4.74.1 |
| French | fr | 1.3.6.1.4.1.42.2.27.9.4.76.1 |
| French | fr-FR | 1.3.6.1.4.1.42.2.27.9.4.76.1 |
| French | fr-BE | 1.3.6.1.4.1.42.2.27.9.4.77.1 |
| French | fr-CA | 1.3.6.1.4.1.42.2.27.9.4.78.1 |
| French | fr-CH | 1.3.6.1.4.1.42.2.27.9.4.79.1 |
| French | fr-LU | 1.3.6.1.4.1.42.2.27.9.4.80.1 |
| Hebrew | he | 1.3.6.1.4.1.42.2.27.9.4.85.1 |
| Croatian | hr | 1.3.6.1.4.1.42.2.27.9.4.87.1 |
| Hungarian | hu | 1.3.6.1.4.1.42.2.27.9.4.88.1 |
| Icelandic | is | 1.3.6.1.4.1.42.2.27.9.4.91.1 |
| Italian | it | 1.3.6.1.4.1.42.2.27.9.4.92.1 |

*Table 17–2 (Cont.) Supported Collation Rules*

| Locale | Character Suffix | OID |
|---|---|---|
| Italian-Swiss | it-CH | 1.3.6.1.4.1.42.2.27.9.4.93.1 |
| Japanese | ja | 1.3.6.1.4.1.42.2.27.9.4.94.1 |
| Korean | ko | 1.3.6.1.4.1.42.2.27.9.4.97.1 |
| Lithuanian | lt | 1.3.6.1.4.1.42.2.27.9.4.100.1 |
| Latvian | lv | 1.3.6.1.4.1.42.2.27.9.4.101.1 |
| Macedonian | mk | 1.3.6.1.4.1.42.2.27.9.4.102.1 |
| Dutch | nl | 1.3.6.1.4.1.42.2.27.9.4.105.1 |
| Dutch Netherlands | nl-NL | 1.3.6.1.4.1.42.2.27.9.4.105.1 |
| Dutch Belgium | nl-BE | 1.3.6.1.4.1.42.2.27.9.4.106.1 |
| Norwegian | no | 1.3.6.1.4.1.42.2.27.9.4.107.1 |
| Norwegian Norway | no-NO | 1.3.6.1.4.1.42.2.27.9.4.107.1 |
| Norwegian Nynorsk | no-NO-NY | 1.3.6.1.4.1.42.2.27.9.4.108.1 |
| Polish | pl | 1.3.6.1.4.1.42.2.27.9.4.114.1 |
| Portuguese | pt | 1.3.6.1.4.1.42.2.27.9.4.115.1 |
| Portuguese Portugal | pt-PT | 1.3.6.1.4.1.42.2.27.9.4.115.1 |
| Portugues Brazil | pt-BR | 1.3.6.1.4.1.42.2.27.9.4.116.1 |
| Romanian | ro | 1.3.6.1.4.1.42.2.27.9.4.117.1 |
| Russian | ru | 1.3.6.1.4.1.42.2.27.9.4.118.1 |
| Russian Russia | ru-RU | 1.3.6.1.4.1.42.2.27.9.4.118.1 |
| Slovak | sk | 1.3.6.1.4.1.42.2.27.9.4.121.1 |
| Slovenia | sl | 1.3.6.1.4.1.42.2.27.9.4.122.1 |
| Albanian | sq | 1.3.6.1.4.1.42.2.27.9.4.127.1 |
| Serbian | sr | 1.3.6.1.4.1.42.2.27.9.4.128.1 |
| Swedish | sv | 1.3.6.1.4.1.42.2.27.9.4.129.1 |
| Swedish Sweden | sv-SE | 1.3.6.1.4.1.42.2.27.9.4.129.1 |
| Thai | th | 1.3.6.1.4.1.42.2.27.9.4.136.1 |
| Turkish | tr | 1.3.6.1.4.1.42.2.27.9.4.140.1 |
| Ukrainian | uk | 1.3.6.1.4.1.42.2.27.9.4.141.1 |
| Vietnamese | vi | 1.3.6.1.4.1.42.2.27.9.4.142.1 |
| Chinese | zh | 1.3.6.1.4.1.42.2.27.9.4.143.1 |
| Chinese China | zh-CN | 1.3.6.1.4.1.42.2.27.9.4.144.1 |
| Chinese Hong Kong | zh-HK | 1.3.6.1.4.1.42.2.27.9.4.145.1 |
| Chinese Taiwan | zh-TW | 1.3.6.1.4.1.42.2.27.9.4.148.1 |

## 17.6 Adding, Modifying, and Deleting Directory Data

The directory server provides a full set of LDAPv2- and LDAPv3-compliant client tools to manage directory entries. You can add, update, or remove entries by using the `ldapmodify` and `ldapdelete` utilities. The LDAP command-line utilities require

LDAP Data Interchange Format (LDIF)-formatted input, entered through the command line or read from a file.

Before you make modifications to directory data, make sure that you understand the following concepts:

- The privilege and access control mechanisms.

  For information about setting privileges, Chapter 22, "Controlling Access To Data."

- The structure of your directory server.
- The schema of your directory server.

This section contains the following topics:

- Section 17.6.1, "Adding Directory Entries"
- Section 17.6.2, "Adding Attributes"
- Section 17.6.3, "Modifying Directory Entries"
- Section 17.6.4, "Deleting Directory Entries"

## 17.6.1 Adding Directory Entries

You can add one or more entries to a directory server by using the `ldapmodify` command. `ldapmodify` opens a connection to the directory server, binds to it, and performs the modification to the database (in this case, an "add") as specified by the command-line options.

`ldapmodify` enables you to add entries in one of two ways:

- **Using the --defaultAdd option**. Use the `--defaultAdd` option to add new entries to the directory when data is entered on the command line. Press Ctrl-D (UNIX, Linux) or Ctrl-Z (Windows) when finished, or use an input file with your changes.
- **Using LDIF update statements**. LDIF update statements define how `ldapmodify` changes the directory entry. LDIF update statements contain the DN of the entry to be modified, *changetype* that defines how a specific entry is to be modified (add, delete, modify, modrdn), and a series of attributes and their changed values.

> **Note:** Any newly added entry must conform to the directory's schema. If you add any entry that does not conform to the schema, the server responds with an Object Class Violation error. You can view the details of the error in the `errors` log.

This section contains the following topics:

- Section 17.6.1.1, "To Create a Root Entry"
- Section 17.6.1.2, "To Add an Entry Using the `--defaultAdd` Option With `ldapmodify`"
- Section 17.6.1.3, "To Add Entries Using an LDIF Update Statement With `ldapmodify`"

### 17.6.1.1 To Create a Root Entry

The root entry is the topmost entry in the directory and must contain the naming context, or root suffix. You can set up the root entry when you first install the directory

server using the graphical user interface (GUI) or the command-line. If you install the directory without any data, create a root entry using the `ldapmodify` command with the `--defaultAdd` option.

1. Create the root entry using `ldapmodify`.

```
$ ldapmodify --hostname localhost --port 1389 --defaultAdd \
  --bindDN "cn=Directory Manager" --bindPassword password
dn: dc=example,dc=com
objectclass: domain
objectclass: top
dc: example
(Press Ctrl-D on Unix, Linux)
(Press Ctrl-Z on Windows), then press ENTER.

Processing ADD request for dc=example,dc=com
ADD operation successful for DN dc=example,dc=com
```

> **Note:** The `--bindDN` and `--bindPassword` options specify the bind DN and password, respectively, of the user with permissions to add new entries. You can provide the clear-text version of the password. The server encrypts this value and store only the encrypted one. Be sure to limit read permissions to protect clear passwords that appear in LDIF files. To avoid this security issue, use SSL or startTLS.

2. Verify the change by using the `ldapsearch` command.

```
$ ldapsearch --hostname localhost --port 1389 --baseDN "dc=example,dc=com" \
  --searchScope base --bindDN "cn=Directory Manager" --bindPassword password \
  "(objectclass=*)"
dn: dc=example,dc=com
objectClass: domain
objectClass: top
dc: example
```

#### 17.6.1.2 To Add an Entry Using the `--defaultAdd` Option With `ldapmodify`

1. Create your directory entry in LDIF format.

   Before you add an entry, ensure that the suffix to which you want to add the entry exists in your database (for example, `ou=People,dc=example,dc=com`).

   For this example, create an input file called `new.ldif` with the following contents:

```
dn: uid=Marcia Garza,ou=People,dc=example,dc=com
cn: Marcia Garza
sn: Garza
givenName: Marcia
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
ou: Accounting
ou: People
l: Santa Clara
uid: mgarza
mail: mgarza@example.com
```

```
roomnumber: 5484
userpassword: donuts
```

2. Add the entry using `ldapmodify` with the `--defaultAdd` option.

```
$ ldapmodify --hostname localhost --port 1389 --bindDN "cn=Directory Manager" \
  --bindPassword password --defaultAdd --filename /tmp/new.ldif
```

#### 17.6.1.3 To Add Entries Using an LDIF Update Statement With `ldapmodify`

1. Create the entry in LDIF format with the `changetype:add` element.

   Make sure that there are no trailing spaces after `add`. If a space exists after `add`, the server base-64 encodes the value to represent the space, which can cause problems.

   For this example, create an input LDIF file named `new.ldif`.

```
dn: uid=Marcia Garza,ou=People,dc=example,dc=com
changetype: add
cn: Marcia Garza
sn: Garza
givenName: Marcia
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
ou: Accounting
ou: People
l: Santa Clara
uid: mgarza
mail: mgarza@example.com
roomnumber: 5484
userpassword: donuts
```

2. Add the entry using `ldapmodify`.

   Do not include the `-a` option as the `changetype` attribute specifies the action.

```
$ ldapmodify --hostname localhost --port 1389 --bindDN "cn=Directory Manager" \
  --bindPassword password --filename /tmp/new.ldif

Processing ADD request for uid=Marcia Garza,ou=People,dc=example,dc=com
ADD operation successful for DN uid=Marcia Garza,ou=People,dc=example,dc=com
```

## 17.6.2 Adding Attributes

The LDIF `changetype:add` statement adds an entry to the directory. To add attributes to an entry, use the `changetype:modify` statement, as shown in the following examples. You can combine multiple commands within a file by separating each command with a dash ("-").

This section describes how to manage an entry, and contains the following topics:

- Section 17.6.2.1, "To Add an Attribute to an Entry"
- Section 17.6.2.2, "To Add an ACI Attribute"
- Section 17.6.2.3, "To Add an International Attribute"

#### 17.6.2.1 To Add an Attribute to an Entry

1. Create the entry in LDIF format with the `changetype:modify` element.

Use the `modify` change type, because you are modifying an existing entry with the addition of a new attribute. Make sure that there are no trailing spaces after `modify`. After the `changetype`, specify `add: newAttributeName` and, on the following line, the value of the new attribute.

For this example, create an input LDIF file called `add_attribute.ldif`, as follows:

```
dn: uid=Marcia Garza,ou=People,dc=example,dc=com
changetype: modify
add: telephonenumber
telephonenumber: +1 408 555 8283
```

---

**Note:** To add multiple attributes, separate the attributes with a dash (-), for example:

```
dn: uid=Marcia Garza,ou=People,dc=example,dc=com
changetype: modify
add: telephonenumber
telephonenumber: +1 408 555 8283
-
add: building
building: sc09
```

---

2. Add the attribute by using `ldapmodify`.

```
$ ldapmodify --hostname localhost --port 1389 --bindDN "cn=Directory Manager" \
  --bindPassword password --filename /tmp/add_attribute.ldif

Processing MODIFY request for uid=Marcia Garza,ou=People,dc=example,dc=com
MODIFY operation successful for DN uid=Marcia Garza,ou=People,dc=example,dc=com
```

#### 17.6.2.2 To Add an ACI Attribute

You can use `ldapmodify` to add access control instructions (ACIs) to manage access rights for a user's account. For more information, see Chapter 22, "Controlling Access To Data" and ACI Syntax.

The following example allows a user to modify her own directory attributes.

1. Create the LDIF file containing the ACI.

```
dn: uid=Marcia Garza,ou=People,dc=example,dc=com
changetype: modify
add: aci
aci: (target="ldap:///uid=Marcia Garza,ou=People,dc=example,dc=com")
 (targetattr="*")(version 3.0; acl "mgarza rights"; allow (write)
 userdn="ldap:///self";)
```

2. Add the attribute by using `ldapmodify`.

```
$ ldapmodify --hostname localhost --port 1389 --bindDN "cn=Directory Manager" \
  --bindPassword password --filename /tmp/add_aci.ldif

Processing MODIFY request for uid=Marcia Garza,ou=People,dc=example,dc=com
MODIFY operation successful for DN uid=Marcia Garza,ou=People,dc=example,dc=com
```

#### 17.6.2.3 To Add an International Attribute

The directory server represents international locales using a language tag in the form *attribute;language-subtype*. For example, `homePostalAddress;lang-jp:address` specifies the postal address with the locale in Japan (`subtype=jp`).

Use `ldapmodify` to add the attribute.

Affix the language subtype, `lang-cc`, where *cc* is the country code.

```
$ ldapmodify --hostname localhost --port 1389 --bindDN "cn=Directory Manager" \
  --bindPassword password
dn: uid=jarrow,ou=People,dc=example,dc=com
changetype: modify
add: homePostalAddress;lang-jp
homePostalAddress;lang-jp: 1-8-15 Azuchimachi, Chuo-ku
(Press Ctrl-D on Unix, Linux)
(Press Ctrl-Z on Windows), then press ENTER.
```

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

### 17.6.3 Modifying Directory Entries

Use the LDIF update statement `changetype:modify` to make changes to existing directory data. The following procedures provide examples of modifying directory entries, and contains the sections:

- Section 17.6.3.1, "To Modify an Attribute Value"
- Section 17.6.3.2, "To Modify an Attribute With Before and After Snapshots"
- Section 17.6.3.3, "To Delete an Attribute"
- Section 17.6.3.4, "To Change an RDN"
- Section 17.6.3.5, "To Move an Entry"

For more information, see Appendix A.4.3, "ldapmodify."

#### 17.6.3.1 To Modify an Attribute Value

Use `ldapmodify` to change the entry, using the `changetype:modify` and `replace` elements.

Ensure that there are no trailing spaces after `modify`.

This example modifies a user's existing telephone number.

```
$ ldapmodify -h localhost -p 1389 D "cn=Directory Manager" -j pwd-file \
dn: uid=Marcia Garza,ou=People,dc=example,dc=com
changetype: modify
replace: telephonenumber
telephonenumber: +1 408 555 8288

Processing MODIFY request for uid=Marcia Garza,ou=People,dc=example,dc=com
MODIFY operation successful for DN uid=Marcia Garza,ou=People,dc=example,dc=com
```

To modify multiple attributes, separate the attributes with a dash (-), for example:

```
dn: uid=Marcia Garza,ou=People,dc=example,dc=com
changetype: modify
replace: telephonenumber
telephonenumber: +1 408 555 6465
```

```
-
add: facsimiletelephonenumber
facsimiletelephonenumber: +1 408 222 4444
-
replace: l
l: Sunnyvale
```

#### 17.6.3.2 To Modify an Attribute With Before and After Snapshots

The `ldapmodify` command provides the options, `--preReadAttribute` and `--postReadAttribute`, that return the modified attribute value with a *before* and *after* snapshot, respectively.

Use `ldapmodify` with the `--preReadAttribute` and `--postReadAttribute` options.

This example modifies a user's existing telephone number.

```
$ ldapmodify -h localhost -p 1389 D "cn=Directory Manager" -j pwd-file \
  --preReadAttributes telephoneNumber --postReadAttributes telephoneNumber
dn: uid=Marcia Garza,ou=People,dc=example,dc=com
changetype: modify
replace: telephonenumber
telephonenumber: +1 408 555 8288

Processing MODIFY request for uid=Marcia Garza,ou=People,dc=example,dc=com
MODIFY operation successful for DN uid=Marcia Garza,ou=People,dc=example,dc=com

Target entry before the operation:
dn: uid=Marcia Garza,ou=People,dc=example,dc=com
telephonenumber: +1 408 555 4283

Target entry after the operation:
dn: uid=Marcia Garza,ou=People,dc=example,dc=com
telephonenumber: +1 408 555 8288
```

#### 17.6.3.3 To Delete an Attribute

This example deletes the location (l) attribute from an entry.

Use the `ldapmodify` to delete the attribute.

```
$ ldapmodify -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file
dn: uid=Marcia Garza,ou=People,dc=example,dc=com
changetype: modify
delete: l
(Press CTRL-D for Unix, Linux) (Press CTRL-Z for Windows), then press ENTER.

Processing MODIFY request for uid=Marcia Garza,ou=People,dc=example,dc=com
MODIFY operation successful for DN uid=Marcia Garza,ou=People,dc=example,dc=com
```

Type control-D (UNIX, Linux) or control-Z (Windows) to complete the input.

#### 17.6.3.4 To Change an RDN

The distinguished name (DN) of an entry uniquely identifies and describes that entry. A distinguished name consists of the name of the entry itself as well as the names, in order from bottom to top, of the objects above it in the directory.

The relative distinguished name (RDN) is the leftmost element in an entry DN. For example, the RDN for `uid=Marcia Garza,ou=People,dc=example,dc=com` is

uid=Marcia Garza. To change an RDN, use the changetype:moddn LDIF update statement.

You can specify if the old RDN should be retained in the directory by using the deleteoldrdn attribute. A deleteoldrdn value of 0 indicates that the existing RDN should be retained in the directory. A value of 1 indicates that the existing RDN should be replaced by the new RDN value.

1. Use the ldapmodify command to rename the entry.

   In this example, an employee Marcia Garza wants to change to her married name, Marcia Peters.

```
$ ldapmodify -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file
dn: uid=Marcia Garza,ou=Marketing,dc=example,dc=com
changetype: moddn
newrdn: uid=Marcia Peters
deleteoldrdn: 1
Processing MODIFY DN request for uid=Marcia Garza,ou=People,dc=example,dc=com
MODIFY DN operation successful for DN uid=Marcia
Garza,ou=People,dc=example,dc=com
```

2. Change any other attributes as necessary.

   In this example, certain attributes might still list the user's previous name.

```
$ ldapmodify -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file
dn: uid=Marcia Peters,ou=People,dc=example,dc=com
changetype: modify
replace: sn
sn: Peters
-
replace: cn
cn: Marcia Peters
-
replace: uid
uid: mpeters
uid: Marcia Peters
-
replace: mail
mail: mpeters@example.com
(Press Ctrl-D on Unix, Linux)
(Press Ctrl-Z on Windows), then press ENTER.

Processing MODIFY request for uid=Marcia Peters,ou=People,dc=example,dc=com
MODIFY operation successful for DN uid=Marcia
Peters,ou=People,dc=example,dc=com
```

### 17.6.3.5 To Move an Entry

If you are moving an entry from one parent to another, extend the access control instruction (ACI) rights on the parent entries. On the current parent entry of the entry to be moved, ensure that the ACI allows the export operations by using the syntax allow(export...). On the future parent entry of the entry to be moved, ensure that the ACI allows the import operations by using the syntax allow(import...).

In this example, move uid=sgarza from the ou=Contractors,dc=example,dc=com suffix to the ou=People,dc=example,dc=com subtree.

1. Use ldapmodify with the moddn changetype to move the entry.

```
$ ldapmodify -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file
```

```
dn: uid=sgarza,ou=Contractors,dc=example,dc=com
changetype: moddn
newrdn: uid=sgarza
deleteoldrdn: 0
newsuperior: ou=People,dc=example,dc=com
--filename move_entry.ldif
Processing MODIFY DN request for uid=sgarza,ou=Contractors,dc=example,dc=com
MODIFY DN operation successful for DN
uid=sgarza,ou=Contractors,dc=example,dc=com
```

2. Change any other attribute values, as required.

   The following example provides before and after snapshot changes for the `ou` attribute.

```
$ ldapmodify -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
  --preReadAttributes ou --postReadAttributes ou
dn: uid=sgarza,ou=People,dc=example,dc=com
changetype: modify
replace: ou
ou: People
ou: Product Testing
(Press Ctrl-D on Unix, Linux)
(Press Ctrl-Z on Windows), then press ENTER.

Processing MODIFY request for uid=sgarza,ou=People,dc=example,dc=com
MODIFY operation successful for DN uid=sgarza,ou=People,dc=example,dc=com

Target entry before the operation:
dn: uid=sgarza,ou=People,dc=example,dc=com
ou: Contractors
ou: Product Testing

Target entry after the operation:
dn: uid=sgarza,ou=People,dc=example,dc=com
ou: People
ou: Product Testing
```

### 17.6.4 Deleting Directory Entries

You can use `ldapmodify` and `ldapdelete` to remove entries from the directory. The `ldapmodify` command removes entries and attributes by using the LDIF update statements `changetype:delete` and `changetype:modify` with the `delete` attribute, respectively. The `ldapdelete` tool removes only entries.

> **Note:** You cannot delete an entry that has children entries. If you want to delete an entry that has children, first delete all the children entries below the targeted entry, then delete the entry.

For more information, see Appendix A.4.2, "ldapdelete."

This section describes how to delete directory entries, and contains the following topics:

- Section 17.6.4.1, "To Delete an Entry With `ldapmodify`"
- Section 17.6.4.2, "To Delete an Entry With `ldapdelete`"
- Section 17.6.4.3, "To Delete Multiple Entries by Using a DN File"

#### 17.6.4.1 To Delete an Entry With `ldapmodify`

Use the `ldapmodify` command with the `changetype:delete` statement.

```
$ ldapmodify -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file
dn: uid=Marcia Garza,ou=People,dc=example,dc=com
changetype: delete
(Press CTRL-D for Unix)
(Press CTRL-Z for Windows), then press ENTER.

Processing DELETE request for uid=Marcia Garza,ou=People,dc=example,dc=com
DELETE operation successful for DN uid=Marcia Garza,ou=People,dc=example,dc=com
The number of entries deleted was 1
```

#### 17.6.4.2 To Delete an Entry With `ldapdelete`

Use the `ldapdelete` command and specify the entry that you want to delete.

```
$ ldapdelete -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file
"uid=mgarza,ou=People,dc=example,dc=com"

Processing DELETE request for uid=Marcia Garza,ou=People,dc=example,dc=com
DELETE operation successful for DN uid=Marcia Garza,ou=People,dc=example,dc=com
```

#### 17.6.4.3 To Delete Multiple Entries by Using a DN File

1. Create a file that contains a list of DNs to be deleted.

   In this example, the file is named `delete.ldif`. The file must list each DN on a separate line, for example:

   ```
   uid=mgarza,ou=People,dc=example,dc=com
   uid=wsmith,ou=People,dc=example,dc=com
   uid=jarrow,ou=People,dc=example,dc=com
   uid=mbean,ou=People,dc=example,dc=com
   ```

2. Delete the entries by passing the file as an argument to the `ldapdelete` command.

   ```
   $ ldapdelete -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
     --continueOnError --filename delete.ldif

   Processing DELETE request for uid=mgarza,ou=People,dc=example,dc=com
   DELETE operation successful for DN uid=mgarza,ou=People,dc=example,dc=com
   Processing DELETE request for uid=wsmith,ou=People,dc=example,dc=com
   DELETE operation successful for DN uid=wsmith,ou=People,dc=example,dc=com
   Processing DELETE request for uid=jarrow,ou=People,dc=example,dc=com
   DELETE operation successful for DN uid=jarrow,ou=People,dc=example,dc=com
   Processing DELETE request for uid=mbean,ou=People,dc=example,dc=com
   DELETE operation successful for DN uid=mbean,ou=People,dc=example,dc=com
   ```

   **Note:** The `--continueOnError` option specifies that if an error occurs, the command continues to the next search item.

## 17.7 Indexing Directory Data

This section describes how to index attributes using the `dsconfig` command-line tool. Indexes are configured per server and index configuration is not replicated.

You can use `dsconfig` to create local database indexes and Virtual List View (VLV) indexes. A local database index is used to find entries that match search criteria. A VLV index is used to process searches efficiently with VLV controls.

Unindexed searches are denied by default, unless the user has the `unindexed-search` privilege. For more information, see Section 23.3.5, "To Change a Root User's Privileges."

You can determine whether a search is indexed in two ways:

- Try to perform the search anonymously. (The server rejects unindexed anonymous searches by default.)
- Use the `debugsearchindex` operational attribute. This attribute provides the indexes used in the search, the number of candidate entries from each index, and the final indexed status. Include the `debugsearchindex` attribute in your `ldapsearch` command, as follows:

```
$ ldapsearch -h localhost -p 1389 -b "dc=example,dc=com" "(objectClass=*)"
debugsearchindex
```

This section describes how to index data, and contains the following topics:

- Section 17.7.1, "Configuring Indexes on the Local DB Back End"
- Section 17.7.2, "Configuring VLV Indexes"

## 17.7.1 Configuring Indexes on the Local DB Back End

The Local DB back end supports the following index types:

- `approximate` — Improves the efficiency of searches using approximate search filters.
- `equality` - Improves the efficiency of searches using equality search filters.
- `ordering` - Improves the efficiency of searches using "greater than or equal to" or "less than or equal to" search filters. In the future, this index type might also be used for server-side sorting.
- `presence` - Improves the efficiency of searches using presence search filters.
- `substring` - Improves the efficiency of searches using substring search filters.

The directory server supports indexing for only a subset of extensible matching operations, including indexes based on collation matching rules and the relative time and partial date and time matching rules. For more information, see Section 17.5.5, "Searching Internationalized Entries," and Section 9.1.3, "Relative Time Matching Rules" and Section 9.1.4, "Partial Date Or Time Matching Rules."

When you create a new local DB back end with `dsconfig`, the following default indexes are created automatically:

- `aci` (presence index)
- `ds-sync-hist` (ordering index)
- `entryuuid` (equality index)
- `objectclass` (equality index)

This section contains the following topics:

- Section 17.7.1.1, "To Create a New Local DB Index"
- Section 17.7.2.1, "To Create a New VLV Index"

#### 17.7.1.1 To Create a New Local DB Index

This procedure demonstrates the steps for creating a new local DB index.

> **Note:** After you have created a new index, you must rebuild the indexes using the `rebuild-index` utility. The directory server cannot use the new index until the indexes have been rebuilt. For more information, see Appendix A.3.13, "rebuild-index."

1. Create the new index.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  create-local-db-index \
  --element-name backend --index-name attribute \
  --set index-type:index-type
```

2. Check that the index was created by listing the local DB indexes for that back end.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  list-local-db-indexes \
  --element-name backend
```

3. Configure any specific index properties.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  set-local-db-index-prop \
  --element-name backend --index-name attribute \
  --set property:value
```

4. List the index properties to verify your change.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  get-local-db-index-prop \
  --element-name backend --index-name attribute
```

5. Rebuild the index.

   a. Either stop the server, rebuild the index, then restart the server.

```
$ stop-ds
$ rebuild-index --baseDN baseDN --index attribute
$ start-ds
```

   b. Or, rebuild the index online by running the rebuild-index command as a task.

```
$ rebuild-index -h localhost -p 4444 -D "cn=Directory manager" -j pwd-file
\
 -X -n --baseDN dc=example,dc=com --index aci
Rebuild Index task 20110201162742312 scheduled to start immediately
...
Rebuild Index task 20110201162742312 has been successfully completed
```

   Note that, even for an online re-index operation, the backend is unavailable during the re-index. In a replicated topology, the overall service remains available through the referral on update feature. For more information, see Section 17.13.1, "Referrals in a Replicated Topology".

***Example 17–7 Creating a New Equality Index***

This example creates a new equality index for the `employeeNumber` attribute, verifies the index properties, and sets the index entry limit to 5000.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  create-local-db-index \
  --element-name userRoot --index-name employeeNumber \
  --set index-type:equality

$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  list-local-db-indexes \
  --element-name userRoot
Local DB Index : Type    : index-type
---------------:---------:-----------
...
employeeNumber : generic : equality
...
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  get-local-db-index-prop \
  --element-name userRoot --index-name employeeNumber
Property                      : Value(s)
------------------------------:---------------
attribute                     : employeenumber
index-entry-limit             : 4000
index-extensible-matching-rule : -
index-type                    : equality
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  set-local-db-index-prop \
  --element-name userRoot --index-name employeeNumber --set index-entry-limit:5000
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  get-local-db-index-prop \
  --element-name userRoot --index-name employeeNumber
Property                      : Value(s)
------------------------------:---------------
attribute                     : employeenumber
index-entry-limit             : 5000
index-extensible-matching-rule : -
index-type                    : equality
$ rebuild-index -h localhost -p 4444 -D "cn=Directory manager" -j pwd-file -X \
  --baseDN dc=example,dc=com --index employeeNumber
```

***Example 17–8 Adding a Substring Index***

This example adds a substring index to the index created in the previous example.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  set-local-db-index-prop \
  --'
p userRoot --index-name employeeNumber \
  --add index-type:substring
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  get-local-db-index-prop \
  --element-name userRoot --index-name employeeNumber
Property                      : Value(s)
------------------------------:---------------
attribute                     : employeenumber
index-entry-limit             : 5000
index-extensible-matching-rule : -
index-type                    : equality, substring
$ rebuild-index -h localhost -p 4444 -D "cn=Directory manager" -j pwd-file -X \
  --baseDN dc=example,dc=com --index employeeNumbe
```

## 17.7.2 Configuring VLV Indexes

A VLV index applies to a particular search on a given base entry and its subtree. The sort order, scope of the index, base DN, and filter must be defined when you create the index.

After you have created a new VLV index, you must rebuild the indexes using the `rebuild-index` command, appending `vlv.` in front of the index name. The directory server cannot use the new index until the indexes have been rebuilt. For more information, see Appendix A.3.13, "rebuild-index."

Note that access to the VLV request control is allowed to authenticated users only, by default. If you want to allow unauthenticated users to use the VLV control in search requests, you must change the corresponding global ACIs. For more information, see Section 17.5.3.16.4, "Allowing Anonymous Access to the Virtual List View Control".

### 17.7.2.1 To Create a New VLV Index

1. Use `dsconfig` to create a new VLV index as follows:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -n \
  create-local-db-vlv-index \
  --element-name backend --index-name name --set sort-order:attributes \
  --set scope:scope --set base-dn:baseDN --set filter:filter
```

   where:

   - `index-name` specifies a unique index name, which cannot be altered after the VLV index is created.
   - `sort-order` specifies the names of the attributes by which the entries are sorted and their order of precedence, from highest to lowest.
   - `scope` specifies the LDAP scope of the query being indexed and can be one of `base-object`, `single-level`, `subordinate-subtree`, or `whole-subtree`.
   - `base-dn` specifies the base DN used in the search query being indexed.
   - `filter` specifies the LDAP filter used in the query being indexed and can be any valid LDAP filter.

2. Check that the index was created by listing the existing VLV indexes.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -n \
  list-local-db-vlv-indexes \
  --element-name backend
```

3. Display the index properties to verify your change.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -n \
  get-local-db-vlv-index-prop \
  --element-name backend --index-name name
```

4. Rebuild the index.

   a. Either stop the server, rebuild the index, then restart the server.

```
$ stop-ds
$ rebuild-index --baseDN baseDN --index vlv.name
$ start-ds
```

   b. Or, rebuild the index online by running the rebuild-index command as a task.

```
$ rebuild-index -h localhost -p 4444 -D "cn=Directory manager" -j pwd-file
```

```
    -X \
      --baseDN baseDN --index vlv.name
```

***Example 17–9 Creating a New VLV Index***

The following example creates a new VLV index to sort entries first by surname and then by common name for queries `sn=*`. The example then rebuilds the index online.

```
$ dsconfig -D "cn=directory manager" -j pwd-file -n create-local-db-vlv-index \
  --element-name userRoot --index-name myVLVIndex --set sort-order:"sn cn" \
  --set scope:base-object --set base-dn:dc=example,dc=com --set filter:sn=*
$ rebuild-index -h localhost -p 4444 -D "cn=Directory manager" -j pwd-file -X \
  -b "dc=example,dc=com" --index vlv.myVLVIndex
```

## 17.8 Reducing Stored Data Size

The directory server provides two mechanisms for reducing the size of stored data:

- **Compact encoding**. When compact encoding is enabled, the back end uses a compact form when encoding entries by compressing the attribute descriptions and object class sets. This property applies only to the entries themselves and does not impact the index data. Compact encoding is enabled by default but can be disabled if required. If your deployment requires user-supplied capitalization in object class and attribute type names, you might want to disable compact encoding because user-supplied capitalization is not preserved in compacted entries. The compaction does, however, provide a performance gain and is therefore beneficial in deployments where user-supplied capitalization can be sacrificed for performance, or is not required.
- **Entry compression**. Entry compression uses a deflator to compress the data before it is stored. When entry compression is enabled, the back end attempts to compress entries before storing them in the database. This property also applies only to the entries themselves and does not impact the index data. The effectiveness of entry compression is based on the type of data contained in the entry.

You can enable one or both of these mechanisms to reduce the size of the stored data. Because enabling these mechanisms affects future writes only, the database might contain a mixture of compressed and uncompressed records. Either type of record can be read regardless of the compression settings.

This section describes the following topics:


- Section 17.8.1, "To Enable or Disable Compact Encoding"
- Section 17.8.2, "To Enable or Disable Entry Compression"


### 17.8.1 To Enable or Disable Compact Encoding

Compact encoding is configured by setting the `compact-encoding` property of a local backend workflow element. Changes to this setting will only take effect for writes that occur after the change is made. Existing data is not changed retroactively.

Disable compact encoding on the "userRoot" workflow element.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  set-workflow-element-prop --element-name="userRoot" --set compact-encoding:false
```

### 17.8.2 To Enable or Disable Entry Compression

Entry compression is configured by setting the `entries-compressed` property of a local backend workflow element. Changes to this setting will only take effect for writes that occur after the change is made. Existing data is not changed retroactively.

Enable entry compression on the "userRoot" back end.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  set-workflow-element-prop --element-name="userRoot" \
  --set entries-compressed:true
```

## 17.9 Ensuring Attribute Value Uniqueness

A directory's structure requires that distinguished names be unique to identify the object and its place in the directory information tree. The directory server provides a *Unique Attribute* plug-in, which ensures that the value of an attribute is unique when the attribute is added, modified, or moved within the directory.

This section describes the following topics:

- Section 17.9.1, "Overview of the Unique Attribute Plug-In"
- Section 17.9.2, "Configuring the Unique Attribute Plug-In Using `dsconfig`"
- Section 17.9.3, "Replication and the Unique Attribute Plug-In"

### 17.9.1 Overview of the Unique Attribute Plug-In

The unique attribute plug-in is disabled by default. You can enable the plug-in by using the `dsconfig` command and can define the suffix and attributes that it should check. When it is enabled, the plug-in identifies whether an LDAP add, modify, or modify DN operation causes two entries to have the same attribute value before the database is updated by the operation. If the server recognizes a conflict, the operation is terminated and an `LDAP_CONSTRAINT_VIOLATION` error is returned to the client.

When you enable attribute uniqueness on an existing directory, the server does not check for uniqueness among existing entries. After the plug-in is enabled, uniqueness is enforced when an entry is added, modified, or moved.

The unique attribute plug-in can be configured to enforce uniqueness in one or more subtrees in the directory or among entries of a specific object class. You can define several instances of the unique attribute plug-in if you want to enforce the uniqueness of other attributes. Typically, you define one plug-in instance for each attribute whose value must be unique. You can also have several plug-in instances for the same attribute to enforce "separate" uniqueness in several sets of entries.

The unique attribute plug-in is disabled by default, so that multi-master replication configuration is not affected. When the plug-in is enabled, it checks that the `uid` attribute is unique prior to any add, modify, or modify DN operations for stand-alone systems and checks for uniqueness after synchronization in replicated environments.

Like other plug-ins, the unique attribute plug-in is configured by using the `dsconfig` command. For more information, see Section 14.1.9, "Configuring Plug-Ins With `dsconfig`." The easiest way to configure plug-ins is to use `dsconfig` in interactive mode. Interactive mode functions like a wizard and walks you through the plug-in configuration. Because the interactive mode is self-explanatory, the examples in this section do not demonstrate interactive mode, but provide the equivalent complete `dsconfig` commands.

## 17.9.2 Configuring the Unique Attribute Plug-In Using `dsconfig`

The following procedures explain how to configure attribute value uniqueness.

### 17.9.2.1 To Ensure Uniqueness of the Value of the `uid` Attribute

The unique attribute plug-in checks the `uid` attribute by default. The following task enables the unique attribute plug-in, and sets the base DN under which attribute value uniqueness for the `uid` attribute should be checked.

1. Display the plug-ins that are currently defined in the server.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -n \
  list-plugins
```

Depending on your installation, the output will be similar to the following.

```
Plugin                          : Type                            : enabled
--------------------------------:---------------------------------:--------
7-Bit Clean                     : seven-bit-clean                 : false
Change Number Control           : change-number-control           : true
Entry UUID                      : entry-uuid                      : true
LastMod                         : last-mod                        : true
LDAP Attribute Description List : ldap-attribute-description-list : true
Password Policy Import          : password-policy-import          : true
Profiler                        : profiler                        : true
Referential Integrity           : referential-integrity           : false
UID Unique Attribute            : unique-attribute                : false
```

2. Display the properties that are configured for the unique attribute plug-in

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -n \
  get-plugin-prop \
  --plugin-name "UID Unique Attribute" \
Property : Value(s)
---------:---------
base-dn  : -
enabled  : false
type     : uid
```

3. Enable the unique attribute plug-in.

```
$ dsconfig --advanced -h localhost -p 4444 -D "cn=directory manager" -j
pwd-file -n \
  set-plugin-prop \
  --plugin-name "UID Unique Attribute" --set enabled:true
```

> **Note:** Ensure that you run the `dsconfig` command with `--advanced` subcommand. This subcommand modifies the display output to show the advanced plug-ins like `postaddoperation`, `postmodifyoperation`, and `postmodifydnoperation` that can be selected. The default values are pre-operation plug-ins like `preaddoperation`, `premodifyoperation`, and `postmodifyoperation`. You must select a matching pre-operation plug-in with a post-operation plug-in.

4. Set the base DN under which uniqueness is checked.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -n \
```

```
  set-plugin-prop \
  --plugin-name "UID Unique Attribute" --set
base-dn:ou=People,dc=example,dc=com
```

#### 17.9.2.2 To Ensure Uniqueness of the Value of Any Other Attribute

The unique attribute plug-in checks the `uid` attribute by default. If you want to ensure uniqueness for a different attribute, create a new instance of the unique attribute plug-in and set its `type` property.

This example creates a new instance of the unique attribute plug-in and ensures uniqueness of the `mail` attribute.

1. Create and enable a new instance of the unique attribute plug-in.

   Set the `type` property to the name of the attribute that should be unique (in this case, `mail`.

   ```
   $ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -n \
     create-plugin \
     --type unique-attribute --plugin-name "MAIL unique attribute"
     --set enabled:true --set type:mail
   ```

2. Enable the new unique attribute plug-in.

   ```
   $ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -n \
     set-plugin-prop \
     --plugin-name "MAIL Unique Attribute" --set enabled:true
   ```

3. Set the base DN under which uniqueness is checked.

   ```
   $ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -n \
     set-plugin-prop \
     --plugin-name "MAIL Unique Attribute" --set
   base-dn:ou=People,dc=example,dc=com
   ```

4. Specify the attribute whose value must be unique.

   This example specifies the `mail` attribute.

   ```
   $ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -n \
     set-plugin-prop \
     --plugin-name "MAIL Unique Attribute" --set type:mail
   ```

To ensure that the values of more than one attribute are unique, create and enable multiple instances of the unique attribute plug-in.

### 17.9.3 Replication and the Unique Attribute Plug-In

The Unique Attribute plug-in does not check attribute uniqueness when an update is performed as part of a replication operation. To ensure attribute value uniqueness in a replication environment, enable the unique attribute plug-in for the same attribute in the same subtree on all servers in the topology.

## 17.10 Configuring Virtual Attributes

*Virtual attributes* are attributes whose values do not exist in persistent storage but are dynamically generated in some way.

Oracle Unified Directory supports the following virtual attribute types:

- `collective attribute subentries`

- entryDN
- entryUUID
- governingStructureRule
- hasSubordinates
- isMemberOf
- member
- nsuniqueid
- numSubordinates
- orclguid
- Password Policy Subentry
- Proximity
- structuralObjectClass
- subschemaSubentry
- User-defined virtual attributes

Virtual attributes are configured by using the `dsconfig` command. The `dsconfig` command accesses the plug-in configuration over SSL via the Section 14.3, "Managing Administration Traffic to the Server." The easiest way to configure virtual attributes is to use `dsconfig` in interactive mode. Interactive mode functions like a wizard and walks you through the virtual attribute configuration. Because the interactive mode is self-explanatory, the examples in this section do not demonstrate interactive mode, but provide the equivalent complete `dsconfig` commands.

For more information about using `dsconfig`, see Section 14.1, "Managing the Server Configuration With `dsconfig`."

This section describes the following topics:

- Section 17.10.1, "To List the Existing Virtual Attributes"
- Section 17.10.2, "To Create a New Virtual Attribute"
- Section 17.10.3, "To Enable or Disable a Virtual Attribute"
- Section 17.10.4, "To Display the Configuration of a Virtual Attribute"
- Section 17.10.5, "To Change the Configuration of a Virtual Attribute"

## 17.10.1 To List the Existing Virtual Attributes

The directory server provides a number of virtual attribute rules by default. This example lists all configured virtual attribute rules.

Run the `dsconfig` command as follows:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -n list-virtual-attributes

Virtual Attribute               : Type                            : enabled : attribute-type
--------------------------------:---------------------------------:---------:------------------
Collective Attribute Subentries : collective-attribute-subentries : true    : collectiveattributesubentries
entryDN                         : entry-dn                        : true    : entrydn
entryUUID                       : entry-uuid                      : true    : entryuuid
governingStructureRule          : governing-structure-rule        : true    : governingstructurerule
hasSubordinates                 : has-subordinates                : true    : hassubordinates
isMemberOf                      : is-member-of                    : true    : ismemberof
nsuniqueid                      : nsuniqueid                      : true    : nsuniqueid
```

```
numSubordinates                : num-subordinates               : true    : numsubordinates
orclguid                       : orclguid                       : true    : orclguid
Password Policy Subentry       : password-policy-subentry       : true    : pwdpolicysubentry
Proximity                      : proximity                      : true    : proximity
structuralObjectClass          : structural-object-class        : true    : structuralobjectclass
subschemaSubentry              : subschema-subentry             : true    : subschemasubentry
Virtual Static member          : member                         : true    : member
Virtual Static uniqueMember    : member                         : true    : uniquemember
```

The output of this command shows the following (from left to right):

- `Virtual Attribute`. The name of the virtual attribute, usually descriptive of what it does.
- `Type`. The type of virtual attribute. It is possible to define more then one virtual attribute of a specific type.
- `enabled`. Virtual attributes can either be enabled or disabled. Disabled virtual attributes remain in the server configuration, but their values are never generated.
- `attribute-type`. Specifies the type of attribute for which virtual values are generated.

## 17.10.2 To Create a New Virtual Attribute

This example creates and enables a virtual attribute rule that adds a virtual fax number of `+61 2 45607890` to any user entry with a location of Sydney (unless they already have a fax number in their entry):

Run the `dsconfig` command as follows:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -n \
  create-virtual-attribute \
  --type user-defined --name "Sydney Fax Number" \
  --set attribute-type:facsimiletelephonenumber --set enabled:true \
  --set value:+61245607890 --set filter:"(&(objectClass=person)(l=Sydney))"
```

## 17.10.3 To Enable or Disable a Virtual Attribute

To enable a virtual attribute, set the `enabled` property to `true`. To disable a virtual attribute, set the `enabled` property to `false`. This example disables the virtual attribute created in the previous example:

Run the `dsconfig` command as follows:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -n \
  set-virtual-attribute-prop --name="Sydney Fax Number" --set enabled:false
```

## 17.10.4 To Display the Configuration of a Virtual Attribute

Use the `get-*-prop` subcommand of `dsconfig` to display the virtual attribute configuration. This example displays the properties of the virtual attribute created in the previous example:

Run the `dsconfig` command as follows:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -n \
  get-virtual-attribute-prop --name="Sydney Fax Number"
Property          : Value(s)
------------------:---------------------------------
attribute-type    : facsimiletelephonenumber
base-dn           : -
conflict-behavior : real-overrides-virtual
```

```
enabled           : false
filter            : (&(objectClass=person)(l=Sydney))
group-dn          : -
value             : +61245607890
```

### 17.10.5 To Change the Configuration of a Virtual Attribute

Use the `set-*-prop` subcommand of `dsconfig` to change the virtual attribute configuration. This example changes the behavior of the virtual attribute if a conflict occurs. By default, the value of a real attribute overwrites the value of the virtual attribute. With this change, the value of the real attribute and that of the virtual attribute are merged.

Run the `dsconfig` command as follows:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  set-virtual-attribute-prop --name="Sydney Fax Number" \
  --set conflict-behavior:merge-real-and-virtual
```

## 17.11 Using LDAP Subentries

LDAP subentries are special entries that hold operational data for the server, and have the `ldapSubEntry` object class. They are similar to operational attributes in that they are not returned to clients unless explicitly requested by including a Subentries Control request control.

LDAP subentries can be used to specify a range of entries. This functionality is used in the definition of collective attributes and can also be useful in other areas like access control. For more information, see Section 17.12, "Using Collective Attributes" and Section 24.4.7, "To Define a Password Policy as an LDAP Subentry".

A subtree specification uses the following parameters to define the set of entries:

- Base

  This is the relative name of the root of the subtree relative to the administrative point. So, if the administrative point is `ou=system` and the base is `ou=users`, the subtree begins at `ou=users,ou=system`. The base can be any length of name components, including "". In this case, the subtree begins at the administrative point, `ou=system` in the previous example.

- Chop

  The `chopBefore` and `chopAfter` parameters are names relative to the base of the subtree, that specify whether an entry and its descendants should be excluded from the collection.

  The `minimum` parameter describes the minimum number of name components between the base and the target entry required to include entries within the selection. The `maximum` parameter describes the maximum length between the base and the target allowed before entries are excluded from the collection.

- Specification filter

  The specification filter refines the subtree that has been defined by the previous parameters so that it is not a contiguous set of entries but rather a set of collected entries based on the `objectClass` characteristics of the entries.

  For example, you can define a subtree to cover a region of an administrative area but include only `inetOrgPersons` within this region.

The Oracle Unified Directory implementation of LDAP subentries is based on RFC 3672 (http://www.ietf.org/rfc/rfc3672.txt), with one extension - relative subtrees, described in the following section.

### 17.11.1 Relative Subtrees

Relative subtrees function like standard LDAP subtrees, with the exception that the specification filter is not a set of refinements but an LDAP search filter.

For relative subtree specification ensure that you use the relativeBase keyword to specify the root of the subtree. Do not use the base keyword to specify the root of the subtree.

For example, the following subtree definition targets all users under the base DN `ou=People`, whose location is Paris:

```
subtreeSpecification: {relativeBase "ou=people", specificationFilter "(l=Paris)" }
```

## 17.12 Using Collective Attributes

Collective attributes are attributes whose values are shared across a collection of entries. Collective attributes provide similar functionality to the Oracle Directory Server Enterprise Edition Class of Service feature.

Oracle Unified Directory collective attributes are like virtual attributes but are defined and stored with the user data as LDAP subentries. As part of the user data, collective attributes can be replicated to other servers in the topology.

This section describes the collective attribute implementation in Oracle Unified Directory and explains how to configure collective attributes. The section covers the following topics:

- Section 17.12.1, "Extensions to the Collective Attributes Standard"
- Section 17.12.2, "Configuring Collective Attributes"
- Section 17.12.3, "Inherited Collective Attributes"

### 17.12.1 Extensions to the Collective Attributes Standard

The Oracle Unified Directory implementation of collective attributes is based on RFC 3671 (http://www.ietf.org/rfc/rfc3671.txt) and RFC 3672 (http://www.ietf.org/rfc/rfc3672.txt), with a few specific extensions. These extensions make Oracle Unified Directory collective attributes more transparent for LDAP client applications, and are described in the following sections:

- Section 17.12.1.1, "Naming Collective Attributes"
- Section 17.12.1.2, "Collective Attributes and Conflict Resolution"
- Section 17.12.1.3, "Excluding Collective Attributes From Specific Entries"

#### 17.12.1.1 Naming Collective Attributes

According to RFC 3671 (http://www.ietf.org/rfc/rfc3671.txt), collective attributes *must* have the COLLECTIVE attribute type, be derived from regular user attributes defined in the schema, and have the `c-` prefix. For example `c-l` is a collective attribute for the standard `l` attribute, and affected user entries have `c-l` added to them on the fly.

This specification can cause problems for many client applications, which are typically not aware of collective attributes and might need to be modified or extended to handle collective attributes. Oracle Unified Directory therefore removes this restriction and supports the definition of any regular attribute defined in the schema as a collective attribute. This extension is facilitated by adding the required attribute to the related collective attribute subentry and marking the attribute with the collective option.

#### 17.12.1.2 Collective Attributes and Conflict Resolution

Collective attributes can be named in various ways. Due to this, a conflict resolution mechanism is provided for affected user entries already containing related real attributes. Oracle Unified Directory provides the same conflict resolution options for collective attributes as it does for virtual attributes: `real-overrides-virtual`, `virtual-overrides-real`, and `merge-real-and-virtual`.

The default conflict resolution rule is `real-overrides-virtual`. If an entry already has the same attribute type defined, the explicitly defined attribute takes precedence over the collective attribute. This behavior can be changed for each collective attribute subentry (to `virtual-overrides-real` or `merge-real-and-virtual`) by using the `collectiveConflictBehavior` attribute.

The following example dynamically adds the `l` collective attribute with a value of `Paris` to each applicable user entry under `ou=people`. The value of the collective attribute overrides any value for `l` that is specific to the entry:

```
dn: cn=People Locale,dc=example,dc=com
objectClass: top
objectClass: subentry
objectClass: collectiveAttributeSubentry
objectClass: extensibleObject
cn: People Locale
l;collective: Savoie
subtreeSpecification: {base "ou=people", minimum 1}
collectiveConflictBehavior: virtual-overrides-real
```

#### 17.12.1.3 Excluding Collective Attributes From Specific Entries

In some instances, it might be necessary to avoid having collective attributes in specific user entries. You can add the `collectiveExclusions` operational attribute to such entries to achieve this behavior. To exclude specific collective attributes, list the attribute names as values of the `collectiveExclusions` attribute. To exclude all collective attributes, set the value of `collectiveExclusions` to `excludeAllCollectiveAttributes`.

The following example excludes the `preferredLanguage` attribute from being applied to the entry for `user.0`:

```
dn: uid=user.0,ou=People,dc=example,dc=com
objectclasses and other user attributes
collectiveExclusions: preferredLanguage
```

The following example excludes the `c-l` attribute from being applied to the entry for `user.1`:

```
dn: uid=user.1,ou=People,dc=example,dc=com
objectclasses and other user attributes
collectiveExclusions: c-l
```

The following example excludes both the `preferredLanguage` and `c-l` attributes from being applied to the entry for `user.2`:

```
dn: uid=user.2,ou=People,dc=example,dc=com
objectclasses and other user attributes
collectiveExclusions: preferredLanguage
collectiveExclusions: c-l
```

The following example excludes all collective attributes from being applied to the entry for `user.0`:

```
dn: uid=user.0,ou=People,dc=example,dc=com
objectclasses and other user attributes
collectiveExclusions: excludeAllCollectiveAttributes
```

## 17.12.2 Configuring Collective Attributes

Collective attributes are defined using LDAP subentries within the directory tree where they are applicable. The following examples use a simple tree with multiple user entries.

```
dn: dc=example,dc=com
    dn: ou=People,dc=example,dc=com
        dn: uid=user.0,ou=People,dc=example,dc=com
        dn: uid=user.1,ou=People,dc=example,dc=com
        dn: uid=user.2,ou=People,dc=example,dc=com
        ...
```

To add a common `preferredLanguage` attribute for all users, create and add a collective attribute subentry similar to the following:

```
dn: cn=People Preferred Language,dc=example,dc=com
objectClass: top
objectClass: subentry
objectClass: collectiveAttributeSubentry
objectClass: extensibleObject
cn: People Preferred Language
preferredLanguage;collective: fr
subtreeSpecification: {base "ou=people", minimum 1}
```

The `preferredLanguage` attribute-value pair is dynamically added to all user entries under `ou=people`, as shown in the following example:

```
dn: uid=user.0,ou=People,dc=example,dc=com
objectclasses and other user attributes
preferredLanguage: fr

dn: uid=user.1,ou=People,dc=example,dc=com
objectclasses and other user attributes
preferredLanguage: fr

...
```

The same procedure applies for *collective* attribute types. For example, the `c-l` collective attribute type specifies a locality name for a collection of entries. The following example adds a common `c-l` collective attribute:

```
dn: cn=People Locale,dc=example,dc=com
objectClass: top
objectClass: subentry
objectClass: collectiveAttributeSubentry
objectClass: extensibleObject
cn: People Locale
c-l: Paris
```

```
subtreeSpecification: {base "ou=people", minimum 1}
```

The `c-l: Paris` attribute is added to applicable entries, as shown in this example:

```
dn: uid=user.0,ou=People,dc=example,dc=com
objectclasses and other user attributes
c-l: Paris

dn: uid=user.1,ou=People,dc=example,dc=com
objectclasses and other user attributes
c-l: Paris

...
```

You can define multiple collective attributes in the subentry of any collective attribute in the following ways:

- by adding the *collective* attribute types to the subentry
- by adding regular attribute types with the collective option
- by adding a combination of the two

Collective attribute subentries allow for flexible and complex definitions. For information about collective attribute scoping and the `subtreeSpecification` syntax, see RFC 3671 (`http://www.ietf.org/rfc/rfc3671.txt`) and RFC 3672 (`http://www.ietf.org/rfc/rfc3672.txt`).

This section describes the following topics about collective attributes:

- Section 17.12.2.1, "To Create a New Collective Attribute"
- Section 17.12.2.2, "To Delete a Collective Attribute"
- Section 17.12.2.3, "To List the Collective Attributes That Apply to an Entry"

### 17.12.2.1 To Create a New Collective Attribute

1. Create an LDIF file with the `changetype: add` element that specifies the collective attribute subentry.

   Make sure that there are no trailing spaces after `add`. If a space exists after add, the server base-64 encodes the value to represent the space, which can cause problems.

   This example uses an input LDIF file named `add_collective_attr.ldif`.

   ```
   dn: cn=People Preferred Language,dc=example,dc=com
   changetype: add
   objectClass: top
   objectClass: subentry
   objectClass: collectiveAttributeSubentry
   objectClass: extensibleObject
   cn: People Preferred Language
   preferredLanguage;collective: fr
   subtreeSpecification: {base "ou=people", minimum 1}
   ```

2. Use the `ldapmodify` command to add the collective attribute, as shown in the following example.

   ```
   $ ldapmodify -p 1389 -h localhost -D "cn=Directory Manager" -j pwd-file \
   -f /usr/local/add_collective_attr.ldif
   Processing ADD request for cn=People Preferred Language,dc=example,dc=com
   ADD operation successful for DN cn=People Preferred Language,dc=example,dc=com
   ```

#### 17.12.2.2 To Delete a Collective Attribute

You can delete a collective attribute by using either the `ldapdelete` command or the `ldapmodify` command. This example uses the `ldapmodify` command.

Use the `ldapmodify` command with the `changetype: delete` element, as shown in the following example.

```
$ ldapmodify -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file
  dn: cn=People Preferred Language,dc=example,dc=com
  changetype: delete
  deleting entry cn=People Preferred Language,dc=example,dc=com
```

#### 17.12.2.3 To List the Collective Attributes That Apply to an Entry

To list the collective attribute subentries that apply to a specific user entry, request the `collectiveAttributeSubentries` operational attribute for that entry.

Use the `ldapsearch` command to list the collective attribute subentries that apply to the `user.0` entry:

```
$ ldapsearch -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
  -b "uid=user.0,ou=People,dc=example,dc=com" \
  "objectclass=*" "collectiveAttributeSubentries"

  version: 1
  dn: uid=user.0,ou=People,dc=example,dc=com
  collectiveAttributeSubentries: cn=People Preferred Language,dc=example,dc=com
```

### 17.12.3 Inherited Collective Attributes

Inherited attributes enable a common set of attributes to be shared by nature of their inheritance. Inherited collective attributes provide flexible scoping mechanisms using the standard subentry subtree specification, and support any attribute type for RDN definition and construction.

The main difference between collective attributes and inherited collective attributes is the source of attribute values:

- A collective attribute always derives its value from its definition entry.
- An inherited collective attribute can inherit the collective attribute values from other entities, either directly or indirectly.

The inherited collective attributes functionality is built upon and extends collective attributes. Inherited attributes are defined as a specific type of collective attribute subentry (`inheritedCollectiveAttributeSubentry`). This type is further divided into the following two distinct subtypes:

- `inheritedFromDNCollectiveAttributeSubentry`
- `inheritedFromRDNCollectiveAttributeSubentry`

Each subtype has its own set of configuration attributes. The subtypes cannot be mixed in a single definition, so an inherited attribute definition can be of only one subtype.

Entries that are under the scope of an inherited collective attribute entry can potentially point to multiple "template" entries and can therefore inherit values for the inheritAttribute from multiple entries. In this case, the first value that is processed takes precedence.

As with other virtual attributes, no schema checking is performed on inherited attributes. Inheritance can, therefore, result in entries that violate the schema.

However, since these attributes are all virtual, this kind of schema violation can be ignored as it does not have an impact on server function.

Inherited collective attributes provide similar functionality to the Oracle Directory Server Enterprise Edition Class of Service (Classic CoS). For example, suppose you have the following user entry:

```
uid=psmith,ou=people,dc=example,dc=com
departmentNumber: 123
...
```

the following department entry:

```
cn=123,ou=departments,dc=example,dc=com
telephoneNumber: 4486152643
...
```

and the following inherited attribute definition:

```
dn: cn=classicCOS,dc=example,dc=com
objectClass: top
objectClass: subentry
objectClass: inheritedCollectiveAttributeSubentry
objectClass: inheritedFromRDNCollectiveAttributeSubentry
cn: classicCOS
subtreeSpecification: {base "ou=people"}
inheritFromBaseRDN: ou=departments
inheritFromRDNAttribute: departmentNumber
inheritFromRDNType: cn
inheritAttribute: telephoneNumber
```

The inherited collective attribute sub-entry would apply to user entries under `ou=people,dc=example,dc=com`. The `telephoneNumber` attribute would be added to each of these entries. The value of the `telephoneNumber` attribute would be inherited from the entry whose DN is constructed with the following logic:

*inheritFromRDNType=inheritFromRDNAttribute,inheritFromBaseRDN,"inherited collective attribute sub-entry rootDN*"

or cn=123,ou=departments,dc=example,dc=com

The affected user entries would therefore be of the form:

```
uid=psmith,ou=people,dc=example,dc=com
departmentNumber: 123
...
telephoneNumber: 4486152643
```

#### 17.12.3.1 Specifying Inherited Collective Attributes

Like regular collective attributes, inherited collective attributes are defined using LDAP subentries within the directory tree where they are applicable.

The following examples use a simple tree with multiple user entries.

```
dn: dc=example,dc=com
     dn: ou=People,dc=example,dc=com
         dn: uid=hpollock,ou=People,dc=example,dc=com
         dn: uid=cventer,ou=People,dc=example,dc=com
         dn: uid=sdonnelly,ou=People,dc=example,dc=com
         ...
```

To add an inherited `postalAddress` attribute for all users, create and add an inherited collective attribute subentry similar to the following:

```
dn: cn=indirectCOS,dc=example,dc=com
objectClass: top
objectClass: subentry
objectClass: inheritedCollectiveAttributeSubentry
objectClass: inheritedFromDNCollectiveAttributeSubentry
cn: indirectCOS
subtreeSpecification: {base "ou=people"}
inheritFromDNAttribute: manager
inheritAttribute: postalAddress
```

This subentry specifies that the user entry inherits its `postalAddress` value from the entry referenced by the `manager` attribute in the user's entry.

The manager's entry contains the real value for the `postalAddress` attribute:

```
dn: uid=dsmith,ou=People,dc=example,dc=com
... objectclasses and other user attributes
postalAddress: 650 Granger Parkway, Redwood Shores, CA 94065
```

Each user entry references the manager entry, and inherits its `postalAddress` from that entry:

```
dn: uid=hpollock,ou=People,dc=example,dc=com
... objectclasses and other user attributes
manager: uid=dsmith,ou=People,dc=example,dc=com
postalAddress: 650 Granger Parkway, Redwood Shores, CA 94065

dn: uid=cventer,ou=People,dc=example,dc=com
... objectclasses and other user attributes
manager: uid=dsmith,ou=People,dc=example,dc=com
postalAddress: 650 Granger Parkway, Redwood Shores, CA 94065

dn: uid=sdonnelly,ou=People,dc=example,dc=com
... objectclasses and other user attributes
manager: uid=dsmith,ou=People,dc=example,dc=com
postalAddress: 650 Granger Parkway, Redwood Shores, CA 94065
```

## 17.13 Configuring Referrals

A *referral* is a pointer to a remote suffix or entry that is returned to a client instead of a result. When a server is unable to handle a client's request, it sends a list of referrals to the client, which point the client to other servers in the topology. The client then performs the operation again on one of the remote servers in the referral list.

The server returns a list of referrals in the following cases:

- Writability is disabled or set to `internal-only` on the server or on the local backend workflow element. For more information, see Section D.22.6, "writability mode".

  This kind of referral is called *referral on update*.

- The local backend workflow element has been placed in maintenance mode.

  You can place a local backend workflow element in maintenance mode if you want to prevent the server from responding to client requests temporarily.

  To place a backend in maintenance mode, set the `maintenance` property of the local backend workflow element to `true`.

- The backend is unavailable for some reason, for example a data import or reindex is in process.
- The client request specifically targets a *smart referral*. For more information, see Section 17.13.3, "Smart Referrals".

A referral URL is an LDAP URL that includes the host name, port number, and optionally a DN on the local host or on another server. For more information, see Section 17.13.4, "LDAP URLs".

The server returns the result code `REFERRAL (10)` along with a list of referral URLs, if available. If no referral URLs are available, the server returns the result code `UNAVAILABLE (52)`.

The list of referral URLs can be created in two ways:

- For replicated servers, use the replication service to propagate the list. For more information, see Section 17.13.1, "Referrals in a Replicated Topology".
- Create the list manually by setting the `ds-cfg-referrals-url` property of the DB local backend workflow element. For more information, see Section 17.13.2, "Configuring the Referral List Manually".

## 17.13.1 Referrals in a Replicated Topology

The replication service generates a list of referral URLs to which requests can be redirected. This list corresponds to the LDAP/LDAPS connection handlers configured on each local server. To publish a value other than the LDAP/LDAPS connection handler, you can define your own referral URLs as values of the `referrals-url` property of the replication domain on the local server.

When a client request targets a replicated server that is unavailable, the server sends the list of referral URLs to which the request can be redirected.

The list of referral URLs is organized according to the protocol that was used for the request. For example, if an operation is done over LDAPS, the first URLs that are provided are those that use the same secure protocol (LDAPS).

In addition, the list is organized by groupID. The URLs that represent a server in the same replication group are presented first. The list of URLs is limited to 16 URLs for each protocol type (LDAP/LDAPS) and excludes any untrusted servers.

For security considerations, referrals that are propagated by the replication service are not returned on untrusted servers. Untrusted servers should not divulge information about the rest of the topology. If a client request targets an untrusted server, the list of referral URLs will only include the servers that are managed by the administrator on the local backend. In addition, the referral URLs that are provided by the replication service exclude any untrusted servers in the topology.

If the `publish-referrals` configuration property of a replication domain is set to false, that server will not be included in the list of referrals that is generated by the replication service.

## 17.13.2 Configuring the Referral List Manually

To override the list of referral URLs that is presented by the replication service, or to set up referrals outside of a replicated topology, set the `referrals-url` property of the DB local backend workflow element.

The `referrals-url` property takes one or more LDAP URLs as values.

The following example specifies that any client requests targeting the `dc=example,dc=com` suffix should be referred to the server running on the host `host1.example.com` and listening on port 2389.

```
$dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
 set-workflow-element-prop --element-name userRoot \
 --set referrals-url:ldap://host1.example.com:2389/dc=example,dc=com
```

To specify multiple LDAP URLs, use the `--add` suboption multiple times. For example:

```
$dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
 set-workflow-element-prop --element-name userRoot \
 --add referrals-url:ldap://host1.example.com:2389/dc=example,dc=com
 --add referrals-url:ldap://host2.example.com:1389/dc=example,dc=com
```

## 17.13.3 Smart Referrals

A smart referral is a special type of entry that references content on another server or in another suffix. Smart referral entries contain the **referral** object class with one or more instances of the `ref` attribute. Each `ref` attribute contains an LDAP URL that is used in the referral.

### 17.13.3.1 To Configure a Smart Referral

To configure a smart referral, add a new entry that contains a `referral` object class and a `ref` attribute. The `ref` attribute must contain an LDAP URL.

This example creates a referral on server B for a user entry that exists on server A.

1. Locate the user entry on server A by running the following search command:

```
$ ldapsearch -h serverA -p 1389 -b dc=example,dc=com "uid=user.199" cn
dn: uid=user.199,ou=People,dc=example,dc=com
cn: Alfred Altay
```

2. Add a referral entry to the directory on server B.

```
$ ldapmodify -h serverB -p 2389 -D "cn=directory manager" -j pwd-file
dn: uid=aaltay,ou=People,dc=example,dc=com
changetype: add
objectclass: top
objectclass: extensibleObject
objectclass: referral
uid: aaltay
ref: ldap://serverA:1389/dc=example,dc=com??sub?(uid=user.199)

Processing ADD request for uid=aaltay,ou=People,dc=example,dc=com
ADD operation successful for DN uid=aaltay,ou=People,dc=example,dc=com
```

3. As a user with sufficient access rights, search for the user entry on server B.

```
$ ldapsearch -h serverB -p 2389 -D "cn=directory manager" -j pwd-file \
  -b dc=example,dc=com "uid=aaltay"
SearchReference(referralURLs={ldap://localhost:1389/dc=example,dc=com??sub?})
```

### 17.13.3.2 To Modify a Smart Referral

To view or modify a smart referral, use the `ldapsearch` or `ldapmodify` commands with the `manageDsaIT` control. This control informs the server that you intend to manage the referral object as a regular entry and prevents the server from sending a referral result for requests that read or update referral objects.

1. Use the `ldapsearch` command to view the referral.

```
$ ldapsearch -h serverB -p 2389 -D "cn=Directory Manager" -j pwd-file \
  -b dc=example,dc=com --control managedsait "(uid=aaltay)" ref
dn: uid=aamar,ou=People,dc=example,dc=com
ref: ldap://serverA:1389/dc=example,dc=com??sub?(uid=user.199)
```

2. Use the `ldapmodify` command to modify the referral.

   This example changes the server to which the referral points and the base DN under which the entry is located.

```
$ ldapmodify -h serverB -p 2389 -D "cn=Directory Manager" -j pwd-file \
  --control managedsait
dn: uid=aaltay,ou=People,dc=example,dc=com
changetype: modify
replace: ref
ref: ldap://serverC:1389/ou=People,dc=example,dc=com??sub?(uid=user.199)
Processing MODIFY request for uid=aaltay,ou=People,dc=example,dc=com
MODIFY operation successful for DN uid=aaltay,ou=People,dc=example,dc=com
```

#### 17.13.3.3 To Delete a Smart Referral

To delete a smart referral, use the `ldapdelete` command with the `manageDsaIT` control. This control informs the server that you intend to manage the referral object as a regular entry and prevents the server from sending a referral result for requests that read or update referral objects.

1. Use the `ldapsearch` command to view the referral.

```
$ ldapsearch -h serverB -p 2389 -D "cn=Directory Manager" -j pwd-file \
  -b dc=example,dc=com --control managedsait "(uid=aaltay)" ref
dn: uid=aamar,ou=People,dc=example,dc=com
ref: ldap://serverA:1389/dc=example,dc=com??sub?(uid=user.199)
```

2. Use the `ldapdelete` command to delete the referral.

```
$ ldapdelete -h serverB -p 2389 -D "cn=Directory Manager" -j pwd-file \
  --control managedsait "uid=aaltay,ou=People,dc=example,dc=com"
Processing DELETE request for uid=aaltay,ou=People,dc=example,dc=com
DELETE operation successful for DN uid=aaltay,ou=People,dc=example,dc=com
```

### 17.13.4 LDAP URLs

The format of an LDAP URL is described in RFC 4516 and is summarized as follows:

```
ldap[s]://hostname:port/base_dn?attributes?scope?filter
```

An LDAP URL includes the following components:

`ldap[s]`
Indicates whether to connect to the server (`ldap:`), or connect to the server over SSL (`ldaps:`).

***hostname***
Specifies the host name or IP address of the LDAP server.

***port***
Specifies the port number of the LDAP server. If no port is specified, the default LDAP port (389) or LDAPS port (636) is used.

***base_dn***
Specifies the distinguished name (DN) of an entry in the directory. This DN identifies the entry that is the starting point of the search. If no base DN is specified, the search starts at the root of the directory tree.

***attributes***
Returns the specified attributes. Use commas to separate more than one attribute. If no attributes are specified, the search returns all attributes.

***scope***
Specifies the scope of the search:

- **base.** Search only the base entry specified by *base_dn*.
- **one.** Search one level below the base entry specified by base_dn
- **sub.** Search the base entry and all entries below the specified base_dn

If no scope is specified, the server performs a base search.

***filter***
Specifies the search filter to apply to entries within the specified scope of the search. If no filter is specified, the server uses the default (`objectclass=*`).

Any spaces must be escaped using a character appropriate to your shell.

---

**Note:** Unless an LDAP client provides authentication, any search request initiated by means of an LDAP URL is anonymous (unauthenticated).

---

#### 17.13.4.1 Example LDAP URLs

- The following LDAP URL specifies a search for all entries that have the surname `Jensen` at any level under `dc=example,dc=com`. No port is specified, so the default (`389`) is used. No attributes are specified, so all attributes will be returned.

  ```
  ldap://example.com/dc=example,dc=com??sub?(sn=Jensen)
  ```

- The following LDAP URL specifies a search for the `cn` and `telephoneNumber` attributes at any level under `dc=example,dc=com`. The server contacts the remote server at port `2389`. Because no search filter is specified, the server uses the default filter `(objectclass=*)`.

  ```
  ldap://example.com:2389/dc=example,dc=com?cn,telephoneNumber?sub
  ```

## 17.14 Managing Virtual Attributes With Oracle Directory Services Manager

The Configuration tab of each server instance in ODSM enables you create, delete, and modify virtual attributes.

The following sections describe how to manage virtual attributes with ODSM, and contains the topics:

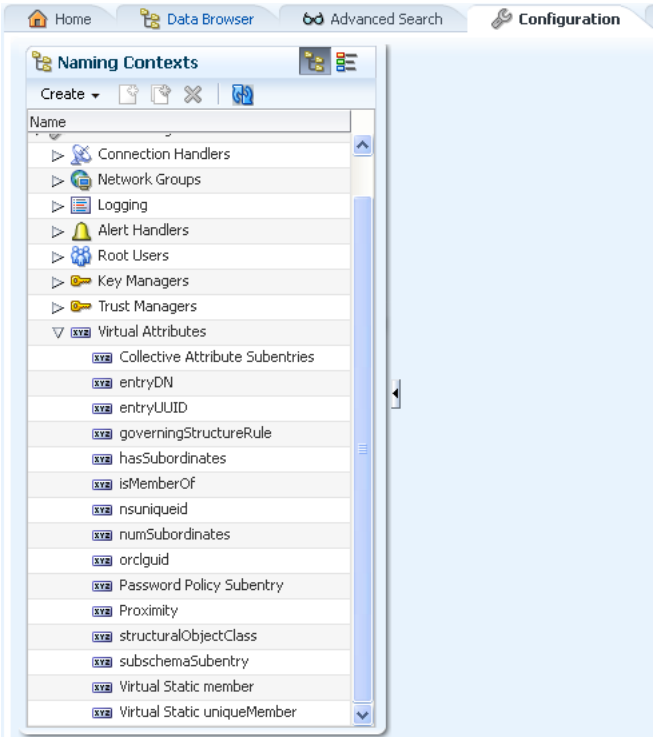
- Section 17.14.1, "Display Existing Virtual Attributes"
- Section 17.14.2, "Create Virtual Attributes"

## 17.14.1 Display Existing Virtual Attributes

To display the existing virtual attributes with ODSM, perform the following steps:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager."
2. Select the **Configuration** tab.
3. Expand the **General Configuration** node.
4. Expand the attributes in the **Virtual Attributes** node to display all the existing virtual attributes.

***Figure 17–1 Virtual Attributes***




5. Click the virtual attribute that you want to view in the left-hand pane.

   The virtual attribute details are displayed in the right-hand pane.

## 17.14.2 Create Virtual Attributes

To create a virtual attribute using ODSM, perform the following steps:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager."
2. Select the **Configuration** tab.
3. From the Create menu, select **Virtual Attributes.**
4. In the Name field, type the name of the virtual attribute.
5. From the Virtual Attribute Type list, select the type of virtual attribute.

6. Click the **Add** icon to enter the Base DN for the branches containing entries that are eligible to use this virtual attribute.

   Do one of the following to enter the Base DN:

   - In the Base DN field, type the desired Base DN.
   - Click **Select** to use the Tree view or the Search view to select entries.

7. Click the **Add** icon to specify the DNs of the group whose members are eligible to use this virtual attribute.

   Do one of the following to specify the DNs of the group:

   - In the Group DN field, type the desired Group DN.
   - Click **Select** to use the Tree view or the Search view to select entries.

8. Click the **Add** icon to specify the search filters to apply against these entries to determine if virtual attribute needs to be generated for those entries.

9. Click **Create.**

---

**Note:** The properties described in the preceding procedure are the same for all virtual attributes except for User Defined and Member virtual attributes, because they contain some additional properties.

**User Defined Virtual Attributes**

It contains the following additional properties:

`Conflict Behavior`: Specifies the behavior that the server has to exhibit for entries that already contain one or more real values for the associated attribute. It has the following values:

- `Merge real and virtual`: Indicates that the virtual attribute provider is to preserve any real values contained in the entry and merge them with the set of generated virtual values so that both the real and virtual values are used.
- `Real overrides virtual`: Indicates that any real values contained in the entry are preserved and used, and virtual values are not generated.
- `Virtual overrides real`: Indicates that the virtual attribute provider suppresses any real values contained in the entry, and generates virtual values and uses them.

`Value`: Specifies the values to be included in the virtual attribute.

**Member Virtual Attributes**

It contains the following additional properties:

`Conflict Behavior`: It is similar to the User Defined Virtual Attributes.

`Allow Retrieving Membership`: Indicates whether to handle requests that demands all values for the virtual attribute. The default value is `false`.

---

## 17.15 Managing Data With Oracle Directory Services Manager

The Data Browser tab of each server instance in ODSM enables you to perform a basic search on the directory data, and to add, delete, and modify entries.

ODSM includes an "auto-suggest" facility that enables you to enter a subset of characters in any of the data fields. ODSM then returns all entries that match that subset of characters. The auto-suggest feature returns only those entries that have already been cached by ODSM.

The following sections describe how to manage data with ODSM, and contains the topics:

- Section 17.15.1, "Display Entries"
- Section 17.15.2, "View the Attributes of an Entry"
- Section 17.15.3, "Search for Entries"
- Section 17.15.4, "Add an Entry"
- Section 17.15.5, "Add an Entry Based on an Existing Entry"
- Section 17.15.6, "Delete an Entry"
- Section 17.15.7, "Delete an Entry and its Subtree"
- Section 17.15.8, "Modify an Entry's RDN"
- Section 17.15.9, "Import Data From an LDIF File"
- Section 17.15.10, "Export Data to an LDIF File"

### 17.15.1 Display Entries

To display directory entries by using the ODSM data browser, complete the following steps:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager."
2. Select the **Data Browser** tab.
3. Select the appropriate network group from the **Network Group** list.
4. Expand the entries in the **Entry** pane to display all of the entries in the required subtree.

   A maximum of 200 entries is displayed at a time.
5. To restrict the entries to a specific entry set, select the subtree (for example, `ou=People`) and click the **Filter** icon.

   In the **Filter** field, type the required filter (for example, `surname=a*`) and click **OK**.
6. Select the entry that you want to view in the left hand pane.

   The entry details are displayed in the tabs on the right.

See also Section 17.15.2, "View the Attributes of an Entry."

### 17.15.2 View the Attributes of an Entry

To view the attributes of an entry:

1. Display the entry as described in Section 17.15.1, "Display Entries."

2. Select the entry that you want to view in the left hand pane.

   The entry details are displayed in the tabs on the right.

   Every entry has a corresponding Properties tab, that displays all the possible attributes of the entry (mandatory and optional). In addition, the following types of entries have a customized tab that displays the mandatory attributes of the entry in a layout that is logical for the entry type:

   - `inetorgperson` entries have a corresponding User Page tab.
   - `group` entries have a corresponding Group Page tab.
   - `country` entries have a corresponding Country Page tab.
   - `domain` entries have a corresponding Domain Page tab.
   - `organization` entries have a corresponding Organization Page tab.
   - `organization unit` entries have a corresponding Organization Unit Page tab.

### 17.15.3 Search for Entries

The basic search function on the Data Browser tab enables you to search for user or group entries. To perform a basic search on the directory data, complete the following steps:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager."
2. Select the **Data Browser** tab.
3. Select the appropriate network group from the **Network Group** list.
4. Select the **Search** tab on the left hand pane.
5. From the **For** list, select whether you are searching for a user entry or a group entry.
6. Enter any part of the entry name and click the right arrow button. For example, to search for user John Smith, you might enter `Smith`, or `Smi`, or `John`, and so forth.
7. When the entry is displayed in the left pane, double-click on the entry to display its details in the right pane.

### 17.15.4 Add an Entry

To add or delete entries with Oracle Directory Services Manager, you must have write access to the parent entry and you must know the DN to use for the new entry. To add an entry by using the ODSM data browser, complete the following steps:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager."
2. Select the **Data Browser** tab.
3. Select the appropriate network group from the **Network Group** list.
4. Click the **Add Entry** icon and select the kind of entry that you want to add, for example User Entry.
5. Enter the DN of the parent entry. This is the entry beneath which the new entry will appear in the directory tree, for example, `ou=people,dc=example,dc=com`.

To select an existing entry as the parent entry, click **Select**.

In the **Entry Picker** window, select **Tree View** to navigate the directory tree and locate the entry, or **Search View** to search for the entry.

6. Enter any additional information for the new entry.
7. When the required details have been entered, click **Create**.

### 17.15.5 Add an Entry Based on an Existing Entry

To add an entry that is based on an existing entry by using the ODSM data browser, complete the following steps:

1. Display the existing entries as described in Section 17.15.1, "Display Entries."
2. Select the entry on which you want to base the new entry and click the **Create like entry** icon.

   The details of the existing entry are displayed in the right pane.
3. Provide a new **Common Name** and **User Name** for the entry.
4. Modify any other details of the entry.
5. Click **Create**.

### 17.15.6 Delete an Entry

To delete an entry by using the ODSM data browser, complete the following steps:

1. Display the existing entries as described in Section 17.15.1, "Display Entries."
2. Select the entry that you want to delete and click the **Delete** icon.
3. On the Delete Entry dialog, verify that you are deleting the correct entry and click **OK**.

### 17.15.7 Delete an Entry and its Subtree

To delete an entry and all entries beneath it in the directory tree, complete the following steps:

1. Display the existing entries as described in Section 17.15.1, "Display Entries."
2. Select the entry that you want to delete and click the **Delete Entry and its Subtree** icon.
3. On the Delete Subtree dialog, verify that you are deleting the correct entry and click **OK**.

### 17.15.8 Modify an Entry's RDN

To modify the RDN of an entry by using the ODSM data browser, complete the following steps:

1. Display the existing entries as described in Section 17.15.1, "Display Entries."
2. Select the entry whose RDN you want to modify on which you want to base the new entry and click the **Edit RDN** icon.
3. Provide a new RDN in the **New RDN value** field.

4. Select **Delete Old RDN** if you want the values that formed the old RDN to be deleted from the entry. If you do not select this checkbox, the values that formed the old RDN are retained as non-distinguished attribute values of the entry.
5. Optionally, click the **Refresh subtree entries** icon to verify the RDN change.

## 17.15.9 Import Data From an LDIF File

You can import entries from an LDIF file, as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager."
2. Select the **Data Browser** tab.
3. Select the appropriate network group from the **Network Group** list.
4. Click the **Import LDIF** icon.
5. On the **Import Entry(ies)** dialog, click **Choose File**.
6. Locate the LDIF file on your system and click **OK**.
7. On the **LDIF Import Progress** dialog, monitor the progress of the import and click **OK** when the export has completed.
8. The Data Browser tree refreshes to show the new entries.

## 17.15.10 Export Data to an LDIF File

You can export entries to an LDIF file, by using ODSM, as follows:

To export entries to an LDIF file, by using the ODSM data browser, complete the following steps:

1. Display the entries as described in Section 17.15.1, "Display Entries."
2. Navigate to the top level DN of the subtree you want to export and click the **Export LDIF** icon.
3. On the **Export Entry** dialog, select **Export Operational Attributes** if you want the operational attributes to be exported.
4. Click **OK**.
5. **Click here to open the LDIF file**.

   The complete LDIF file is displayed in a separate tab of the browser window in which ODSM is running.
6. Save the LDIF file to a writable location.
7. Click **OK** on the Export Entry dialog to exit the export.

# 18

# Accessing Oracle Unified Directory by Using Oracle Directory Services Manager

Oracle Directory Services Manager (ODSM) is an interface for managing instances of Oracle Unified Directory. ODSM enables you to configure the structure of the directory, define objects in the directory, add and configure users, groups, and other entries. ODSM is also the interface you use to manage entries, schema, security, and other directory features.

This section covers the following topics:

- Section 18.1, "Invoking Oracle Directory Services Manager"
- Section 18.2, "Connecting to the Server From Oracle Directory Services Manager"
- Section 18.3, "Displaying Server Information With Oracle Directory Services Manager"

Additional information about using ODSM to manage Oracle Unified Directory is available in the following sections:

- Section 14.2, "Managing the Server Configuration With Oracle Directory Services Manager"
- Section 15.2, "Managing the Proxy Configuration With ODSM"
- Section 17.4.5, "Searching Data With Oracle Directory Services Manager"
- Section 17.15, "Managing Data With Oracle Directory Services Manager"
- Section 27.6, "Managing the Schema With Oracle Directory Services Manager"
- Section 22.3, "Managing Access Control With Oracle Directory Services Manager"
- Section 24.5, "Configuring Password Policies by Using Oracle Directory Services Manager"

## 18.1 Invoking Oracle Directory Services Manager

For information about supported browsers for ODSM, refer to System Requirements and Supported Platforms for Oracle Fusion Middleware 11*g*R2, which is linked from: http://www.oracle.com/technetwork/middleware/ias/downloads/fusion-certification-100350.html

To invoke ODSM, enter the following URL into your browser's address field: `http://host:port/odsm` where *host* is the name of the managed server on which ODSM is running and *port* is the managed server port number of the admin server. The default admin port is 7001.

Connect to the server as described in the following section.

## 18.2 Connecting to the Server From Oracle Directory Services Manager

The following image shows a portion of the main ODSM screen, from which you can connect to a specific Oracle Unified Directory instance.

***Figure 18–1 ODSM Screen***


Enter the following information to connect to an Oracle Unified Directory instance:

- **Server.** Enter the name of the directory server to which you want to connect.
- **Port.** Enter the administration port number of the directory server to which you want to connect.
- **User name.** Enter the bind DN to connect to the directory.
- **Password.** Enter the bind password to connect to the directory.

If SSL is enabled, you are asked to trust the server certificate.

---

**Note:** If you change the browser language setting, you must update the session in order to use the new setting. To update the session, either reenter the ODSM URL in the URL field and press Enter or quit and restart the browser.

---

## 18.3 Displaying Server Information With Oracle Directory Services Manager

The Home tab of each server instance in ODSM enables you to view specific information about the server.

This section describes how to view server information and contains the following topics:

### 18.3.1 View Version Information

The version information panel indicates the version number of the ODSM instance, the Oracle Unified Directory instance, and the version of the Java Runtime Edition (JRE).

### 18.3.2 View the Server Role

The server role can be one or more of the following, depending on how the Oracle Unified Directory instance was set up.

- Directory
- Proxy
- Load Balancer
- Distributor
- Replication Gateway
- Replication Server

For more information, see "Selecting a Server Role" in *Oracle Fusion Middleware Installation Guide for Oracle Unified Directory*.

### 18.3.3 View Server Statistics

The OUD Statistics panel displays installation details and basic monitoring information for this server instances. The following information is displayed:

- **Server Start Time.** The latest date and time on which the server was started successfully.
- **Installation Path.** The network path to the installation files for this server instance.
- **Instance Path.** The network path to the instance files for this server instance.
- **Administrative User.** The root user that was configured when the server was set up. For more information, see Chapter 23, "Managing Administrative Users."
- **Total LDAP Operations Completed (per sec) (since startup).** The total number of LDAP operations performed on the server, divided by the number of seconds that have passed since server startup.
- **Average Elapsed Time per Operation (since startup) (ms).** The average time taken to complete an LDAP operation.
- **Connection Rate (con/sec).** The number of connections that the server is currently handling per second.

### 18.3.4 View the Configured Connection Handlers

The Connection Handlers panel details of all the connection handlers that are configured for this server instance, including the type of connection handler, the port on which that connection handler is listening, and whether the connection handler is enabled.

For more information about connection handlers, see Section 14.1.5, "Configuring Connection Handlers With `dsconfig`" and Section 14.3, "Managing Administration Traffic to the Server."

### 18.3.5 View the Configured Naming Contexts

The Naming Contexts panel displays all naming contexts, or suffixes, that are configured on this server instance, including the network group to which that naming context belongs, the number of entries in the naming context and whether or not that naming context is replicated.

### 18.3.6 View the Configured Data Sources

For proxy servers, the Data Sources panel displays all of the data sources, or back-end LDAP servers that are managed through that proxy instance.

# 19 Managing Users and Groups

Oracle Unified Directory provides a comprehensive user management model that includes identity mapping, and account status notification. This section describes how to configure these elements by using the command-line utilities and by using the Oracle Directory Services Manager interface.

The chapter covers the following topics:

- Section 19.1, "Managing User Accounts"
- Section 19.2, "Configuring Root Users"
- Section 19.3, "Defining Groups"
- Section 19.4, "Maintaining Referential Integrity"
- Section 19.5, "Simulating ODSEE Roles in an Oracle Unified Directory Server"

For information about user passwords, see Chapter 24, "Managing Password Policies."

## 19.1 Managing User Accounts

User accounts are essentially user entries that you create, modify, or remove in your directory.

Before you begin to manage user accounts, ensure that you have the appropriate password policies set up on the directory server. For more information, see Chapter 24, "Managing Password Policies."

This section describes how to manage user accounts and passwords by using the `manage-account` and `ldappasswordmodify` command-line utilities. The section covers the following topics:

- Section 19.1.1, "Changing Passwords"
- Section 19.1.2, "Managing a User's Account Information"
- Section 19.1.3, "Setting Resource Limits on a User Account"

### 19.1.1 Changing Passwords

Directory administrators are often asked to create, reset, or remove passwords for other users. The `ldappasswordmodify` utility enables you to change or reset a user's password with the LDAP password modify extended operation. You can specify authorization IDs with the `--authzid` option by prefixing `dn:`, `u:`, or by specifying the full DN.

This section describes how to manage passwords, and contains the following topics:

#### 19.1.1.1 To Change the Directory Manager's Password

Use the `ldappasswordmodify` command, as shown in the following example:

```
$ ldappasswordmodify -h localhost -p 1389 \
  --authzID "dn:cn=Directory Manager" \
  --currentPassword mypassword --newPassword mynewpassword

The LDAP password modify operation was successful
```

#### 19.1.1.2 To Reset and Generate a New Password for a User

This example assumes that the user does not remember the existing password.

Use the `ldappasswordmodify` command, as shown in the following example:

```
$ ldappasswordmodify -h localhost -p 1389 -D "cn=Directory Manager" \
  -j pwd-file --authzID u:jvedder

The LDAP password modify operation was successful
Generated Password:  evx07npv
```

#### 19.1.1.3 To Change a User's Password

This example assumes that the user remembers the existing password. The new password is passed to the server in a specified file.

Use the `ldappasswordmodify` command, as shown in the following example:

```
$ ldappasswordmodify -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
  --authzID uid=jvedder,ou=People,dc=example,dc=com \
  --currentPassword password --newPasswordFile pwdFile

The LDAP password modify operation was successful
```

### 19.1.2 Managing a User's Account Information

You can use the `manage-account` command to display information about the user's account and any password policy that is applied to the user. You can also use this command to enable or disable a user's account. The `manage-account` command accesses the server over SSL via the administration port. For more information, see Section 14.3, "Managing Administration Traffic to the Server."

This section describes how to manage a user's account information, and covers the following topics:

#### 19.1.2.1 To View a User's Account Information

The `manage-account` command returns the DN of the password policy in effect on a user account, as well as the account status, and password and login related information

1. To display all available information on a user account, use the `manage-account` command with the `get-all` subcommand, as shown in the following example:

```
$ manage-account -D "cn=directory manager" -j pwd-file get-all \
  --targetDN uid=kvaughan,ou=People,dc=example,dc=com
Password Policy DN:  cn=Default Password Policy,cn=Password Policies,cn=config
Account Is Disabled:  false
Account Expiration Time:
Seconds Until Account Expiration:
Password Changed Time:  19700101000000.000Z
Password Expiration Warned Time:
Seconds Until Password Expiration:  432000
Seconds Until Password Expiration Warning:  0
Authentication Failure Times:
Seconds Until Authentication Failure Unlock:
Remaining Authentication Failure Count:
Last Login Time:
Seconds Until Idle Account Lockout:
Password Is Reset:  false
Seconds Until Password Reset Lockout:
Grace Login Use Times:
Remaining Grace Login Count:  4
Password Changed by Required Time:
Seconds Until Required Change Time:
Password History:
```

2. To display just a single property of the account, substitute the `get-all` subcommand with the subcommand corresponding to the property you want to view.

   For example, to view just the password history, run the following command:

```
$ manage-account -D "cn=directory manager" -j pwd-file get-password-history \
  --targetDN "uid=kvaughan,ou=People,dc=example,dc=com"
```

   For a complete list of subcommands, run the following command:

```
$ manage-account --help
```

#### 19.1.2.2 To View Account Status Information

You can use the `manage-account` command to assess whether an account is enabled or disabled.

Use the `manage-account` command with the `get-account-is-disabled` subcommand, as shown in the following example:

```
$ manage-account -D "cn=directory manager" -j pwd-file get-account-is-disabled \
  --targetDN "uid=kvaughan,ou=People,dc=example,dc=com"
Account Is Disabled:  false
```

#### 19.1.2.3 To Disable an Account

Use the `manage-account` command with the `set-account-is-disabled` subcommand, as shown in the following example:

```
$ manage-account -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X \
```

```
  set-account-is-disabled --operationValue true \
  --targetDN "uid=kvaughan,ou=People,dc=example,dc=com"
Account Is Disabled:  true
```

#### 19.1.2.4 To Enable an Account

Use the `manage-account` command with the `clear-account-is-disabled` subcommand, as shown in the following example:

```
$ manage-account -D "cn=directory manager" -j pwd-file clear-account-is-disabled \
  --targetDN "uid=kvaughan,ou=People,dc=example,dc=com"
Account Is Disabled:  false
```

#### 19.1.2.5 To Enable an Account Using `orclIsEnabled`

To enable Oracle Unified Directory using `orclIsEnabled`, complete the following steps:

1. Create and enable a new workflow element as follows:

   ```
   $ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n /
     create-workflow-element --element-name fawe --type fa \
     --set enabled:true --set next-workflow-element:userRoot
   ```

2. Assign the new workflow element to the default workflow, as follows:

   ```
   $ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n /
     set-workflow-prop --workflow-name userRoot0 --set workflow-element:fawe
   ```

### 19.1.3 Setting Resource Limits on a User Account

You can control search operations on the server for each client account by assigning resource limits to the entry. Resource limits are assigned by adding specific operational attributes to the user entry. The directory server then enforces the limits based on the account that the client uses to bind to the directory.

The resource limits that you set on specific user accounts take precedence over the resource limits set in the server-wide configuration. For details of all the configurable resource limit properties, see "Global Configuration" in the *Oracle Unified Directory Configuration Reference*.

The following limits can be set:

- **Look-through limit**. Specifies the maximum number of entries examined for a search operation. Use the `ds-rlim-lookthrough-limit` operational attribute.
- **Size limit**. Specifies the maximum number of entries returned in response to a search operation. use the `ds-rlim-size-limit` operational attribute.
- **Time limit**. Specifies the maximum time spent processing a search operation. Use the `ds-rlim-time-limit` operational attribute.

> **Note:** The Directory Manager can use unlimited resources by default.

#### 19.1.3.1 To Set Resource Limits on an Account

1. Modify the entry in an LDIF file, adding the operational attributes, as shown here:

   ```
   dn: uid=kvaughan,ou=people,dc=example,dc=com
   changetype: modify
   add: ds-rlim-lookthrough-limit
   ```

```
ds-rlim-lookthrough-limit: 1000
-
add: ds-rlim-size-limit
ds-rlim-size-limit: 500
-
add: ds-rlim-time-limit
ds-rlim-time-limit: 300
```

2. Use the `ldapmodify` command to apply the changes, as shown here:

```
$ ldapmodify -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
  --filename add_resource.ldif
Processing MODIFY request for uid=kvaughan,ou=people,dc=example,dc=com
MODIFY operation successful for DN uid=kvaughan,ou=people,dc=example,dc=com
```

## 19.2 Configuring Root Users

A root user is a special user whose account can bypass access controls and other restrictions that might be enforced for regular users. You can define multiple root users, each with their own set of credentials, to control access at a fine-grained level. For example, you can assign privileges to a user who might need root access for a particular task but might not need the full set of root user privileges. You can configure each root user to have his own strong authentication mechanism (such as GSSAPI SASL), his own specific password policy, and his own resource limits.

A set of global root user privileges is defined by default. These privileges apply to all configured root users, including the default root user, unless you modify the privilege in the root user entry. You can change the global root user privileges that are inherited by all root users.

During the setup process, a default root user with full administrative rights is created. The DN proposed by the setup for this root user is `"cn=directory manager"`, so if you do not change the defaults proposed by the setup, a root user with the DN `"cn=directory manager,cn=Root DNs,cn=config"` is configured.

You can manage root users and their privileges, by using the procedures outlined in the following sections.

### 19.2.1 Configuring Root Users by Using the Command-Line Utilities

You can view and edit the global root user properties by using the `dsconfig` command. To create and manage additional root users, you must use the `ldapmodify` command to add the user entries to the server configuration. The following sections describe how to manage root users by using the command line.

- Section 19.2.1.1, "To Change the Global Root User Privileges"
- Section 19.2.1.2, "To Create a New Root User"
- Section 19.2.1.3, "To Edit an Existing Root User"

#### 19.2.1.1 To Change the Global Root User Privileges

To display the global root user privileges, run the following `dsconfig` command:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  get-root-dn-prop
Property                    : Value(s)
----------------------------:-------------------------------------------------
default-root-privilege-name : backend-backup, backend-restore, bypass-acl,
                            : bypass-lockdown, cancel-request, config-read,
```

```
                                : config-write, disconnect-client, ldif-export,
                                : ldif-import, modify-acl, password-reset,
                                : privilege-change, server-restart,
                                : server-shutdown, subentry-write,
                                : unindexed-search, update-schema
```

To change the global root user privileges, run the following dsconfig command run the `dsconfig set-root-dn-prop` command with the `--add` or `--remove` option.

The following example removes the default privilege of root users to perform a backup or restore operation on the server.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  set-root-dn-prop --remove default-root-privilege-name:backend-backup \
  --remove default-root-privilege-name:backend-restore
```

For a complete list of the privileges and an explanation of each privilege, see Section 10.2, "Privilege Subsystem".

#### 19.2.1.2 To Create a New Root User

To create a new root user, create the user entry in an LDIF file, then use the `ldapmodify` command to add the entry to the `cn=Root DNs,cn=config` branch in the server configuration.

Note that the `cn=config` suffix is an administrative suffix and, as such, must be accessed using the administration connector. For more information see Section 14.3, "Managing Administration Traffic to the Server".

Suppose, for example, that you want to give a particular user the right to backup and restore a database, but no other administrative privileges.

1. Create an LDIF file that defines the root user entry with the correct privileges.

   The following sample LDIF file (`add-backup-admin.ldif`) defines a root user with the DN `"cn=backup-admin"` who has these privileges, but no other privileges on the server configuration.

   ```
   dn: cn=backup-admin,cn=Root DNs,cn=config
   changetype: add
   objectClass: person
   objectClass: inetOrgPerson
   objectClass: organizationalPerson
   objectClass: ds-cfg-root-dn-user
   objectClass: top
   cn: backup-admin
   sn: backup-admin
   ds-cfg-alternate-bind-dn: cn=backup-admin
   userPassword: secret
   ds-privilege-name: backend-backup
   ds-privilege-name: backend-restore
   ds-privilege-name: -bypass-acl
   ds-privilege-name: -bypass-lockdown
   ds-privilege-name: -cancel-request
   ds-privilege-name: -config-read
   ds-privilege-name: -config-write
   ds-privilege-name: -disconnect-client
   ds-privilege-name: -ldif-export
   ds-privilege-name: -ldif-import
   ds-privilege-name: -modify-acl
   ds-privilege-name: -password-reset
   ds-privilege-name: -privilege-change
   ```

```
ds-privilege-name: -server-restart
ds-privilege-name: -server-shutdown
ds-privilege-name: -subentry-write
ds-privilege-name: -unindexed-search
ds-privilege-name: -update-schema
```

2. Use the ldapmodify command with the --useSSL option to add the LDIF file to the server configuration.

```
$ ldapmodify -h localhost -p 4444 -D "cn=directory manager" -j pwd-file \
  --useSSL -X -f add-backup-admin.ldif
```

For a complete list of the privileges and an explanation of each privilege, see Section 10.2, "Privilege Subsystem".

#### 19.2.1.3 To Edit an Existing Root User

To edit an existing root user, use the `ldapmodify` command to change the attributes of the user entry under the `cn=Root DNs,cn=config` branch in the server configuration.

Note that the `cn=config` suffix is an administrative suffix and, as such, must be accessed using the administration connector. For more information see Section 14.3, "Managing Administration Traffic to the Server".

The following example removes the capability of the root user created in the previous example to perform a restore operation.

```
$ ldapmodify -h localhost -p 4444 -D "cn=directory manager" -j pwd-file \
  --useSSL -X
dn: cn=backup-admin,cn=root DNs,cn=config
changetype: modify
delete: ds-privilege-name
ds-privilege-name: backend-restore
```

### 19.2.2 Configuring Root Users by Using ODSM

You can view and edit the default root user, and create and manage additional root users, by using the ODSM interface. This section covers the following topics:

- Section 19.2.2.1, "Configure the Global Root User Privileges"
- Section 19.2.2.2, "Create a New Root User"
- Section 19.2.2.3, "Edit an Existing Root User"

#### 19.2.2.1 Configure the Global Root User Privileges

A set of global root user privileges is defined by default. These privileges apply to all configured root users, unless you modify the privilege in the root user entry.

To modify the global root user privileges by using ODSM, follow these steps:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager."
2. Select the **Configuration** tab.
3. Under the **General Configuration** item, select **Root Users**.

   The global root user privileges are displayed in the right hand pane.

   A check mark next to a privilege indicates that root users have that privilege by default.

4. To add a privilege to the list of global root user privileges, check the box next to that privilege.

   To remove a privilege, uncheck the box next to that privilege.

   For a complete list of the privileges and an explanation of each privilege, see Section 10.2, "Privilege Subsystem".

5. When you have made the modifications that you require, click **Apply**.

#### 19.2.2.2 Create a New Root User

You can create a new root user by using ODSM, as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager."
2. Select the **Configuration** tab.
3. From the **Create** menu, select **Root User.**
4. In the **General Properties** region, enter the following details:
   a. In the **Name** field, type a name for the root user that you want to create.
   b. In the **Alternative Bind DNs** region, click **Add** to specify one or more alternative DNs that can be used when this root users binds to the server.

      For example, the alternative bind DN for the default root user is `"cn=Directory Manager"`. This allows you to bind as `"cn=Directory Manager"` instead of having to use `"cn=Directory Manager,cn=Root DNs,cn=config"`, which is the actual entry DN.

      The alternative bind DN must be unique among all root users.

      If you do not want to specify an alternative bind DN for the new root user, leave the table empty.

   c. In the **Password** field, enter a password for the root user.
   d. In the **Confirm Password** field, retype the password for the root user.
5. In the **Privileges** region, select the settings for the different privileges that must be applied to this new root user.

   For each privilege, you can select one of the following:

   - `Enable`. The privilege is enabled for this root user.
   - `Disable`. The privilege is disabled for this root user.
   - `Default Privilege (enable)` or `Default Privilege (disable`. The user inherits the default setting for this privilege, as defined in the global privilege configuration
6. Click **Create.**

   The following confirmation message appears:

   `Root User created successfully.`

#### 19.2.2.3 Edit an Existing Root User

You can edit an existing root user by using ODSM, as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager."

2. Select the **Configuration** tab.
3. Under the **General Configuration** item, expand the **Root Users** item.
4. Select the root user whose configuration you want to change.

   The properties of the root user are displayed in the right hand pane.
5. Edit the required properties and click **Apply**.
6. You are prompted to save the new configuration. Click **Yes**.

## 19.3 Defining Groups

Oracle Unified Directory supports *groups*, which are collections of entries that are manageable as a single object. Typically, directory administrators configure groups of printers, groups of software applications, groups of employees, and so forth. Groups are especially useful when assigning special access privileges to a set of users. For example, you can assign access managers the privileges to employee data while restricting those same privileges to others in the company.

The following group types are supported:

- **Static groups**. A static group defines its membership by providing explicit sets of distinguished names (DNs) using the `groupOfNames`, `groupOfUniqueNames`, or `groupOfEntries` object class. Statics groups are well supported by external clients and provide good performance.

  A disadvantage of static groups is that as the group membership increases, the ability to easily manage the data becomes more difficult. For every entry that changes, all groups containing the changed entry must also be changed. This task becomes more difficult as the number of members of a group grows large. As a result, static groups are best used for relatively small groups that change infrequently.

  For more info, see Section 19.3.1, "Defining Static Groups."

- **Dynamic groups**. A dynamic group defines its membership using a set of search criteria in the form of an LDAP URL, using the `groupOfUrls` object class. Compared to static groups, dynamic groups handle large numbers of members well (millions of entries). As entries are updated, all parent groups are updated automatically.

  A disadvantage of dynamic groups is that not all clients support them. Performance also is adversely affected if you need to query the whole list of entries. Thus, dynamic groups are best suited for groups with a very large number of entries or for clients that need to determine specific group membership for an entry.

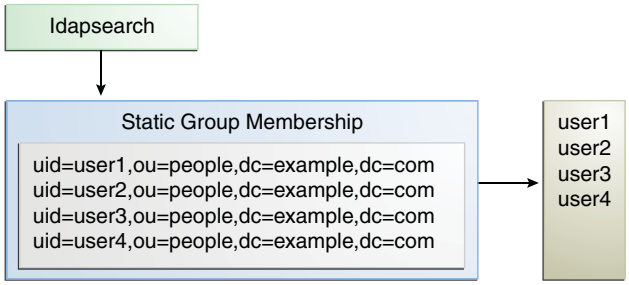
  For more info, see Section 19.3.2, "Defining Dynamic Groups."

- **Virtual static groups**. A virtual static group appears and behaves like a static group to external clients, except that each member is represented by a virtual attribute that defines its membership on the fly from another dynamic group. Virtual static groups provide an efficient way to manage large numbers of entries and avoid the scalability issues for clients that only support static groups.
- For more info, see Section 19.3.3, "Defining Virtual Static Groups."

### 19.3.1 Defining Static Groups

A static group is one whose entry contains a membership list of explicit DNs. Many clients support static groups, but static groups are difficult to manage as the number of members in a group increases in size. For example, if you have a member entry that requires a DN change, then you must change the user's DN for each group she belongs to.

Because a static group contains a list of explicit member DNs, its database footprint increases as the membership list grows. For this reason, a static group is best suited for small groups (less than 10,000) whose entries do not change frequently. Using large static groups can have a detrimental impact on performance. If you know that group membership will exceed 10,000, consider using dynamic groups instead.




The directory server supports the following three types of static groups, divided according to the object class they use:

- `groupOfNames` You can define a static group by using the `groupOfNames` object class and by explicitly specifying the member DNs using the `member` attribute.

  **Note:** RFC 4519 (`https://opends.dev.java.net/public/standards/rfc4519.txt`) requires that the `member` attribute be mandatory within the `groupOfNames` object class. This membership requirement has traditionally caused data management problems when an administrator attempted to delete the last member in the group. The directory server solves this problem by allowing the `member` attribute to be optional. The optional membership requirement allows you to have an empty object class when you delete the last member of the group.

```
dn: cn=Example Static Group 1,ou=Groups,dc=example,dc=com
objectClass: top
objectClass: groupOfNames
member: uid=user1,ou=People,dc=example,dc=com
member: uid=user2,ou=People,dc=example,dc=com
cn: Example Static Group 1
```

- `groupOfUniqueNames` You can define a static group by using the `groupOfUniqueNames` object class and by explicitly specifying the member DNs using the `uniqueMember` attribute. The `groupOfUniqueNames` object class differs from the `groupOfNames` object class in that you can enumerate the group's members by specifying a unique DN plus an optional identifier. The identifier ensures that the unique objects can be identified when adding, deleting, or renaming any object.

For example, you could delete or move an employee (cn=Tom Smith) and add a new employee who has the same name (cn=Tom Smith) to the directory. To distinguish the two, you must add a separate identifier by using a bit string. The following example shows two users with the same name, but the second uniqueMember has an optional identifier.

```
uniqueMember: uid=tsmith,ou=People,dc=example,dc=com
uniqueMember: uid=tsmith,ou=People,dc=example,dc=com#'0111101'B
```

**Note:** Few LDAP applications actually use the optional UID identifier.

RFC 4519 (https://opends.dev.java.net/public/standards/rfc4519.txt) requires that the uniqueMember attribute be mandatory within the groupOfUniqueNames object class. This membership requirement has historically caused data management problems when an administrator tried to delete the last member in the group. Oracle Unified Directory solves this problem by allowing the uniqueMember attribute to be optional. The optional membership requirement allows you to have an empty object class when you delete the last member of the group.

```
dn: cn=Example Static Group 2,ou=Groups,dc=example,dc=com
objectClass: top
objectClass: groupOfUniqueNames
uniqueMember: uid=user1,ou=People,dc=example,dc=com
uniqueMember: uid=user2,ou=People,dc=example,dc=com
cn: Example Static Group 2
```

- groupOfEntries You can define a static group using the groupOfEntries object class. Based on the original specifications (RFC 4519 (http://www.rfc-editor.org/rfc/rfc4519.txt) and draft-findlay-ldap-groupofentries-00.txt (http://ietfreport.isoc.org/idref/draft-findlay-ldap-groupofentries/), which expired in March, 2008), the groupOfEntries object class differs from the groupOfNames and groupOfUniqueNames object classes in that attributes are optional. This allows you to specify an empty object class without any members.

  **Note:** Oracle Unified Directory supports the groupOfEntries draft but also allows empty groupOfNames and groupOfUniqueNames object classes. As a result, you can create empty groups of any type (groupOfEntries, groupOfNames, and groupOfUniqueNames).

  ```
  dn: cn=Example Static Group 3,ou=Groups,dc=example,dc=com
  objectClass: top
  objectClass: groupOfEntries
  cn: Example Static Group 3
  ```

This section contains the following topics:

- Section 19.3.1.1, "To Create a Static Group With groupOfNames"

#### 19.3.1.1 To Create a Static Group With groupOfNames

1. Create the group entry in LDIF, including the group name (cn) and the groupOfNames object class.

   This example shows an LDIF file, named static-group1.ldif, that defines the new group.

   ```
   dn: cn=Directory Administrators,ou=Groups,dc=example,dc=com
   cn: Directory Administrators
   objectclass: top
   objectclass: groupOfNames
   ou: Groups
   member: uid=ttully,ou=People,dc=example,dc=com
   member: uid=charvey,ou=People,dc=example,dc=com
   member: uid=rfisher,ou=People,dc=example,dc=com
   ```

2. Add the group by using ldapmodify to apply the LDIF file.

   ```
   $ ldapmodify -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
     --defaultAdd --filename static-group1.ldif
   Processing ADD request for cn=Directory
   Administrators,ou=Groups,dc=example,dc=com
   ADD operation successful for DN cn=Directory
   Administrators,ou=Groups,dc=example,dc=com
   ```

#### 19.3.1.2 To Create a Static Group With groupOfUniqueNames

1. Create the group entry in LDIF, including the group name (cn) and the groupOfUniqueNames object class.

   This example shows an LDIF file, named static-group2.ldif, that defines the new group.

   ```
   dn: cn=Directory Administrators2,ou=Groups,dc=example,dc=com
   cn: Directory Administrators2
   objectclass: top
   objectclass: groupOfUniqueNames
   ou: Groups
   uniquemember: uid=alangdon,ou=People,dc=example,dc=com
   uniquemember: uid=drose,ou=People,dc=example,dc=com
   uniquemember: uid=polfield,ou=People,dc=example,dc=com
   ```

2. Add the group by using ldapmodify to apply the LDIF file.

   ```
   $ ldapmodify -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
     --defaultAdd --filename static-group2.ldif
   ```

3. Verify the change by using ldapsearch and the isMemberOf attribute.

   ```
   $ ldapsearch -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
     --baseDN dc=example,dc=com "(uid=rdaugherty)" isMemberOf
   dn: uid=alangdon,ou=People,dc=example,dc=com
   isMemberOf: cn=Directory Administrators2,ou=Groups,dc=example,dc=com
   ```

#### 19.3.1.3 To Create a Static Group With `groupOfEntries`

1. Create the group entry in LDIF, including the group name (`cn`) and the `groupOfEntries` object class.

   This example shows an LDIF file, named `static-group3.ldif`, that defines the new group.

   ```
   dn: cn=Directory Administrators3,ou=Groups,dc=example,dc=com
   cn: Directory Administrators3
   objectclass: top
   objectclass: groupOfEntries
   ou: Groups
   member: uid=bfrancis,ou=People,dc=example,dc=com
   member: uid=tjames,ou=People,dc=example,dc=com
   member: uid=bparker,ou=People,dc=example,dc=com
   ```

2. Add the group by using `ldapmodify` to apply the LDIF file.

   ```
   $ ldapmodify -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
     --defaultAdd --filename static-group3.ldif
   ```

3. Verify the change by using `ldapsearch` and the `isMemberOf` attribute.

   ```
   $ ldapsearch -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
     --baseDN dc=example,dc=com "(uid=bparker)" isMemberOf
   dn: uid=bparker,ou=People,dc=example,dc=com
   isMemberOf: cn=Directory Administrators3,ou=Groups,dc=example,dc=com
   ```

#### 19.3.1.4 To List All Members of a Static Group

You can use the `isMemberOf` virtual attribute to search for a group. The attribute is added to the user entry at the start of the search and then removed after the search has finished. This functionality provides easy management of groups with fast read access.

Use the `ldapsearch` command with the virtual attribute `isMemberOf`.

This example searches for all users who are members of the group "Accounting Managers".

```
$ ldapsearch -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
  -b dc=example,dc=com \
  "(isMemberOf=cn=Accounting Managers,ou=Groups,dc=example,dc=com)"
dn: uid=scarter,ou=People,dc=example,dc=com
objectClass: person
objectClass: inetOrgPerson
objectClass: top
objectClass: organizationalPerson
ou: Accounting
ou: People
sn: Carter
facsimiletelephonenumber: +1 408 555 9751
roomnumber: 4612
userpassword: {SSHA}3KiJ51sx2Ug7DxZoq0vA9ZY6uaomevbJUBm7OA==
l: Sunnyvale
cn: Sam Carter
telephonenumber: +1 408 555 4798
givenname: Sam
uid: scarter
mail: scarter@example.com
dn: uid=tmorris,ou=People,dc=example,dc=com
objectClass: person
objectClass: inetOrgPerson
```

```
objectClass: top
objectClass: organizationalPerson
ou: Accounting
ou: People
sn: Morris
facsimiletelephonenumber: +1 408 555 8473
roomnumber: 4117
userpassword: {SSHA}bjFFHv6k1kbI6fZoCEfqmTj9XOZxWR06gxpKpQ==
l: Santa Clara
cn: Ted Morris
telephonenumber: +1 408 555 9187
givenname: Ted
uid: tmorris
mail: tmorris@example.com
```

#### 19.3.1.5 To List All Static Groups of Which a User Is a Member

Search using `ldapsearch` and the virtual attribute `isMemberOf`, as shown in the following example:

```
$ ldapsearch -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
  -b dc=example,dc=com "(uid=scarter)" isMemberOf
dn: uid=scarter,ou=People,dc=example,dc=com
isMemberOf: cn=Accounting Managers,ou=groups,dc=example,dc=com
```

#### 19.3.1.6 To Determine Whether a User is a Member of a Group

Search using `ldapsearch`, as shown in the following example:

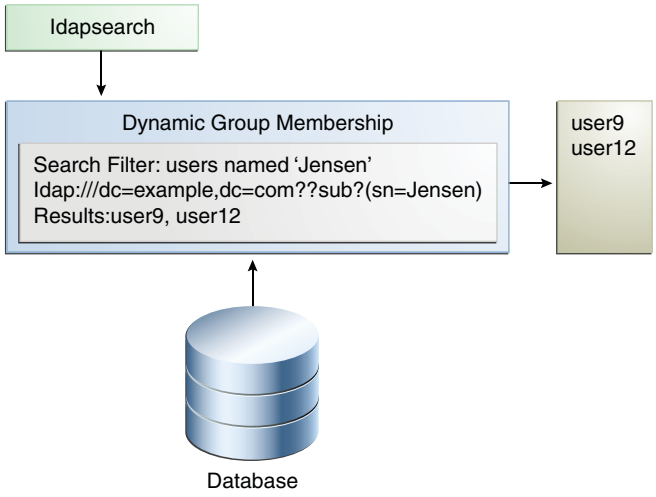
```
$ ldapsearch -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
  -b "cn=Account Managers,ou=Groups,dc=example,dc=com" \
  "(&(objectclass=groupOfUniqueNames) \
  (uniquemember=uid=scarter,ou=People,dc=example,dc=com))"
dn: cn=Accounting Managers,ou=groups,dc=example,dc=com
objectClass: groupOfUniqueNames
objectClass: top
ou: groups
description: People who can manage accounting entries
cn: Accounting Managers
uniquemember: uid=scarter, ou=People, dc=example,dc=com
uniquemember: uid=tmorris, ou=People, dc=example,dc=com
```

### 19.3.2 Defining Dynamic Groups

A *dynamic group* is one whose membership, rather than being maintained explicitly in a list, is determined by search criteria using an LDAP URL. For example, suppose that you want to send an email to all managers in the `dc=example,dc=com` naming context. To do this, you create a dynamic group in which you specify `cn=Managers,ou=Groups,dc=example,dc=com`. You further specify that you want only email addresses returned. When the email application queries the directory for that particular group, the directory server computes the membership dynamically and returns the corresponding list of email addresses.

Dynamic groups use the `groupOfURLs` object class and the `memberURL` attribute to define LDAP URLs with the criteria (search base, scope, and filter) to be used for determining members of the group. The mechanism for determining whether a user is a member of a dynamic group is a constant-time operation, so it is just as efficient for groups with millions of members as it is for a group with only a few members. However, care must be taken when specifying the search criteria as it can adversely affect performance if searching over a large set of data.

***Figure 19–1 Structure of a Dynamic Group***




This section describes the following topics:

- Section 19.3.2.1, "To Create a Dynamic Group"
- Section 19.3.2.2, "To List All Members of a Dynamic Group"
- Section 19.3.2.3, "To List All Dynamic Groups of Which a User Is a Member"
- Section 19.3.2.4, "To Determine Whether a User Is a Member of a Dynamic Group"

### 19.3.2.1 To Create a Dynamic Group

1. Create an LDIF file that specifies the group.

   This example specifies the dynamic group for employees located at Cupertino.

   ```
   dn: cn=cupertinoEmployees,ou=Groups,dc=example,dc=com
   cn: CupertinoEmployees
   objectclass: top
   objectclass: groupOfURLs
   ou: Groups
   memberURL: ldap:///ou=People,dc=example,dc=com??sub?(l=Cupertino)
   ```

2. Add the group by using `ldapmodify` to process the LDIF file.

   ```
   $ ldapmodify -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
     --defaultAdd --filename dynamic_group.ldif
   Processing ADD request for cn=cupertionEmployees,ou=Groups,dc=example,dc=com
   ADD operation successful for DN
   cn=cupertionEmployees,ou=Groups,dc=example,dc=com
   ```

### 19.3.2.2 To List All Members of a Dynamic Group

This procedure illustrates the use of the virtual attribute `isMemberOf`. Do not use this procedure for very large groups, because it adversely affects the directory server's performance.

Search using `ldapsearch` and the virtual attribute `isMemberOf`.

```
$ ldapsearch -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
  -b "dc=example,dc=com" \
"(isMemberOf=cn=cupertinoEmployees,ou=Groups,dc=example,dc=com)"
dn: uid=abergin,ou=People,dc=example,dc=com
```

```
objectClass: person
objectClass: inetOrgPerson
objectClass: top
objectClass: organizationalPerson
ou: Product Testing
ou: People
sn: Bergin
facsimiletelephonenumber: +1 408 555 7472
roomnumber: 3472
userpassword: {SSHA}YcDl0pHLxkd/ouW2jslAk1XaT5SiY4ium5qh8w==
l: Cupertino
cn: Andy Bergin
telephonenumber: +1 408 555 8585
givenname: Andy
uid: abergin
mail: abergin@example.com
...(more entries)...
```

#### 19.3.2.3 To List All Dynamic Groups of Which a User Is a Member

Search using `ldapsearch` and the virtual attribute `isMemberOf`.

```
$ ldapsearch -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
  -b dc=example,dc=com "(uid=abergin)" isMemberOf
dn: uid=abergin,ou=People,dc=example,dc=com
isMemberOf: cn=QA Managers,ou=groups,dc=example,dc=com
isMemberOf: cn=cupertinoEmployees,ou=Groups,dc=example,dc=com
```

#### 19.3.2.4 To Determine Whether a User Is a Member of a Dynamic Group

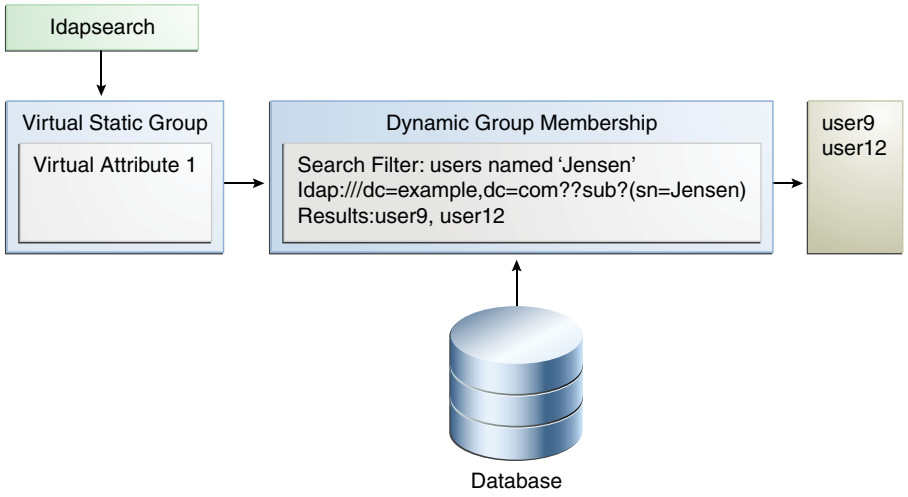
Search using `ldapsearch` and the virtual attribute `isMemberOf`.

```
$ ldapsearch -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
  -b dc=example,dc=com \
"(&(uid=abergin)(isMemberOf=cn=cupertinoEmployees,ou=Groups,dc=example,dc=com))"
dn: uid=abergin,ou=People,dc=example,dc=com
objectClass: person
objectClass: inetOrgPerson
objectClass: top
objectClass: organizationalPerson
ou: Product Testing
ou: People
sn: Bergin
facsimiletelephonenumber: +1 408 555 7472
roomnumber: 3472
userpassword: {SSHA}YcDl0pHLxkd/ouW2jslAk1XaT5SiY4ium5qh8w==
l: Cupertino
cn: Andy Bergin
telephonenumber: +1 408 555 8585
givenname: Andy
uid: abergin
mail: abergin@example.com
```

### 19.3.3 Defining Virtual Static Groups

A *virtual static group*, efficiently manages scalability for clients that can only support static groups. In a virtual static group, each entry behaves like a static group entry by using virtual attributes. The virtual attributes are dynamically determined when invoked, and the operations that determine group membership are passed to another group, such as a dynamic group, as shown in the following diagram.

***Figure 19–2 Virtual Static Group***




Virtual static groups should include either the `groupOfNames` or `groupOfUniqueNames` object class but should not include the `member` or `uniqueMember` attribute. Virtual static groups should also contain the `ds-virtual-static-group` auxiliary object class and the `ds-target-group-dn` attribute. The `ds-target-group-dn` attribute is used to reference the actual group to mirror as a virtual static group and is used in place of the `member` or `uniquemember` attribute. For example:

```
dn: cn=Example Virtual Static Group,ou=Groups,dc=example,dc=com
objectClass: top
objectClass: groupOfUniqueNames
objectClass: ds-virtual-static-group
cn: Example Virtual Static Group
ds-target-group-dn: cn=Example Real Group,ou=Groups,dc=example,dc=com
```

Virtual static groups are most efficient when the application issues a search targeted at the membership attribute but does not actually retrieve the entire set of members. It is common for applications to use a filter such as the following to attempt to determine whether a user is a member of a given group:

```
(&(objectClass=groupOfUniqueNames)(uniqueMember=uid=john.doe,\
  ou=People,dc=example,dc=com))
```

For applications that retrieve the set of members, virtual static groups might not be ideal because the process of constructing the entire member list can be expensive.

This section describes the following topics:

- Section 19.3.3.1, "To Create a Virtual Static Group"
- Section 19.3.3.2, "To List All Members of a Virtual Static Group"
- Section 19.3.3.3, "To List All Virtual Static Groups of Which a User Is a Member"
- Section 19.3.3.4, "To Determine Whether a User is a Member of a Virtual Static Group"

#### 19.3.3.1 To Create a Virtual Static Group

1. Create an LDIF file that specifies the group.

This sample file, `virtual-static.ldif`, specifies a virtual static group named `cupertinoEmployees`.

```
dn: cn=virtualStatic,ou=Groups,dc=example,dc=com
cn: Virtual Static
objectclass: top
objectclass: groupOfUniqueNames
objectclass: ds-virtual-static-group
ou: Groups
ds-target-group-dn: cn=cupertinoEmployees,ou=Groups,dc=example,dc=com
```

2. Add the group by using `ldapmodify` to process the LDIF file.

```
$ ldapmodify -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
  --defaultAdd --filename virtual-static.ldif
Processing ADD request for cn=virtualStatic,ou=Groups,dc=example,dc=com
ADD operation successful for DN cn=virtualStatic,ou=Groups,dc=example,dc=com
```

#### 19.3.3.2 To List All Members of a Virtual Static Group

Virtual static groups are best used in cases where the search is targeted at the membership attribute. This procedure is therefore not recommended but is included to show how to access the list.

This example uses the dynamic group, `cupertinoEmployees` that was created in the previous example.

Search using `ldapsearch` and the virtual attribute `isMemberOf`.

```
$ ldapsearch -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
  -b dc=example,dc=com "(isMemberOf=cn=virtualStatic,ou=Groups,dc=example,dc=com)"
dn: uid=abergin,ou=People,dc=example,dc=com
objectClass: person
objectClass: inetOrgPerson
objectClass: top
objectClass: organizationalPerson
ou: Product Testing
ou: People
sn: Bergin
facsimiletelephonenumber: +1 408 555 7472
roomnumber: 3472
userpassword: {SSHA}YcDl0pHLxkd/ouW2jslAk1XaT5SiY4ium5qh8w==
l: Cupertino
cn: Andy Bergin
telephonenumber: +1 408 555 8585
givenname: Andy
uid: abergin
mail: abergin@example.com
...(more entries)...
```

#### 19.3.3.3 To List All Virtual Static Groups of Which a User Is a Member

Search using `ldapsearch` and the virtual attribute `isMemberOf`.

```
$ ldapsearch -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
  -b dc=example,dc=com "(uid=abergin)" isMemberOf
dn: uid=abergin,ou=People,dc=example,dc=com
isMemberOf: cn=QA Managers,ou=groups,dc=example,dc=com
isMemberOf: cn=cupertinoEmployees,ou=Groups,dc=example,dc=com
isMemberOf: cn=virtualStatic,ou=Groups,dc=example,dc=com
```

#### 19.3.3.4 To Determine Whether a User is a Member of a Virtual Static Group

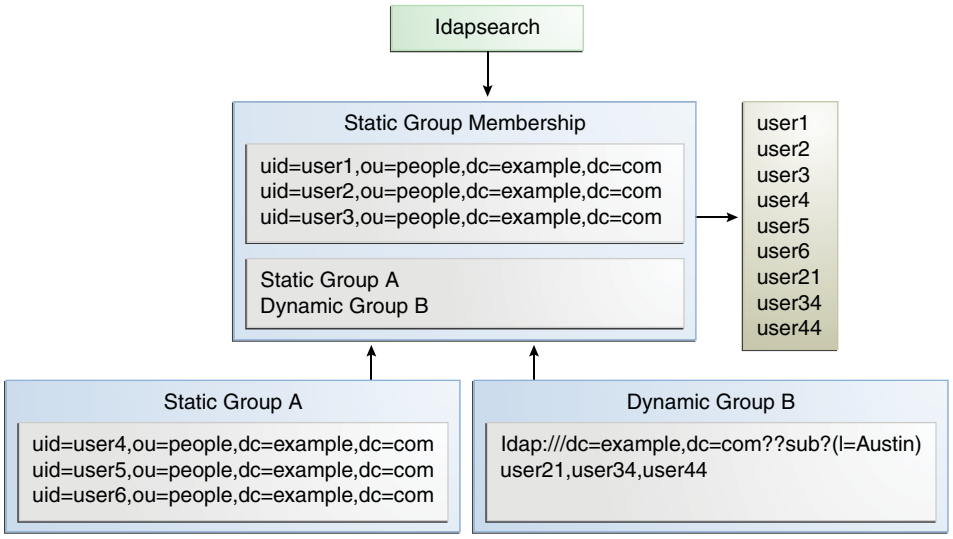
Search using `ldapsearch` and the `uniqueMember` attribute.

```
$ ldapsearch -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
  -b "cn=virtualStatic,ou=Groups,dc=example,dc=com" \
  "(&(objectclass=groupOfUniqueNames) \
  (uniquemember=uid=abergin,ou=People,dc=example,dc=com))"
dn: cn=virtualStatic,ou=Groups,dc=example,dc=com
objectClass: groupOfUniqueNames
objectClass: top
objectClass: ds-virtual-static-group
ou: Groups
ds-target-group-dn: cn=cupertinoEmployees,ou=Groups,dc=example,dc=com
cn: Virtual Static
cn: virtualStatic
```

### 19.3.4 Defining Nested Groups

Groups can be nested, where one group is defined as a child group entry whose DN is listed within another group, its parent. The nesting of groups allows you to set up inherited group memberships when performance is not a priority. You can add zero or more member attributes with their values set to the DNs of nested child groups, including both static and dynamic groups.

***Figure 19–3 Nested Static Group***




#### 19.3.4.1 To Create a Nested Group

This example procedure creates a nested group using one static group and one dynamic group.

1. Create an LDIF file that specifies a static group.

   This example file, `static-group.ldif`, specifies a virtual static group named `Dev Contractors`.

   ```
   dn: cn=Contractors,ou=Groups,dc=example,dc=com
   cn: Dev Contractors
   objectclass: top
   objectclass: groupOfUniqueNames
   ```

```
ou: Dev Contractors Static Group
uniquemember: uid=wsmith,ou=Contractors,dc=example,dc=com
uniquemember: uid=jstearn,ou=Contractors,dc=example,dc=com
uniquemember: uid=pbrook,ou=Contractors,dc=example,dc=com
uniquemember: uid=njohnson,ou=Contractors,dc=example,dc=com
uniquemember: uid=sjones,ou=Contractors,dc=example,dc=com
```

2. Add the group by using `ldapmodify` to process the LDIF file.

```
$ ldapmodify -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
  --defaultAdd --filename static-group.ldif
```

3. Create an LDIF file that specifies a dynamic group.

   This example file, `dynamic-group.ldif`, specifies a dynamic group named `Developers`.

```
dn: cn=Developers,ou=Groups,dc=example,dc=com
cn: Developers
objectclass: top
objectclass: groupOfURLs
ou: Groups
memberURL: ldap:///ou=People,dc=example,dc=com??sub?(ou=Product Development)
```

4. Add the group by using `ldapmodify` to process the LDIF file.

```
$ ldapmodify -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
  --defaultAdd --filename dynamic-group.ldif
```

5. Create an LDIF file that specifies a nested static group.

   This example file, `nested-group.ldif`, specifies a nested group named `Developers Group`.

```
dn: cn=DevelopersGroup,ou=Groups,dc=example,dc=com
cn: Developers Group
objectclass: top
objectclass: groupOfUniqueNames
ou: Nested Static Group
uniquemember: cn=Contractors,ou=Groups,dc=example,dc=com
uniquemember: cn=Developers,ou=Groups,dc=example,dc=com
```

6. Add the group by using `ldapmodify` to process the LDIF file,

```
$ ldapmodify -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
  --defaultAdd --filename nested-group.ldif
```

## 19.4 Maintaining Referential Integrity

Referential integrity is a database mechanism for ensuring that all references are properly maintained after delete, rename, or move operations. For example, if an entry is removed from the directory, the directory server also removes the entry from any groups of which the entry is listed as a member.

The referential integrity mechanism is configured as a plug-in the directory server and can be enabled using the `dsconfig` command. For more information, see Section 14.1, "Managing the Server Configuration With `dsconfig`."

This section describes referential integrity, and contains the following topics:

- Section 19.4.1, "Overview of the Referential Integrity Plug-In"
- Section 19.4.2, "To Enable the Referential Integrity Plug-In"

### 19.4.1 Overview of the Referential Integrity Plug-In

By default, the referential integrity plug-in is disabled. When you enable the plug-in by using `dsconfig`, it performs integrity updates on the `member` and `uniquemember` attributes immediately after a delete, rename, or move operation. Whenever you delete, rename, or move a user or group entry in the directory, the operation is logged to the referential integrity log file, `INSTANCE_DIR/OUD/logs/referint`.

After a specified time, known as the *update interval*, the server performs a search on the specified attributes and matches the results with the DNs of the deleted or modified entries recorded in the log. If the log file shows that an entry was deleted, the corresponding attribute is deleted. If the log file shows that an entry was changed, the corresponding attribute value is modified accordingly.

You can configure the properties of the referential integrity plug-in to suit your requirements. The following properties can be configured:

- **Enabled**. Turn on the referential integrity plug-in.
- **plugin type**. By default, the delete, rename, and move operations are set. You can change a plug-in type to only delete, for example.
- **Attribute type**. By default, the attribute types are set to `member`, `uniquemember` but can be changed to some other attribute. If you use or define attributes containing DN values, you can use the referential integrity plug-in to monitor these attributes.
- **Base-DN**. By default, the scope is to use all public naming contexts but this can be changed to a specific context.
- **Log file**. By default, `logs/referint` is the log file. You can record the referential integrity updates in a different file. For example, if you want to record changes in a replicated environment, you can write to the *changelog* file on a replication server, so that it can be replicated to a consumer server.
- **Update interval**. By default, the update interval is set to 0 seconds, which will run referential integrity immediately after a delete, rename, or move operation. To minimize the impact of the updates on system performance, increase the amount of time between updates. Typical update intervals are as follows:
  - 0 seconds, update immediately
  - 90 seconds (updates every 90 seconds)
  - 3600 seconds (updates every hour)
  - 10,800 seconds (updates every 3 hours)
  - 28,800 seconds (updates every 8 hours)
  - 86,400 seconds (updates once a day)
  - 604,800 seconds (updates once a week)

### 19.4.2 To Enable the Referential Integrity Plug-In

To enable referential integrity by using `dsconfig`, set the `enabled` property of the plug-in to true.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -n \
  set-plugin-prop --plugin-name "Referential Integrity" --set enabled:true
```

## 19.5 Simulating ODSEE Roles in an Oracle Unified Directory Server

Oracle Directory Server Enterprise Edition (ODSEE) includes a roles subsystem that is used to provide a specialized type of grouping mechanism. This capability is not included directly in Oracle Unified Directory, because it is based on non-standard functionality, uses Netscape-proprietary schema elements, and is not widely used in LDAP-enabled applications.

However, Oracle Unified Directory does provide all of the functionality offered by ODSEE roles, and this functionality is available for use with standard grouping mechanisms. If you have an application that was specifically written to rely on the roles functionality available in ODSEE and cannot work with standard grouping mechanisms, you can configure Oracle Unified Directory to simulate ODSEE roles to satisfy such applications.

---

**Note:** If your application needs to create and destroy role entries (for example, an entry containing one of the subordinates of the `nsRoleDefinition` object class), that functionality is currently not available in Oracle Unified Directory.

---

This section contains the following topics:

- Section 19.5.1, "To Determine Whether a User is a Member of a Role"
- Section 19.5.2, "To Alter Membership by Using the `nsRoleDN` Attribute"

### 19.5.1 To Determine Whether a User is a Member of a Role

If the application needs only to determine whether a user is a member of a given role, it should only need to look at the `nsRole` attribute in the target user's entry to determine whether the DN of the appropriate role is present. In this case, you can simulate role functionality by following these steps.

After these steps are completed, the `nsRole` virtual attribute appears as an operational attribute in user entries, and should include the DNs of all groups in which that user is a member. Note that `nsRole` is an operational attribute, and must be explicitly requested for it to be returned in search results. You must also ensure that the authenticated user has permission to see that attribute.

1. Update the directory server to include the necessary schema for the ODSEE roles implementation.

   This schema is provided in the LDIF file named `03-dsee-roles.ldif`.

   a. Either copy the file into the `config/schema` directory of the directory server implementation and restart the server, or

   b. Use the `add schema file` task to cause the server to load the schema file into a running server instance.

2. Create a static or dynamic group to define role membership.

   Make sure that the group has an appropriate set of members.

3. Create a new instance of the `isMemberOf` virtual attribute to provide the `nsRole` virtual attribute.

   The `nsRole` attribute will include a list of the DNs of all groups in which the target user is a member. Use the `dsconfig` command to create the virtual attribute, as follows:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -n \
  create-virtual-attribute \
  --type is-member-of --name nsRole --set attribute-type:nsRole --set
enabled:true
```

### 19.5.2 To Alter Membership by Using the `nsRoleDN` Attribute

Follow this procedure if the application you are using expects to be able to alter membership by placing the name of the corresponding role in the `nsRoleDN` virtual attribute in a user's entry.

After these steps are completed, any user entry that contains an `nsRoleDN` value of `"cn=Test Role,ou=Roles,dc=example,dc=com"` also has that DN present in the `nsRole` operational attribute.

1. Create a dynamic group entry with the DN of the desired role.
2. Configure the group to include members that contain an `nsRoleDN` attribute with a value equal to the DN of the target role.

   For example, if the application is going to add an `nsRoleDN` value of `"cn=Test Role,ou=Roles,dc=example,dc=com"`, add the following entry:

   ```
   dn: cn=Test Role,ou=Roles,dc=example,dc=com
   objectClass: top
   objectClass: groupOfURLs
   cn: Test Role
   memberURL: ldap:///dc=example,dc=com??sub?(nsRoleDN=\
     cn=Test Role,ou=Roles,dc=example,dc=com)
   ```

# Part IV

# Advanced Administration: Security, Access Control, and Password Policies

This part describes how to configure all aspects of a deployment that relate to securing the servers themselves or securing the data that is stored in the directory.

This part includes the following chapters:

# 20

# Configuring Security Between Clients and Servers

Oracle Unified Directory provides several mechanisms to secure traffic between the client and the server. The topics in this section describe these mechanisms, and how to configure them.

For information about securing access to directory data, see Chapter 22, "Controlling Access To Data".

For information about configuring security between the proxy and the directory server or data source, see Chapter 21, "Configuring Security Between the Proxy and the Data Source".

This chapter covers the following topics:

- Section 20.1, "Getting SSL Up and Running Quickly"
- Section 20.2, "Configuring Key Manager Providers"
- Section 20.3, "Configuring Trust Manager Providers"
- Section 20.4, "Configuring Certificate Mappers"
- Section 20.5, "Configuring SSL and StartTLS for LDAP and JMX"
- Section 20.6, "Using SASL Authentication"
- Section 20.7, "Configuring SASL Authentication"
- Section 20.8, "Configuring Kerberos and the Oracle Unified Directory Server for GSSAPI SASL Authentication"
- Section 20.9, "Testing SSL, StartTLS, and SASL Authentication With `ldapsearch`"
- Section 20.10, "Debugging SSL Using OpenSSL `s_client` Test Utility"
- Section 20.11, "Debugging SSL or TLS Using Java Debug Information"
- Section 20.12, "Controlling Connection Access Using Allowed and Denied Rules"
- Section 20.13, "Configuring Unlimited Strength Cryptography"

## 20.1 Getting SSL Up and Running Quickly

Oracle Unified Directory provides a number of options for configuring and using SSL and StartTLS. The numerous possibilities for configuration might be daunting for those who are unfamiliar with the technology or who just want to get up and running as quickly as possible for testing purposes.

This section provides a list of the steps that must be performed to allow Oracle Unified Directory to accept SSL-based connections using a self-signed certificate.

The procedures in this section assume a knowledge of truststores and keystores. For detailed information about keystores, see Section 20.2, "Configuring Key Manager Providers". For detailed information about truststores, see Section 20.3, "Configuring Trust Manager Providers".

---

**Note:** Using a self-signed certificate is not recommended for production purposes. To install a certificate for production purposes, follow the instructions in Section 20.2, "Configuring Key Manager Providers".

---

### 20.1.1 To Accept SSL-Based Connections Using a Self-Signed Certificate

This procedure assumes the following:

- Oracle Unified Directory is installed on the system on which you are working.
- The Java `keytool` utility is in your path. If it is not, either add it to your path or provide the complete path to it when invoking the commands. The `keytool` utility is provided with the Java Runtime Environment (JRE).
- The administration connector is listening on the default port (4444) and the `dsconfig` command is accessing the server running on the local host. If this is not the case, the `--port` and `--hostname` options must be specified.

1. Generate a private key for the certificate, using the `keytool` command with the `-genkeypair` option.

   For example:

   ```
   $ keytool -genkeypair -alias server-cert -keyalg rsa \
     -dname "CN=myhost.example.com,O=Example Company,C=US" \
     -keystore config/keystore -storetype JKS
   ```

   - `-alias` *alias*. Specifies the name that should be used to refer to the certificate in the keystore. The default name used by the server is `server-cert`.
   - `-keyalg` *algorithm*. Specifies the algorithm that should be used to generate the private key. This should almost always be `rsa`.
   - `-dname` *subject*. Specifies the subject to use for the certificate.

     Change the value of the `-dname` argument so that it is suitable for your environment:

     The value of the `CN` attribute should be the fully-qualified name of the system on which the certificate is being installed.

     The value of the `O` attribute should be the name of your company or organization.

     The value of the `C` attribute should be the two-character abbreviation for your country.
   - `-keystore` *path*. Specifies the path to the keystore file. The file will be created if it does not already exist. The default keystore path used by the server is `config/keystore`.

- `-keypass` *password*. Specifies the password that should be used to protect the private key in the keystore. If the password is not provided, you will be prompted for it.
- `-storepass` *password*. Specifies the password that should be used to protect the contents of the keystore. If the password is not provided, you will be prompted for it. The server expects the password used for the `-keypass` and `-storepass` options to be the same.
- `-storetype` *type*. Specifies the keystore type that should be used. For the JKS keystore, for example, the value should always be `JKS`.

You are prompted for a password to protect the contents of the keystore and for a password to protect the private key.

2. Generate a self-signed certificate for the key.

   For example:

```
$ keytool -selfcert -alias server-cert -validity 1825 \
   -keystore config/keystore -storetype JKS
```

   - `-alias` *alias*. Specifies the name that should be used to refer to the certificate in the keystore. This name should be the same as the value used when creating the private key with the `-genkeypair` option.
   - `-validity` *days*. Specifies the length of time in days that the certificate should be valid. The default validity is 90 days.
   - `-keystore` *path*. Specifies the path to the keystore file. The file will be created if it does not already exist.
   - `-keypass` *password*. Specifies the password that should be used to protect the private key in the keystore. If this is not provided, then you will be interactively prompted for it.
   - `-storepass` *password*. Specifies the password that should be used to protect the contents of the keystore. If this is not provided, then you will be interactively prompted for it.
   - `-storetype` *type*. Specifies the keystore type that should be used. For the JKS keystore, the value should always be `JKS`.

   When you are prompted for the keystore password, enter the same password that you provided in the previous step.

3. Create a text file named `config/keystore.pin`.

   The file must contain the password that you chose to protect the contents of the keystore. If you change this file, remember that it must match the keystore manager configuration. If you decide to create a file with a different name, for example, the corresponding keystore manager's `key-store-file` property for JKS must match the path and file name.

4. Export the public key for the certificate that you created.

   For example:

```
$ keytool -exportcert -alias server-cert -file config/server-cert.txt -rfc \
   -keystore config/keystore -storetype JKS
```

5. Create a new trust store and import the server certificate into that trust store.

   For example:

```
$ keytool -importcert -alias server-cert -file config/server-cert.txt \
  -keystore config/truststore -storetype JKS
```

6. Type `yes` when you are prompted to trust the certificate.

   This step is required *only* if the SSL and StartTLS settings were not specified during installation, or if you want to change those settings.

7. Use the `dsconfig` command to enable the key manager provider, trust manager provider, and connection handler.

   For example:

```
$ dsconfig -D "cn=directory manager" -j pwd-file -X -n \
  set-key-manager-provider-prop --provider-name JKS --set enabled:true
$ dsconfig -D "cn=directory manager" -j pwd-file -X -n \
  set-trust-manager-provider-prop --provider-name "Blind Trust" \
  --set enabled:true
$ dsconfig -D "cn=directory manager" -j pwd-file -X -n \
  set-connection-handler-prop --handler-name "LDAPS Connection Handler" \
  --set "trust-manager-provider:Blind Trust" --set key-manager-provider:JKS \
  --set listen-port:1636 --set enabled:true
```

   Port `1636` is the standard LDAPS port, but you might not be able to use this port if it is already taken or if you are a regular user. If you need to accept SSL-based connections on a port other than `1636`, change the `listen-port` property in the last command to the port number being used.

   If, in step 3, you created a text file with a location and name other than `config/keystore.pin`, for example a text file called `config/mykeystore.pin`, specify that information as follows:

```
$ dsconfig -D "cn=directory manager" -j pwd-file -X -n \
  set-key-manager-provider-prop --provider-name JKS --set enabled:true \
  --set keystore-pin-file:/config/mykeystore.pin
```

   For detailed information about keystores, see Section 20.2, "Configuring Key Manager Providers". For detailed information about truststores, see Section 20.3, "Configuring Trust Manager Providers".

8. The server should now have a second listener that accepts SSL-based client connections. Test the configuration with the `ldapsearch` command, for example:

```
$ ldapsearch --port 1636 --useSSL --baseDN "" --searchScope base
"(objectClass=*)"
```

   You are prompted to trust the server's certificate. On typing `yes`, the root DSE entry should be returned.

## 20.2 Configuring Key Manager Providers

Key manager providers provide access to the certificate that should be used by the directory server when performing SSL or StartTLS negotiation.

This section covers the following topics:

- Section 20.2.1, "Key Manager Provider Overview"
- Section 20.2.2, "Using the JKS Key Manager Provider"
- Section 20.2.3, "Using the PKCS #12 Key Manager Provider"
- Section 20.2.4, "Using the PKCS #11 Key Manager Provider"

- Section 20.2.5, "Replacing a Certificate in a Production Server"
- Section 20.2.6, "Configuring Key Managers With ODSM"

For more information, see "Key Manager Provider Configuration" in the *Configuration Reference for Oracle Unified Directory*.

## 20.2.1 Key Manager Provider Overview

Oracle Unified Directory supports keystore formats for the following key manager providers:

- JKS keystore, which is the default keystore format used by Java Secure Socket Extension (JSSE)
- PKCS #12 file
- PKCS #11 device, such as a hardware security module or cryptographic accelerator

> **Note:** PKCS #11 is not supported for use with a proxy server instance.

The process for configuring Oracle Unified Directory to use these key manager providers is described in detail in the following sections.

The administration connector is an LDAPS connector. As with all SSL-based connectors, the administration connector requires a key manager. Oracle Unified Directory provides a dedicated key manager for the administration connector, that is enabled by default. For more information, see Section 14.3, "Managing Administration Traffic to the Server".

## 20.2.2 Using the JKS Key Manager Provider

The JKS keystore is the default keystore used by most JSSE implementations, and is the preferred keystore type in many environments. To configure the server to use this keystore type, you must first obtain a JKS keystore that contains a valid certificate. To do this, you can either generate a self-signed certificate or issue a certificate signing request to an existing Certificate Authority (CA) and import the signed certificate.

All of the steps described here require the use of the `keytool` utility, which is provided with the Java runtime environment. This utility is typically found in the `bin` directory below the root of the Java installation. For more information about using the `keytool` utility, see the official Java documentation (`http://download.oracle.com/javase/6/docs/technotes/tools/windows/keytool.html`).

Using the JKS key manager provider involves the following:

1. Generating the private key
2. Self-signing the certificate, or using an external certificate authority to sign the certificate
3. Configuring the JKS key manager provider

### 20.2.2.1 To Generate the Private Key

Whether you use a self-signed certificate or generate a certificate signing request, you must first generate a private key. You can do this using the `keytool` utility with the `-genkeypair` option. The following arguments can be used with this option:

- `-alias` *alias*. Specifies the name that should be used to refer to the certificate in the keystore. The default name used by server is `server-cert`.
- `-keyalg` *algorithm*. Specifies the algorithm that should be used to generate the private key. This should almost always be `rsa`.
- `-dname` *subject*. Specifies the subject to use for the certificate. The subject typically contains at least a `CN` attribute, which is the fully-qualified name of the system on which the certificate will be installed, an `O` attribute that specifies the name of the organization (or company), and a `C` attribute that specifies the country in which the certificate will be used.
- `-keystore` *path*. Specifies the path to the keystore file. The file will be created if it does not already exist. The default keystore path used by the directory server is `config/keystore`.
- `-keypass` *password*. Specifies the password that should be used to protect the private key in the keystore. If the password is not provided, you will be prompted for it.
- `-storepass` *password*. Specifies the password that should be used to protect the contents of the keystore. If the password is not provided, you will be prompted for it. The directory server expects the password used for the `-keypass` and `-storepass` options to be the same.
- `-storetype` *type*. Specifies the keystore type that should be used. For the JKS keystore, the value should always be `JKS`.

Use the `keytool -genkeypair` command to create a private key, as follows:

```
$ keytool -genkeypair -alias server-cert -keyalg rsa \
  -dname "CN=server.example.com,O=example.com,C=US" \
  -keystore config/keystore -keypass password \
  -storetype JKS -storepass password
```

#### 20.2.2.2 To Self-Sign the Certificate

If the certificate is to be self-signed, use the `-selfcert` option. The most important arguments for use with this option include:

- `-alias` *alias*. Specifies the name that should be used to refer to the certificate in the keystore. This name should be the same as the value used when creating the private key with the `-genkeypair` option.
- `-validity` *days*. Specifies the length of time in days that the certificate should be valid. The default validity is 90 days.
- `-keystore` *path*. Specifies the path to the keystore file. The file will be created if it does not already exist.
- `-keypass` *password*. Specifies the password that should be used to protect the private key in the keystore. If this is not provided, then you will be interactively prompted for it.
- `-storepass` *password*. Specifies the password that should be used to protect the contents of the keystore. If this is not provided, then you will be interactively prompted for it.
- `-storetype` *type*. Specifies the keystore type that should be used. For the JKS keystore, the value should always be `JKS`.

Use the `keytool -selfcert` command to generate a self-signed certificate, as follows:

```
$ keytool -selfcert -alias server-cert -validity 1825 \
  -keystore config/keystore -keypass password -storetype JKS \
  -storepass password
```

#### 20.2.2.3 To Sign the Certificate by Using an External Certificate Authority

If the certificate is to be signed by an external certificate authority, you must first generate a certificate signing request (CSR) using the `-certreq` option. The CSR can be submitted to a certificate authority to be signed. The method for doing this, and the method for obtaining the signed certificate, might vary from one certificate authority to another.

When you receive the signed certificate from the Certificate Authority, import it into the keystore with the `-importcert` option.

1. Use the `-certreq` option to obtain a certificate signing request.

   ```
   $ keytool -certreq -alias server-cert -file /tmp/server-cert.csr \
     -keystore config/keystore -keypass password -storetype JKS \
     -storepass password
   ```

   The arguments used with this command are as follows:

   - `-alias` *alias*. Specifies the name that should be used to refer to the certificate in the keystore. This name should be the same as the value used when creating the private key with the `-genkeypair` option.
   - `-file` *path*. Specifies the path to the file to which the CSR should be written. If this is not provided, the request will be written to standard output.
   - `-keystore` *path*. Specifies the path to the keystore file. The file will be created if it does not already exist.
   - `-keypass` *password*. Specifies the password that should be used to protect the private key in the keystore. If this is not provided, you will be interactively prompted for it.
   - `-storepass` *password*. Specifies the password that should be used to protect the contents of the keystore. If this is not provided, you will be interactively prompted for it.
   - `-storetype` *type*. Specifies the keystore type that should be used. For the JKS keystore, the value should always be `JKS`.

2. Send the certificate request to an external certificate authority. The certificate authority will send you a signed certificate file. Save the file in `/tmp/server-cert.txt`

3. Use the `-importcert` to import the signed certificate.

   ```
   $ keytool -importcert -alias server-cert -file /tmp/server-cert.cert \
     -keystore config/keystore -storetype JKS -storepass password
   ```

   The arguments used with this command are as follows:

   - `-alias` *alias*. Specifies the name that should be used to refer to the certificate in the keystore. This name should be the same as the value used when creating the private key with the `-genkeypair` option.
   - `-file` *path*. Specifies the path to the file containing the signed certificate. The file should be in either the DER-encoded binary format or the base64-encoded ASCII format as described in RFC 1421 (`http://www.ietf.org/rfc/rfc1421.txt`).

- `-keystore` *path*. Specifies the path to the keystore file. The file will be created if it doesn't already exist.
- `-storepass` *password*. Specifies the password that should be used to protect the contents of the keystore. If this is not provided, then you will be interactively prompted for it.
- `-storetype` *type*. Specifies the keystore type that should be used. For the JKS keystore, the value should always be `JKS`.

#### 20.2.2.4 To Configure the JKS Key Manager Provider

When you have created a JKS keystore containing a signed certificate (whether self-signed or signed by an external CA), you can configure the server to use that keystore by configuring a key manager provider entry for that keystore.

This example uses `dsconfig` to configure the properties of the default JKS key manager provider. For details about all the properties of the key manager provider, see "File Based Key Manager Provider Configuration" in the *Oracle Unified Directory Configuration Reference*.

Use the `dsconfig` command to configure the key manager provider entry.

```
$ dsconfig -D "cn=Directory Manager" -j pwd-file -X -n \
  set-key-manager-provider-prop --provider-name "JKS" \
  --set enabled:true --set "key-store-type:JKS" \
  --set "key-store-file:config/keystore" \
  --set "key-store-pin:password"
  --reset key-store-pin-file
```

### 20.2.3 Using the PKCS #12 Key Manager Provider

PKCS #12 is a standard format for storing certificate information, including private keys. Oracle Unified Directory can use a PKCS #12 file as a certificate keystore if it includes the private key for the certificate.

Because PKCS #12 is a common format for storing certificate information, you might already have a certificate in this format, or the certificate authority (CA) that you use might create certificates in this form. In some cases, it might also be possible to convert an existing certificate into PKCS #12 format. For example, if you already have a certificate in a Network Security Services (NSS) certificate database, then the NSS `pk12util` tool can import it. The following example uses the `pk12util` tool to export a certificate named `server-cert` contained in the database `../../alias/slapd-config-key3.db` to a PKCS #12 file, `/tmp/server-cert.p12`:

```
$ ./pk12util -n server-cert -o /tmp/server-cert.p12 \
  -d ../../alias -P "slapd-config-"
```

To create a new certificate in PKCS #12 format, use the procedure described in Section 20.2.2, "Using the JKS Key Manager Provider" for obtaining a certificate in a JKS keystore. The only difference in the process is that you should use `-storetype PKCS12` instead of `-storetype JKS` when you invoke the `keytool` commands. For example, to create a self-signed certificate in a PKCS #12 file, use the following commands:

```
$ keytool -genkeypair -alias server-cert -keyalg rsa \
  -dname "CN=server.example.com,O=example.com,C=US" \
  -keystore config/keystore.p12 -keypass password \
  -storetype PKCS12 -storepass password
```

```
$ keytool -selfcert -alias server-cert -validity 1825 \
  -keystore config/keystore.p12 -keypass password \
  -storetype PKCS12 -storepass password
```

As with JKS, the server provides a template key manager provider for use with PKCS #12 certificate files that uses the same set of configuration attributes as the configuration entry for the JKS key manager provider. The only differences are that the value of the `key-store-type` attribute must be `PKCS12`, and the `key-store-file` attribute should refer to the location of the PKCS #12 file rather than a JKS keystore. The following example uses `dsconfig` to configure the PKCS #12 keystore manager provider:

```
$ dsconfig -D "cn=directory manager" -j pwd-file -X -n\
  set-key-manager-provider-prop --provider-name "PKCS12" --set enabled:true \
  --set java-class:org.opends.server.extensions.FileBasedKeyManagerProvider \
  --set enabled:true --set "key-store-type:PKCS12" \
  --set "key-store-file:/config/keystore" \
  --set "key-store-pin:secret"
```

For a complete list of configurable properties, see "File Based Key Manager Provider Configuration" in the *Configuration Guide for Oracle Unified Directory*.

### 20.2.4 Using the PKCS #11 Key Manager Provider

PKCS #11 is a standard interface used for interacting with devices capable of holding cryptographic information and performing cryptographic functions. The PKCS #11 interface has two common uses of interest for the directory server:

- Cryptographic accelerators use this interface to allow products to offload their cryptographic processing to an external board (or in some cases, a special module inside the system's CPU or a framework inside the OS kernel), which might provide better performance for those operations.
- Hardware security modules (HSMs) use this interface to provide a secure repository for storing key information. This significantly reduces the likelihood that sensitive key information will be exposed and helps protect the overall integrity of the secure communication mechanisms.

> **Note:** The PKCS #11 format is not supported for use with a proxy server instance.

At present, the PKCS #11 support that Oracle Unified Directory provides has been tested and verified only on systems running at least Solaris 10 (on SPARC and x86/x64 systems) through the use of the Solaris OS cryptographic framework. Any device that plugs into this Solaris cryptographic framework should be supported in this manner. This includes the *softtoken* device, which is simulated in software and is therefore available on all systems supporting the Solaris cryptographic framework regardless of whether they have a hardware device providing PKCS #11 support.

If you do have a third-party PKCS #11 device installed in a Solaris system, it is likely that the Solaris OS cryptographic framework is already configured to access that device. However, if you will simply be using the software token or if you are running on a Sun Fire T1000 or T2000 system and want to take advantage of the cryptographic processor included in the UltraSPARC—T1 CPU, you will likely need to initialize the PKCS #11 interface. This should first be accomplished by choosing a PIN to use for the certificate store, which can be done with this command:

```
$ pktool setpin
```

This command prompts you for the current passphrase. If you have not yet used the Solaris OS cryptographic framework, the default passphrase is `changeme`. You are then prompted twice for the new password.

> **Note:** This step should be done while you are logged in as the user or as the role that will be used to run the directory server, because each user might have a different set of certificates.

At this point, it should be possible to use the Java `keytool` utility to interact with the Solaris cryptographic framework through PKCS #11. This will work much in the same way as it does when working with JKS or PKCS#12 keystores, with the following exceptions:

- The value of the `-keystore` argument must be `NONE`.
- The value of the `-storetype` argument must be `PKCS11`.
- You should not use the `-keypass` argument, and the tool will not prompt you for that password interactively if you do not provide it.
- The value of the `-storepass` argument must be the passphrase that you chose when using the `pktool setpin` command. Alternately, if you do not provide this argument on the command line, this is the password that you should enter when prompted.

For example, the following commands use the PKCS #11 interface to generate a self-signed certificate through the Solaris cryptographic framework:

```
$ keytool -genkeypair -alias server-cert -keyalg rsa \
  -dname "CN=server.example.com,O=example.com,C=US" \
  -keystore NONE -storetype PKCS11 -storepass password

$ keytool -selfcert -alias server-cert -validity 1825 \
  -keystore NONE -storetype PKCS11 -storepass password
```

When the certificate is installed in the PKCS #11 keystore, the directory server must be configured to use that keystore. Configure the PKCS #11 keystore provider in the same way as the entry for the JKS and PKCS#12 keystore manager providers, with the exception that the `key-store-file` attribute is not included. However, a PIN is still required and is provided either directly, in a PIN file, through a Java property, or through an environment variable.

The following example uses `dsconfig` to configure the PKCS #11 key manager provider:

```
$ dsconfig -D "cn=directory manager" -j pwd-file -X -n \
  set-key-manager-provider-prop --provider-name "PKCS11" --set enabled:true \
  --set enabled:true --set "key-store-type:PKCS11" \
  --set "key-store-file:/config/keystore" \
  --set "key-store-pin:secret"
```

For a complete list of configurable properties, see "PKCS11 Key Manager Provider Configuration" in the *Configuration Guide for Oracle Unified Directory*.

### 20.2.5 Replacing a Certificate in a Production Server

To replace a certificate in a production server, request the new certificate and configure the appropriate key manager provider, as described in Section 20.2.2, "Using the JKS Key Manager Provider", Section 20.2.3, "Using the PKCS #12 Key Manager Provider" or Section 20.2.4, "Using the PKCS #11 Key Manager Provider".

The `key-manager-provider` property of the SSL-based connection handler (named "LDAPS" by default) specifies the keystore manager that must be used for security. The default value of the `key-manager-provider` property is "JKS", which means that the SSL connection handler uses the JKS key manager provider by default. If you are using a different key manager provider, change this property of the SSL connection handler accordingly.

There is no need to restart the server after the new certificate is installed. The new certificate is used immediately for subsequent attempts to access the server for associated client connections. Existing connections are not reestablished.

### 20.2.6 Configuring Key Managers With ODSM

You can manage the key manager configuration by using ODSM, as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager".
2. Select the **Configuration** tab.
3. Under General Configuration, expand the **Key Managers** item.
4. Select the key manager you want to configure.

   The configurable properties of the key manager are displayed in the right hand pane.
5. Edit the key manager configuration, as required, and click **Apply** to save your changes.

## 20.3 Configuring Trust Manager Providers

Oracle Unified Directory uses trust manager providers to determine whether to trust a certificate that is presented to it. Trust managers serve an important role in the overall security of the system by ensuring that the peer (the system at the other end of the connection, whether it is an inbound connection from a client or an outbound connection to another server) is who it claims to be.

This section covers the following topics:

- Section 20.3.1, "Overview of Certificate Trust Mechanisms"
- Section 20.3.2, "Using the Blind Trust Manager Provider"
- Section 20.3.3, "Using the JKS Trust Manager Provider"
- Section 20.3.4, "Using the PKCS #12 Trust Manager Provider"
- Section 20.3.5, "Configuring Trust Managers With ODSM"

### 20.3.1 Overview of Certificate Trust Mechanisms

A trust manager provider can improve security whenever SSL or StartTLS is used by thwarting attempts to use forged certificates and foiling man-in-the-middle attacks.

The two primary use cases for trust manager providers are as follows:

- Inbound connections: a client presents its own certificate to the server during the SSL or StartTLS negotiation process, potentially for use in SASL EXTERNAL authentication.
- Outbound connections: the server attempts to establish an SSL-based connection to an external system, for example for the purpose of synchronization or for proxied or chained operations.

The trust manager has no impact on the strength of the encryption, so only the server and its peer will be able to understand the communication. Any third-party observer will be unable to decipher the exchange. The trust manager is responsible for ensuring that the peer is who it claims to be so that confidential information is not inadvertently exposed to one peer masquerading as another.

The trust manager considers a number of factors to determine whether a peer certificate should be trusted. This topic describes some of the most common criteria that are taken into account during this process.

One of the simplest trust mechanisms is the validity period for the certificate. All certificates have a specific window during which they should be considered valid, bounded by "notBefore" and "notAfter" time stamps. If the current time is beyond the "notAfter" time stamp, the certificate is expired and trust managers reject it. Similarly, certificates are also typically rejected if the current time is before the "notBefore" time stamp. Most often, the "notBefore" time stamp is set to the time that the certificate was signed, but there are cases in which a certificate might be issued that is not immediately valid. In those cases, it is important to ensure that the peer is not granted access too early.

Another very important factor in deciding whether to trust a peer certificate is the peer certificate chain. When one system presents its certificate to another, it does not present its certificate only, but a chain of certificates that describes all entities involved in the process. When a trust manager is attempting to determine whether to trust a peer, the trust manager first looks in its trust store to determine whether it contains the peer certificate. If that certificate is found, the peer will be trusted (barring rejection for another reason, such as being outside the validity period). If the peer's certificate is not found, the trust manager looks at the next certificate in the chain, which will be the certificate that was used to sign the peer's certificate (also called the issuer certificate). If the trust store contains the issuer's certificate, the server will trust that issuer certificate and will also implicitly trust any certificate that it has signed. This process continues up the certificate chain (looking at the certificate that signed the issuer certificate, and so on) until one of the certificates is found in the trust store or until the root of the chain is reached (in which case, the root certificate will be self-signed and therefore will be its own issuer). If none of the certificates in the peer chain is contained in the trust store, the peer's certificate is rejected.

This process makes it much easier to manage an environment with a large number of certificates (for example, one in which there is a large number of servers or in which many clients use SASL EXTERNAL authentication). It is not necessary for the trust store to have each individual peer certificate. The trust store can contain only one of the certificates in the peer chain. For example, if all of the certificates that might legitimately be presented to the server were signed by the same issuer, it is necessary to have only that issuer's certificate in the trust store in order to implicitly trust any of the peers.

In some environments, there might be other elements taken into account when deciding to trust a peer certificate chain. For example, there might be a certificate revocation list (CRL) that contains a list of all of the certificates that have been revoked and should no longer be considered valid even if they are still within their validity period and were signed by a trusted issuer. This can be useful, for example, if the

certificate belonged to an employee that has left the company or if the private key for the certificate has been compromised. The Online Certificate Status Protocol (OCSP, as described in RFC 2560 (`http://www.ietf.org/rfc/rfc2560.txt`) also provides a similar mechanism, in which the trust manager might ask an OCSP server whether a given certificate is still valid. Oracle Unified Directory currently does not support using CRLs or OCSP when attempting to determine whether a peer certificate chain should be trusted.

The administration connector is an LDAPS connector. As with all SSL-based connectors, the administration connector requires a trust manager. Oracle Unified Directory provides a dedicated trust manager for the administration connector, that is enabled by default. For more information, see Section 14.3, "Managing Administration Traffic to the Server".

### 20.3.2 Using the Blind Trust Manager Provider

The blind trust manager provider is a simple provider that trusts any certificate that is presented to it. It does not look at the expiration date, who signed the certificate, the subject or alternate names, or any other criteria.

Oracle Unified Directory provides a blind trust manager provider that is disabled by default. You can enable the provider by changing the value of the `enabled` attribute to `true`. The blind trust manager provider does not require any other configuration attributes.

> **Note:** The blind trust manager provider is not supported with a proxy server instance.

The following example uses `dsconfig` to configure the blind trust manager provider:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X \
  set-trust-manager-provider-prop --provider-name "Blind Trust"
```

For a list of the configurable properties, see the "Blind Trust Manager Provider Configuration" in the *Configuration Reference for Oracle Unified Directory*.

> **Caution:** The blind trust manager provider is provided as a convenience for testing purposes only and should never be used in a production server, especially one that is configured to allow SASL EXTERNAL authentication. If a client attempts to use SASL EXTERNAL to authenticate to using a certificate and the server blindly accepts any certificate that the client presents, the user can create a self-signed certificate that allows it to impersonate any user in the directory.

### 20.3.3 Using the JKS Trust Manager Provider

Just as the JKS keystore can be used to provide the key material for a key manager provider, it can also be used to provide information that can used by trust manager providers. In general, using a JKS file as a trust store is similar to using it as a keystore. However, because private key information is not accessed when the file is used as a trust store, there is generally no need for a PIN when accessing its contents.

When the JKS trust manager provider determines whether to trust a given peer certificate chain, it considers two factors:

- Is the peer certificate within the validity period?
- Is any certificate in the chain contained in the trust store?

If the peer certificate is not within the validity period or none of the certificates in the peer certificate chain are contained in the trust store, the JKS trust manager rejects that peer certificate.

Use the `keytool -importcert` utility to import certificates into a JKS trust store. The `-importcert` option uses these arguments:

- `-alias alias`. Specifies the name to give to the certificate in the trust store. Give each certificate a unique name, although the nickname is primarily for managing the certificates in the trust store and has no impact on whether a certificate is trusted.
- `-file path`. Specifies the path to the file containing the certificate to import. The file can be in either DER format or in base64-encoded ASCII format, as described in RFC 1421 (`http://www.ietf.org/rfc/rfc1421.txt`).
- `-keystore path`. Specifies the path to the file used as the JKS trust store. This path is typically `config/truststore`.
- `-storetype type`. Specifies the format of the trust store file. For the JKS trust manager, this must be `JKS`.
- `-storepass password`. Specifies the password used to protect the contents of the trust store. If the trust store file does not exist, this value is the password to assign to the trust store, and must be used for future interaction with the trust store. If this option is not provided, the password is interactively requested from the user.

The following command provides an example of importing a certificate into a JKS trust store. If the trust store does not exist, this command creates the trust store before importing the certificate.

```
$ keytool -importcert -alias server-cert -file /tmp/cert.txt \
  -keystore config/truststore -storetype JKS -storepass password
```

Oracle Unified Directory provides a template JKS trust manager provider. Use `dsconfig` to configure the following properties of the JKS trust manager provider:

- `enabled`. Indicates whether the JKS trust manager provider is enabled. The JKS trust manager provider is not available for use by other server components unless the value of this property is `true`.
- `trust-store-file`. The path to the trust store file, which is typically `config/truststore`, although an alternate file can be used if needed. The value of this property can be either an absolute path or a path that is relative to the *INSTANCE_DIR*.
- `trust-store-type`. The format of the trust store. For the JKS trust store provider, the value of this property is `JKS`.

The following example uses `dsconfig` in interactive mode to configure the JKS trust manager provider:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X \
  set-trust-manager-provider-prop --provider-name "JKS"
```

For a list of the configurable properties, see the "File Based Trust Manager Provider Configuration" in the *Configuration Reference for Oracle Unified Directory*.

### 20.3.4 Using the PKCS #12 Trust Manager Provider

The PKCS #12 trust manager provider is primarily useful if you already have the peer or issuer certificates to be used in a PKCS #12 file. If you do not have the certificates in this format, use the JKS trust manager provider instead. The Java `keytool` utility does not currently support importing trusted certificates (that is, those with just a public key and no private key information) into a PKCS #12 file.

Oracle Unified Directory provides a template PKCS #12 trust manager provider. Use `dsconfig` to configure the following properties of the PKCS #12 trust manager provider:

- `enabled`. Indicates whether the PKCS #12 trust manager provider is enabled. The trust manager provider is not available for use by other server components unless this property has a value of `true`.
- `trust-store-type`. Specifies the format of the trust store. For the PKCS #12 trust manager provider, the value is `PKCS12`.
- `trust-store-file`. Specifies the path to the trust store file, which is typically `config/truststore.p12`, although an alternate file can be used if needed. The value of this property can be either an absolute path or a path that is relative to the *INSTANCE_DIR*.

A PIN might be required to access the contents of the PKCS #12 file. In this case, one of the following configuration attributes must be used to provide the password. (At the present time, the password must be provided in clear text.)

- `trust-store-pin`. Specifies the PIN needed to access the trust store directly.
- `trust-store-pin-file`. Specifies the path to a file containing the PIN needed to access the trust store. The value of this property can be either an absolute path or a path that is relative to the server root.
- `trust-store-pin-property`. Specifies the name of a Java property that holds the PIN needed to access the trust store.
- `trust-store-pin-environment-variable`. Specifies the name of an environment variable that holds the PIN needed to access the trust store.

  The following example uses `dsconfig` in interactive mode, to configure the PKCS #12 trust manager provider:

  ```
  $ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X \
    set-trust-manager-provider-prop --provider-name "PKCS12"
  ```

  For a list of the configurable properties, see the "File Based Trust Manager Provider Configuration" in the *Configuration Reference for Oracle Unified Directory*.

### 20.3.5 Configuring Trust Managers With ODSM

You can manage the trust manager configuration by using ODSM, as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager".
2. Select the **Configuration** tab.
3. Under General Configuration, expand the **Trust Managers** item.
4. Select the trust manager you want to configure.

   The configurable properties of the trust manager are displayed in the right hand pane.

5. Edit the trust manager configuration, as required, and click **Apply** to save your changes.

## 20.4 Configuring Certificate Mappers

A *certificate mapper* examines a certificate presented by a client and maps it to the user in the directory that should be associated with that certificate.

Certificate mappers are configured for directory server instances only - not for proxy or gateway instances.

Certificate mappers are primarily used in the context of processing SASL EXTERNAL authentication, where the client wants to authenticate to the server using its SSL certificate rather than a password or some other form of credentials.

Oracle Unified Directory provides the following certificate mappers by default:

- Subject Equals DN
- Subject Attribute to User Attribute
- Subject DN to User Attribute
- Fingerprint Mapper

You can also create a custom certificate mapper to suit the requirements of your deployment.

A certificate mapper is defined either at the global server configuration level, or at the network group level. If a certificate mapper is defined for the network group, that certificate mapper overrides what is defined in the global server configuration. If no certificate mapper is defined for a network group, the global certificate mapper is used. To define the certificate mapper that should be used, set the `certificate-mapper` property of the global configuration, or the network group.

The examples in this section use the `dsconfig` command to modify certificate mappers. The `dsconfig` command accesses the server configuration over SSL, using the administration connector. For more information, see Section 14.1, "Managing the Server Configuration With `dsconfig`".

### 20.4.1 Using the Subject Equals DN Certificate Mapper

The Subject Equals DN certificate mapper is a simple certificate mapper that expects the subject of the client certificate to be exactly the same as the distinguished name (DN) of the corresponding user entry. Using this certificate mapper is easy because there are no configuration attributes associated with it. However, this mapper is not suitable for many environments because certificate subjects and user DNs are often not the same.

The server uses the Subject Equals DN certificate mapper by default. To change the certificate mapper that is used by the server, set the appropriate global configuration property by using `dsconfig`. The following command changes the certificate mapper that the server uses from Subject Equals DN to Subject Attribute to User.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  set-global-configuration-prop \
  --set certificate-mapper:"Subject Attribute to User Attribute"
```

You cannot disable the Subject Equals DN certificate mapper if it is referenced by the global server configuration. To disable the mapper, you must change the default certificate mapper, as described previously.

## 20.4.2 Using the Subject Attribute to User Attribute Certificate Mapper

The Subject Attribute to User Attribute certificate mapper attempts to map a client certificate to a user entry based on a set of attributes that they have in common. In particular, it takes the values of a specified set of attributes from the certificate subject and attempts to locate user entries that contain those same values in a corresponding set of attributes.

Use `dsconfig` to set the properties of this certificate mapper:

- `subject-attribute-mapping`. A multi-valued property that maps attributes from the certificate subject to attributes in user entries. Values for this attribute consist of the name of the attribute in the certificate subject followed by a colon and the name of the corresponding attribute in the user's entry. For example, the value `e:mail` maps the `e` attribute from the certificate subject to the `mail` attribute in user entries. At least one attribute mapping must be defined. The default mappings are `e:mail` and `cn:cn`.
- `user-base-dn`. A multi-valued property that specifies the set of base DNs below which the server should look for matching entries. If this attribute has no value, the server searches below all public naming contexts.

The following example uses `dsconfig` to configure the Subject Attribute to User Attribute certificate mapper, specifying that the server should search only below `ou=people,dc=example,dc=com`:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  set-certificate-mapper-prop \
  --mapper-name "Subject Attribute to User Attribute" \
  --set user-base-dn:ou=people,dc=example,dc=com
```

If multiple attribute mappings are defined, the server combines them with an AND search. For example, if two mappings are defined `cn:cn` and `e:mail`, and the server is presented with a certificate that has a subject of `E=john.doe@example.com,CN=John Doe,O=Example Corp,C=US`, then it generates a search filter of `(&(cn=John Doe)(mail=john.doe@example.com))`. Any attribute for which a mapping is defined but is not contained in the certificate subject is not included in the generated search filter. All attributes that can be used in generated search filters should have corresponding indexes in all remote LDAP databases that can be searched by this certificate mapper.

For the mapping to be successful, the generated search filter must match exactly one user in the directory (within the scope of the base DNs for the mapper). If no users match the generated criteria or if multiple users match, the mapping fails.

## 20.4.3 Using the Subject DN to User Attribute Certificate Mapper

The Subject DN to User Attribute certificate mapper attempts to establish a mapping by searching for the subject of the provided certificate in a specified attribute in user entries. In this case, you must ensure that user entries are populated with the subjects of the certificates associated with those users. However, it is possible that this process could be automated in the future with a plug-in that automatically identifies any certificates contained in a user entry and adds the subjects of those certificates to a separate attribute.

Use `dsconfig` to set the properties of this certificate mapper:

- `subject-attribute`. This is a single-valued attribute whose value is the name of the attribute type that should contain the certificate subject in user entries. This

attribute must be defined in the server schema, and it should be indexed for equality in all back ends that might be searched.

The subject DN of the certificate received by the server will not contain any spaces between its RDN components, even though the certificate might have been created with them. The value of the `subject-attribute` in the user entries must also not contain any spaces between the RDN components, so that they will correctly match the subject DN of the received certificate. For example, if the original certificate looks like:

```
keytool -printcert -file cert.002
Owner: CN=test,  O=Test Certificate
Issuer: CN=test,  O=Test Certificate
Serial number: 49b55976
Valid from: Mon Mar 09 19:01:26 MET 2009 until: Sat Mar 08 19:01:26 MET 2014
Certificate fingerprints:
         MD5:  5E:08:78:36:DF:25:F4:6C:43:9E:7B:CF:1F:1E:B9:6B
         SHA1: B7:B9:1C:A0:B0:52:C3:87:3C:C2:70:27:11:6F:5E:58:C5:33:9D:6B
         Signature algorithm name: SHA1withRSA
         Version: 3
```

The subject DN defined in the subject-attribute of the user entry should be:

```
CN=test,O=Test Certificate
```

Note the removal of the space between the RDN components of the `subject-attribute`.

- `user-base-dn`. This is a multivalued attribute that is used to specify the set of base DNs below which the server should look for matching entries. If this is not present, then the server will search below all public naming contexts.

The following example uses `dsconfig` to configure the Subject DN to User Attribute certificate mapper, specifying that the server should search only below `ou=people,dc=example,dc=com`:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  set-certificate-mapper-prop \
  --mapper-name "Subject DN to User Attribute" \
  --set user-base-dn:ou=people,dc=example,dc=com
```

Although there is no standard attribute for holding the subjects of the certificates that a user might hold, does define a custom attribute type, `ds-certificate-subject-dn`, that can be used for this purpose. This attribute can be added to user entries along with the `ds-certificate-user` auxiliary object class. This attribute is multivalued. If a user has multiple certificates, the attribute should contain the subjects for each of them as separate values.

This attribute is not indexed by default, so if it is to be used, update the corresponding back ends so that they contain an equality index for this attribute.

For the mapping to be successful, the certificate mapper must match exactly one user (within the scope of the base DNs for the mapper). If no entries match or if multiple entries match, the mapping fails.

## 20.4.4 Using the Fingerprint Certificate Mapper

The Fingerprint certificate mapper attempts to establish a mapping by searching for the MD5 or SHA1 fingerprint of the provided certificate in a specified attribute in user entries. In this case, you must ensure that user entries are populated with the certificate fingerprints (in standard hexadecimal notation with colons separating the

individual bytes, for example, `07:5A:AB:4B:E1:DD:E3:05:83:C0:FE:5F:A3:E8:1E:EB`). In the future, this process could be automated by a plug-in that automatically identifies any certificates contained in user entries and adds the fingerprints of those certificates to the appropriate attribute.

Use `dsconfig` to set the properties of this certificate mapper:

- `fingerprint-attribute`. Specifies a single-valued attribute whose value is the name of the attribute type that should contain the certificate fingerprint in user entries. This attribute must be defined in the server schema, and it should be indexed for equality in all back ends that can be searched.
- `fingerprint-algorithm`. Specifies which digest algorithm to use to calculate certificate fingerprints. The value is either `MD5` or `SHA1`.
- `user-base-dn`. Specifies a multi-valued attribute that is used to specify the set of base DNs below which the server is to look for matching entries. If this property is not present, then the server searches below all public naming contexts.

The following example uses `dsconfig` to configure the Fingerprint certificate mapper, specifying that the server should search only below `ou=people,dc=example,dc=com`:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  set-certificate-mapper-prop \
  --mapper-name "Fingerprint Mapper" \
  --set user-base-dn:ou=people,dc=example,dc=com
```

Although there is no standard attribute for holding certificate fingerprints, does define a custom attribute type, `ds-certificate-fingerprint`, that can be used for this purpose. This attribute can be added to user entries along with the `ds-certificate-user` auxiliary object class. This attribute is multi-valued, and if a user has multiple certificates, then it should contain the fingerprints for each of them as separate values. However, this attribute type is not indexed by default in any of the server back ends, so if it is to be used, add the corresponding equality index to all appropriate back ends.

For the mapping to be successful, the certificate mapper must match exactly one user (within the scope of the base DNs for the mapper). If no entries match or if multiple entries match, the mapping fails.

## 20.5 Configuring SSL and StartTLS for LDAP and JMX

When you have configured Oracle Unified Directory with at least one enabled key manager provider and at least one enabled trust manager provider, you can enable SSL and StartTLS for the connection handlers.

The examples in this section use the `dsconfig` command to modify the server configuration. The `dsconfig` command accesses the server configuration over SSL via the administration connector. As such, the relevant connection options must be specified, including how the SSL certificate is trusted. These examples use the `-X` option to trust all certificates.

This section covers the following topics:


- Section 20.5.1, "Configuring the LDAP and LDAPS Connection Handlers"
- Section 20.5.2, "Enabling SSL in the JMX Connection Handler"

## 20.5.1 Configuring the LDAP and LDAPS Connection Handlers

The LDAP connection handler is responsible for managing all communication with clients using LDAP. By default, the LDAP protocol does not specify any form of security for protecting that communication, but it can be configured to use SSL or also to allow the use of the StartTLS extended operation.

The server configures two connection handlers that can be used for this purpose. While the LDAP connection handler entry is enabled by default and is used to perform unencrypted LDAP communication, it can also be configured to support StartTLS. For information, see To Enable StartTLS Support. The LDAPS connection handler entry is disabled, but the default configuration is set up for To Enable SSL-Based Communication.

This section describes how to configure LDAP and LDAPS connection handler parameters with `dsconfig` and includes the following topics:

- Section 20.5.1.1, "To Enable a Connection Handler"
- Section 20.5.1.2, "To Specify a Connection Handler's Listening Port"
- Section 20.5.1.3, "To Specify a Connection Handler's Authorization Policy"
- Section 20.5.1.4, "To Specify a Nickname for a Connection Handler's Certificate"
- Section 20.5.1.5, "To Specify a Connection Handler's Key Manager Provider"
- Section 20.5.1.6, "To Specify a Connection Handler's Trust Manager Provider"
- Section 20.5.1.7, "To Enable StartTLS Support"
- Section 20.5.1.8, "To Enable SSL-Based Communication"

### 20.5.1.1 To Enable a Connection Handler

Set the `enabled` property of the connection handler to `true`.

This example enables the LDAP connection handler.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  set-connection-handler-prop --handler-name "LDAP Connection Handler" \
  --set enabled:true
```

### 20.5.1.2 To Specify a Connection Handler's Listening Port

Set the `listen-port` property of the connection handler.

The `listen-port` property specifies the port number to use when communicating with the server through this connection handler. The standard port to use for unencrypted LDAP communication (or LDAP using StartTLS) is `389`, and the standard port for SSL-encrypted LDAP is `636`. However, it might be desirable or necessary to change this in some environments (for example, if the standard port is already in use, or if you are running on a UNIX system as a user without sufficient privileges to bind to a port below `1024`).

This example sets the LDAPS connection handler's listen port to `1636`.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  set-connection-handler-prop --handler-name "LDAPS Connection Handler" \
  --set listen-port:1636
```

### 20.5.1.3 To Specify a Connection Handler's Authorization Policy

Set the `ssl-client-auth-policy` property of the connection handler.

The `ssl-client-auth-policy` property specifies how the connection handler should behave when requesting a client certificate during the SSL or StartTLS negotiation process. If the value is `optional`, the server requests that the client present its own certificate but still accepts the connection even if the client does not provide a certificate. If the value is `required`, the server requests that the client present its own certificate and rejects any connection in which the client does not do so. If the value is `disabled`, the server does not ask the client to present its own certificate.

This example sets the LDAPS connection handler's authorization policy to `required`.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  set-connection-handler-prop --handler-name "LDAPS Connection Handler" \
  --set ssl-client-auth-policy:required
```

#### 20.5.1.4 To Specify a Nickname for a Connection Handler's Certificate

Set the `ssl-cert-nickname` property of the connection handler.

The `ssl-cert-nickname` property specifies the nickname of the certificate that the server presents to clients during SSL or StartTLS negotiation. This property is primarily useful when multiple certificates are in the keystore and you want to specify which certificate is to be used for that listener instance.

This example sets the nickname of the LDAP connection handler's certificate to `server-cert`.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  set-connection-handler-prop --handler-name "LDAP Connection Handler" \
  --set ssl-cert-nickname:server-cert
```

#### 20.5.1.5 To Specify a Connection Handler's Key Manager Provider

Set the `key-manager-provider` property of the connection handler.

The `key-manager-provider` property specifies which key manager provider among the available Configuring Key Manager Providers that should be used by the connection handler to obtain the key material for the SSL or StartTLS negotiation.

This example sets the LDAP connection handler's key manager provider to `JKS`. The specified manager must already be configured for the command to succeed.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  set-connection-handler-prop --handler-name "LDAP Connection Handler" \
  --set key-manager-provider:JKS
```

#### 20.5.1.6 To Specify a Connection Handler's Trust Manager Provider

Set the `trust-manager-provider` property of the connection handler.

The `trust-manager-provider` property specifies which trust manager provider among the available Configuring Trust Manager Providers to be used by the connection handler to decide whether to trust client certificates presented to it.

This example sets the LDAP connection handler's trust manager to `JKS`. The specified manager must already be configured for the command to succeed.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  set-connection-handler-prop --handler-name "LDAP Connection Handler" \
  --set trust-manager-provider:JKS
```

#### 20.5.1.7 To Enable StartTLS Support

1. Specify the appropriate values for the `key-manager-provider` and `trust-manager-provider` properties.
2. Set the `allow-start-tls` property to `true`, as follows:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  set-connection-handler-prop --handler-name "LDAP Connection Handler" \
  --set allow-start-tls:true
```

> **Note:** If SSL is enabled, the `allow-start-tls` property cannot be set.
>
> StartTLS is not supported for connections between the proxy and the remote LDAP servers. Depending on the setting of the remote LDAP server SSL policy, StartTLS client connections can be passed from the proxy to the remote LDAP servers as SSL connections or as insecure connections. For more information, see To Create a Global Index Catalog Containing Global Indexes.

#### 20.5.1.8 To Enable SSL-Based Communication

1. Display the connection handler properties to ensure that the configured key manager provider and trust manager provider values are correct.

   The following example displays the properties of the LDAPS connection handler:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  get-connection-handler-prop --handler-name "LDAPS Connection Handler"
```

2. Set the `enabled` property to `true`, as follows:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  set-connection-handler-prop --handler-name "LDAPS Connection Handler" \
  --set enabled:true
```

> **Note:** If SSL is enabled, non-SSL communication will not be available for that connection handler instance.

### 20.5.2 Enabling SSL in the JMX Connection Handler

The JMX connection handler can be used to communicate with clients using the JMX (Java Management Extensions) protocol. This protocol does not support the use of StartTLS to allow both encrypted and unencrypted communication over the same port, but it can be configured to accept only unencrypted JMX or only SSL-encrypted JMX communication.

The JMX connection handler provides the server's default configuration for communicating over JMX. To enable SSL for this connection handler, use `dsconfig` to set the following configuration attributes:

- `key-manager-provider`. Specifies the DN of the configuration entry for the key manager provider that is used to obtain the key material for the SSL negotiation.
- `ssl-cert-nickname`. Specifies the nickname (or alias) of the certificate that is presented to clients.
- `use-ssl`. Indicates whether the connection handler is to use SSL to communicate with clients.

The following example uses `dsconfig` in interactive mode to configure the JMX connection handler:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X \
  set-connection-handler-prop --handler-name "JMX Connection Handler"
```

For a list of the configurable properties, see the "JMX Connection Handler Configuration" in the *Configuration Reference for Oracle Unified Directory*.

## 20.6 Using SASL Authentication

The LDAP protocol definition provides two ways in which clients can authenticate to the server: LDAP simple authentication and SASL authentication.

> **Note:** SASL is not supported for use with a proxy server instance.

In LDAP simple authentication, the client specifies the DN and password for the user. This is by far the most common authentication mechanism, and in most cases it is also the easiest to use. However, it has a number of limitations, including the following:

- The user is always required to provide a full DN, rather than something that could be more user-friendly like a user name.
- Only password-based authentication is allowed.
- The client must provide the complete clear-text password to the server.

To address these issues, it is also possible to authenticate clients through the Simple Authentication and Security Layer (SASL), as defined in RFC 4422 (http://www.ietf.org/rfc/rfc4422.txt). This is a very extensible framework, and makes it possible for servers to support many different kinds of authentication.

### 20.6.1 Supported SASL Mechanisms

currently supports the following SASL mechanisms:

> **Note:** With the proxy server, currently the only supported SASL mechanism is ANONYMOUS.

#### ANONYMOUS

This mechanism does not actually authenticate clients, but does provide a mechanism for including trace information in server logs for debugging purposes.

#### CRAM-MD5

This mechanism is provided for backward compatibility only. Do not configure CRAM-MD5 in a production environment. Use the DIGEST-MD5 mechanism instead, because it provides much better security.

#### DIGEST-MD5

This mechanism provides the ability for clients to use password-based authentication without sending the password to the server. Instead, the client only needs to provide information that proves it knows the password. This mechanism offers more options and better security than the CRAM-MD5 mechanism.

**EXTERNAL**
This mechanism provides the ability for clients to identify themselves based on information provided outside of the direct flow of LDAP communication. In Oracle Unified Directory, this may be achieved through the use of SSL client certificates.

**GSSAPI**
This mechanism provides the ability for clients to authenticate to the server through their participation in a Kerberos V5 environment.

**PLAIN**
This mechanism uses a password based authentication, but does offer the ability to use a username rather than requiring a DN.

Support for additional SASL mechanisms can be added by implementing custom SASL mechanism handlers in the server..

Because SASL mechanisms are so extensible, the set of information that the client needs to provide to the server in order to perform the authentication varies from one mechanism to another. As such, Oracle Unified Directory clients use a generic interface for users to provide this information. This is exposed through the `-o` or `--saslOption` argument, and the value for this argument should be a name-value pair. Select which SASL mechanism to use using the `mech` option, for example:

```
--saslOption mech=DIGEST-MD5
```

The other options that are available for use depend on the SASL mechanism that has been chosen, as described in the following sections.

## 20.6.2 Authorization IDs

Many of the SASL mechanisms below provide the ability to identify a user based on an authorization ID rather than a user DN. An authorization ID may be given in one of two forms:

`dn:dn`
This is used to provide the full DN of the user to authenticate (for example, `dn:uid=john.doe,ou=People,dc=example,dc=com`). A value of `dn:` with no DN is to be treated as the anonymous user, although this form is not accepted by many of the SASL mechanisms listed below.

`u:username`
This is used to provide the username of the user rather than the full DN (for example, `u:john.doe`).

If the `u:username` form is used, the mechanism that the server uses to resolve that username to the corresponding user entry is based on the identity mapping configuration within the server.

## 20.6.3 SASL Options for the ANONYMOUS Mechanism

Because the ANONYMOUS mechanism is not really used to perform authentication, no additional options are required. However, the following option can be supplied:

`trace`
This option can be used to provide a trace string that is written to the server's access log. This can be useful for debugging or to identify the client, although without authentication it is not possible to rely on the validity of this value.

The following command demonstrates the use of SASL anonymous authentication:

```
$ ldapsearch --hostname server.example.com --port 1389 --saslOption mech=ANONYMOUS
\
  --saslOption "trace=Example Trace String" --baseDN "" \
  --searchScope base "(objectClass=*)"
```

## 20.6.4 SASL Options for the CRAM-MD5 Mechanism

The CRAM-MD5 mechanism is used to perform password-based authentication to the server without exposing the clear-text password. It does this by providing an MD5 digest of the clear-text password combined with some randomly-generated data provided by the server, which helps prevent replay attacks.

The SASL CRAM-MD5 mechanism has one SASL option that must be provided:

`authid`
This specifies the identity of the user that is authenticating to the server. It should be an authorization ID value as described above.

The password is specified using either the `--bindPassword` or `--bindPasswordFile` option, just as when using simple authentication. The following command demonstrates the use of SASL CRAM-MD5 authentication:

```
ldapsearch --hostname server.example.com --port 1389 --saslOption mech=CRAM-MD5 \
--saslOption authid=u:john.doe --baseDN "" --searchScope base "(objectClass=*)"
```

## 20.6.5 SASL Options for the DIGEST-MD5 Mechanism

The DIGEST-MD5 mechanism is similar to the CRAM-MD5 mechanism, but it is more secure because it combines random data from both the client and the server in order to help foil both replay and man-in-the-middle attacks. DIGEST-MD5 authentication also offers a number of SASL options, including the following:

`authid`
Specifies the identity of the user that is authenticating to the server. This option must be provided.

`realm`
This option should not be specified as a DN.

> **Note:** Do not use the `realm` option, because the server does not use it when mapping identities.

`digest-uri`
Specifies the digest URI that the client uses to communicate with the server. This is an optional parameter, but if it is provided, specify it in the form `ldap/serveraddress`, where *serveraddress* is the fully-qualified address of the server.

> **Note:** Do not use the `digest-uri` option in a production environment.

`authzid`
Specifies the authorization ID that should be used during the authentication process. This option can be used to indicate that the operations requested on the connection after authentication should be performed under the authority of another user.

The password is specified using either the `--bindPassword` or `--bindPasswordFile` option, just as when using simple authentication. The following command demonstrates the use of SASL DIGEST-MD5 authentication:

```
$ ldapsearch --hostname server.example.com --port 1389 --saslOption
mech=DIGEST-MD5 \
  --saslOption authid=u:john.doe --saslOption realm=dc=example,dc=com --baseDN ""
\
  --searchScope base "(objectClass=*)"
```

## 20.6.6 SASL Options for the EXTERNAL Mechanism

The EXTERNAL mechanism is used to perform authentication based on information that is available to the server outside of the LDAP session. At present, this is available only through SSL client authentication, in which case the information that the client's SSL certificate will be used to authenticate that client. As such, it is necessary to use SSL or StartTLS when communicating with the server, and a client certificate keystore must be available.

The EXTERNAL mechanism does not support any additional SASL options. In most cases, it can be requested using either `--saslOption mech=EXTERNAL` or `--useSASLExternal`. The following command demonstrates the use of SASL EXTERNAL authentication:

```
$ ldapsearch --hostname server.example.com --port 1636 --useSSL \
  --keyStorePath /path/to/key.store --keyStorePasswordFile /path/to/key.store.pin
\
  --trustStorePath /path/to/trust.store --saslOption mech=EXTERNAL --baseDN "" \
  --searchScope base "(objectClass=*)"
```

For more information, see Configuring SASL External Authentication.

## 20.6.7 SASL Options for the GSSAPI Mechanism

The GSSAPI mechanism is used to perform authentication in a Kerberos V5 environment, and generally requires that the client system be configured to participate in such an environment. The options available for use with the GSSAPI mechanism include:

`authid`
Specifies the authentication ID that should be used to identify the user. This ID should be in the form of a Kerberos principal and not in the authorization ID form described previously. This option must be provided if the user has not authenticated to Kerberos before attempting to bind.

`authzid`
Specifies the authorization ID that should be used to identify the user under whose authority operations should be performed. does not yet support this capability.

`quality-of-protection`
Specifies the quality of protection to use for the communication. Currently, only the `auth` quality-of-protection value is supported by clients. The `auth-int` and `auth-conf` values are supported by the server.

If the user already has a valid Kerberos ticket on the system when attempting to use GSSAPI, the client attempts to use it so that no password is required. However, if the user does not have a valid Kerberos ticket or if it cannot be accessed for some reason, a

password must be provided using either the `--bindPassword` or `--bindPasswordFile` options.

The following command demonstrates the use of SASL GSSAPI authentication for a user that already has a valid Kerberos session:

```
$ ldapsearch --hostname server.example.com --port 1389 --saslOption mech=GSSAPI \
--saslOption authid=jdoe@EXAMPLE.COM --baseDN "" --searchScope base
"(objectClass=*)"
```

### 20.6.8 SASL Options for the PLAIN Mechanism

The PLAIN mechanism provides many of the same capabilities as LDAP simple authentication, although the user may be identified in the form of an authorization ID rather than requiring a full DN. The following options are available for use when using SASL PLAIN authentication:

`authid`
Specifies the identity of the user that is authenticating to the server. It should be an authorization ID value as described above. This option must be provided.

`authzid`
Specifies the identity of the user under whose authority operations should be performed. It should also be in the form of an authorization ID. does not yet support this capability.

The password is specified using either the `--bindPassword` or `--bindPasswordFile` option, just as when using simple authentication. The following command demonstrates the use of SASL PLAIN authentication:

```
$ ldapsearch --hostname server.example.com --port 1389 --saslOption mech=PLAIN \
--saslOption authid=u:john.doe --baseDN "" --searchScope base "(objectClass=*)"
```

## 20.7 Configuring SASL Authentication

This section describes the requirements for configuring directory server to use the various SASL authentication mechanisms.

**Note:** SASL is not supported for use with a proxy server instance.

### 20.7.1 Configuring SASL External Authentication

The SASL EXTERNAL mechanism is used to allow a client to authenticate itself to the directory server using information provided outside of what is strictly considered LDAP communication. currently supports authentication using a client certificate presented to the server during SSL or StartTLS negotiation, for LDAP communication only.

#### 20.7.1.1 Configuring the LDAP Connection Handler to Allow SASL EXTERNAL Authentication

For the directory server to be able to map the client certificate to a user entry, ensure that the connection handler is configured to handle client certificates. Use the `dsconfig` to set the following LDAP connection handler properties:

- **ssl-client-auth-policy.** Specifies whether the directory server prompts the client to present its own certificate during the SSL or StartTLS negotiation process. To

support SASL EXTERNAL authentication, the value must be either `optional` or `required`. If the value is `disabled`, clients are not prompted to provide a certificate and no certificate is available for authentication.

- **trust-manager-provider.** Specifies the DN of the trust manager provider used to determine whether the directory server trusts the validity of the client certificate. If the server does not trust the client certificate, the SSL or StartTLS negotiation fails and it is not possible for the client to request SASL EXTERNAL authentication. If the server trusts illegitimate client certificates, it is possible for malicious users to forge certificates and impersonate any user in the directory. In most cases, the JKS or PKCS12 trust manager provider should be used and the corresponding trust store loaded only with the issuer certificates that are used to sign client certificates.

> **Note:** The `dsconfig` command accesses the server configuration over SSL via the administration connector. As such, the relevant connection options must be specified, including how the SSL certificate is trusted. These examples use the `-X` option to trust all certificates.

The following example uses `dsconfig` in interactive mode to set LDAP connection handler properties:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager"-j pwd-file -X \
  set-connection-handler-prop --handler-name "LDAP Connection Handler"
```

For a list of the configurable properties, see the "LDAP Connection Handler Configuration" in the *Configuration Reference for Oracle Unified Directory*.

#### 20.7.1.2 Configuring the EXTERNAL SASL Mechanism Handler

SASL EXTERNAL bind requests are processed by the SASL mechanism handler. Use the `dsconfig` command to set the following SASL mechanism handler properties:

- **java-class.** Specifies the fully-qualified name of the Java class that provides the logic for the SASL mechanism handler. For the EXTERNAL mechanism, this value is always `org.opends.server.extensions.ExternalSASLMechanismHandler`. An advanced property.
- **enabled.** Indicates whether the EXTERNAL SASL mechanism is enabled for use. If you do not want to allow clients to use SASL EXTERNAL authentication, change its value to `false`.
- **certificate-mapper.** Specifies the DN of the configuration entry for the certificate mapper to be used to map client certificates to user entries.
- **certificate-validation-policy.** Specifies whether the directory server attempts to locate the client certificate in the user's entry after establishing a mapping. If the value is `always`, the authentication succeeds only if the mapped user's entry contains the certificate presented by the client. If the value is `ifpresent` (the default value) and the user's entry contains one or more certificates, the authentication succeeds only if one of those certificates matches the one presented by the client. If the value is `ifpresent` and the user's entry does not contain any certificates, the authentication still succeeds based on the fact that it would have been accepted by the trust manager and mapped by the certificate mapper. If the value is `never`, the server does not attempt to match the certificate to a value in the user's entry even if that entry contains one or more certificates.

- **certificate-attribute.** Specifies the name of the attribute that holds user certificates to be examined if the `certificate-validation-policy` property has a value of either `always` or `ifpresent`.

The following example uses `dsconfig` in interactive mode to set EXTERNAL SASL mechanism handler properties:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager"-j pwd-file -X \
  set-sasl-mechanism-handler-prop --handler-name "EXTERNAL"
```

For a list of the configurable properties, see the "SASL Mechanism Handler Configuration" in the *Configuration Reference for Oracle Unified Directory*.

## 20.7.2 Configuring SASL DIGEST-MD5 Authentication

This section explains the access control and privilege restrictions on a user using the authorization ID keyword (`authzid`). If the user is not using the `authzid` keyword, these restrictions do not apply. Any user that binds using DIGEST-MD5 and the `authzid` keyword must fulfill the following requirements:

- The authentication ID (`authid`) must be granted access by an ACI that grants it the proxy right to the authorization ID.
- The authentication ID (`authid`) entry must contain the `proxied-auth` privilege. The following example creates a test environment and demonstrates the requirements for user authentication using the DIGEST-MD5 SASL mechanism.

The following example creates a test environment and then demonstrates the requirements for a user authentication using the DIGEST-MD5 SASL mechanism.

1. Import the following entries into the directory. These entries define an ACI and three users:
   - The entry `uid=user.0,ou=People,dc=example,dc=com` does not have the `proxied-auth` privilege but is granted proxy access by the ACI.
   - The entry `uid=user.1,ou=People,dc=example,dc=com` has the `proxied-auth` privilege but is not granted proxy access by the ACI.
   - The entry `uid=user.2,ou=People,dc=example,dc=com` has the `proxied-auth` privilege and is granted proxy access by the ACI.

```
dn: ou=People,dc=example,dc=com
objectClass: top
objectClass: organizationalunit
objectClass: posixGroup
ou: People
aci: (target="ldap:///uid=proxy user,ou=People,dc=example,dc=com") \
 (targetattr="*") (version 3.0; acl "allow SASL Example"; \
 allow (proxy) userdn="ldap:///uid=user.0,ou=People,dc=example,dc=com ||
 ldap:///uid=user.2,ou=People,dc=example,dc=com";)

dn: uid=user.0,ou=People,dc=example,dc=com
objectClass: top
objectClass: person
objectClass: organizationalperson
objectClass: inetorgperson
...
description: This is the description for user.0

dn: uid=user.1,ou=People,dc=example,dc=com
objectClass: top
```

```
objectClass: person
objectClass: organizationalperson
objectClass: inetorgperson
...
description: This is the description for user.1
ds-privilege-name: proxied-auth

dn: uid=proxy user,ou=People,dc=example,dc=com
objectClass: top
objectClass: person
objectClass: organizationalperson
objectClass: inetorgperson
...
description: This is the description for proxy user

dn: uid=user.2,ou=People,dc=example,dc=com
objectClass: top
objectClass: person
objectClass: organizationalperson
objectClass: inetorgperson
...
description: This is the description for user.2
ds-privilege-name: proxied-auth
```

2. Bind using DIGEST-MD5 as `uid=user.1,ou=People,dc=example,dc=com`:

```
$ ldapsearch --port 1389 -j pwd-file --saslOption mech=DIGEST-MD5 \
  --saslOption authid=dn:uid=user.1,ou=People,dc=example,dc=com --saslOption \
  authzid=dn:uid=proxy user,ou=People,dc=example,dc=com --baseDN "" \
  --searchScope base "(objectClass=*)"
 The SASL DIGEST-MD5 bind attempt failed Result Code: 49 (Invalid Credentials)
```

   The search fails because `uid=user.1,ou=People,dc=example,dc=com` is not granted the proxy right by the ACI.

3. Bind using DIGEST-MD5 as `uid=user.0,ou=People,dc=example,dc=com`:

```
$ ldapsearch --port 1389 -j pwd-file --saslOption mech=DIGEST-MD5 \
  --saslOption authid=dn:uid=user.0,ou=People,dc=example,dc=com --saslOption \
  authzid=dn:uid=proxy user,ou=People,dc=example,dc=com --baseDN "" \
  --searchScope base "(objectClass=*)"
The SASL DIGEST-MD5 bind attempt failed Result Code: 49 (Invalid Credentials)
```

   The search fails because `uid=user.0,ou=People,dc=example,dc=com` does not have the `proxied-auth` property.

4. Bind using DIGEST-MD5 as `uid=user.2,ou=People,dc=example,dc=com` authid with both access control access and the `proxied-auth` privilege:

```
$ ldapsearch --port 1389 -j pwd-file --saslOption mech=DIGEST-MD5 \
  --saslOption authid=dn:uid=user.2,ou=People,dc=example,dc=com --saslOption \
  authzid=dn:uid=proxy user,ou=People,dc=example,dc=com --baseDN "" \
  --searchScope base "(objectClass=*)"
dn:
objectClass: ds-root-dse
objectClass: top
```

   The search succeeds because `uid=user.2,ou=People,dc=example,dc=com` has access allowed by the ACI and the `proxied-auth` privilege.

### 20.7.3 Configuring SASL GSSAPI Authentication

This section explains the access control and privilege restrictions on a user using the authorization ID keyword (`authzid`). If the user is not using the `authzid` keyword, the restrictions do not apply.

Any user that binds using GSSAPI must fulfill the following requirements:

- The authentication ID (`authid`) must be granted access by an ACI that grants it the proxy right to the authorization ID.
- The authentication ID (`authid`) entry must contain the `proxied-auth` privilege.

The following example creates a test environment with three example entries and demonstrates the requirements for user authentication using the GSSAPI SASL mechanism. These examples require a fully configured Kerberos environment, including a valid keytab file.

1. Create three Kerberos principals in the realm `TESTLOCAL.NET`:
   - `user.0@TESTLOCAL.NET`
   - `user.1@TESTLOCAL.NET`
   - `user.2@TESTLOCAL.NET`
2. Configure the GSSAPI SASL handler to be enabled, to use the regular expression identity mapper, and to use a valid TESTLOCAL.NET keytab file.

   ```
   $ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
     set-sasl-mechanism-handler-prop --handler-name "GSSAPI" \
     --set enabled:true --set identity-mapper:"Regular Expression" \
     --set keytab:keytabPath
   ```

   The default value of the GSSAPI `enabled` property is `false`, so it must be set to `true`. The default value of `identity-mapper` is Regular Expression. The default value of the `keytab` property is `/etc/krb5/krb5.keytab`.
3. Import the following entries into the directory. These entries define an ACI and three users:
   - The entry `uid=user.0,ou=People,dc=example,dc=com` does not have the `proxied-auth` privilege but is granted proxy access by the ACI.
   - The entry `uid=user.1,ou=People,dc=example,dc=com` has the `proxied-auth` privilege but is not granted proxy access by the ACI.
   - The entry `uid=user.2,ou=People,dc=example,dc=com` has the `proxied-auth` privilege and is granted proxy access by the ACI.

   ```
   dn: ou=People,dc=example,dc=com
   objectClass: top
   objectClass: organizationalunit
   objectClass: posixGroup
   ou: People
   aci: (target="ldap:///uid=proxy user,ou=People,dc=example,dc=com") \
    (targetattr="*") (version 3.0; acl "allow SASL Example"; \
    allow (proxy) userdn="ldap:///uid=user.0,ou=People,dc=example,dc=com"
    || "ldap:///uid=user.2,ou=People,dc=example,dc=com";)

   dn: uid=user.0,ou=People,dc=example,dc=com
   objectClass: top
   objectClass: person
   objectClass: organizationalperson
   objectClass: inetorgperson
   ```

```
uid=user.0
...
description: This is the description for user.0

dn: uid=user.1,ou=People,dc=example,dc=com
objectClass: top
objectClass: person
objectClass: organizationalperson
objectClass: inetorgperson
uid=user.1
...
description: This is the description for user.1
ds-privilege-name: proxied-auth

dn: uid=user.2,ou=People,dc=example,dc=com
objectClass: top
objectClass: person
objectClass: organizationalperson
objectClass: inetorgperson
uid=user.2
...
description: This is the description for user.2
ds-privilege-name: proxied-auth

dn: uid=proxy user,ou=People,dc=example,dc=com
objectClass: top
objectClass: person
objectClass: organizationalperson
objectClass: inetorgperson
uid=proxy user
...
description: This is the description for proxy user
```

4. Run this command to demonstrate a failing GSSAPI SASL bind using the Kerberos principal, `user.0@TESTLOCAL.NET`:

```
$ ldapsearch --port 1389 \
  --saslOption mech=GSSAPI \
  --saslOption authid=user.0@TESTLOCAL.NET \
  --saslOption authzid=dn:uid=proxy user,ou=People,dc=example,dc=com \
  --baseDN "" --searchScope base "(objectClass=*)"
The SASL DIGEST-MD5 bind attempt failed
Result Code:  49 (Invalid Credentials)
```

   This search fails because `user.0@TESTLOCAL.NET` maps to `uid=user.0,ou=People,dc=example,dc=com`, which has access control permissions to `uid=proxy user,ou=People,dc=example,dc=com` but does not have the `proxied-auth` privilege.

5. Run this command to demonstrate a failing GSSAPI SASL bind using the Kerberos principal, `user.1@TESTLOCAL.NET`.

```
$ ldapsearch --port 1389 \
  --saslOption mech=GSSAPI \
  --saslOption authid=user.1@TESTLOCAL.NET \
  --saslOption authzid=dn:uid=proxy user,ou=People,dc=example,dc=com \
  --baseDN "" --searchScope base "(objectClass=*)"
The SASL DIGEST-MD5 bind attempt failed
Result Code:  49 (Invalid Credentials)
```

This search fails because `user.1@TESTLOCAL.NET` maps to `uid=user.1,ou=People,dc=example,dc=com`, which has the `proxied-auth` privilege but does not have access control permissions to `uid=proxy user,ou=People,dc=example,dc=com`.

6. Run this command to demonstrate a successful GSSAPI SASL bind using the Kerberos principal `user.2@TESTLOCAL.NET`:

```
$ ldapsearch --port 1389 \
  --saslOption mech=GSSAPI \
  --saslOption authid=user.2@TESTLOCAL.NET \
  --saslOption authzid=dn:uid=proxy user,ou=People,dc=example,dc=com \
  --baseDN "" --searchScope base "(objectClass=*)"
dn:
objectClass: ds-root-dse
objectClass: top }}} \\ \\
```

This search succeeds because `user.2@TESTLOCAL.NET` maps to `uid=user.2,ou=People,dc=example,dc=com`, which has both the `proxied-auth` privilege and access control permission to `id=proxy user,ou=People,dc=example,dc=com`.

## 20.8 Configuring Kerberos and the Oracle Unified Directory Server for GSSAPI SASL Authentication

The following sections describe how to configure and Kerberos Version 5 for GSSAPI SASL authentication.

- To Configure Kerberos V5 on a Host
- To Specify SASL Options for Kerberos Authentication
- Example Configuration of Kerberos Authentication Using GSSAPI With SASL
- Troubleshooting Kerberos Configuration

### 20.8.1 To Configure Kerberos V5 on a Host

You must configure Kerberos V5 on the host machine where your LDAP clients will run.

1. Install Kerberos V5 according to its installation instructions.

   Sun recommends installing the Sun Enterprise Authentication Mechanism 1.0.1 client software.

2. Configure the Kerberos software.

   Using the Sun Enterprise Authentication Mechanism software, configure the files under `/etc/krb5`. This configuration sets up the `kdc` server, and defines the default realm and any other configuration required by your Kerberos system.

3. If necessary, modify the file `/etc/gss/mech` so that the first value that is listed is `kerberos_v5`.

### 20.8.2 To Specify SASL Options for Kerberos Authentication

You must specify appropriate SASL options for the Kerberos installation.

1. Before using a client application that is enabled with the GSSAPI mechanism, initialize the Kerberos security system with your user Principal.

```
$ kinit user-principal
```

where the *user-principal* is your SASL identity, for example, `bjensen@example.com`.

2. Specify SASL options for using Kerberos.

   Note that in the UNIX environment, you must set the `SASL_PATH` environment variable to the correct path for the SASL libraries. For example in the Korn shell:

   ```
   $ export SASL_PATH=SASL-library
   ```

   This path assumes that the Oracle Unified Directory software is installed on the same host where the LDAP tools are invoked.

   The following example of the `ldapsearch` tool shows the use of the `-o` (lowercase letter o) option to specify SASL options for using Kerberos:

   ```
   $ ldapsearch -h www.host1.com -p 1389 -o mech=GSSAPI -o
   authid="bjensen@EXAMPLE.COM" \
    -o authzid="bjensen@EXAMPLE.COM" -b "dc=example,dc=com" "(givenname=Richard)"
   ```

   The `authid` can be omitted because it is present in the Kerberos cache that was initialized by the `kinit` command. If `authid` is present, `authid` and `authzid` must be identical, although the `authzid` intended for proxy operations is not used. The value of `authid` is the Principal that is used in identity mapping. The Principal must be the full Principal, including the realm.

### 20.8.3 Example Configuration of Kerberos Authentication Using GSSAPI With SASL

Configuring Kerberos for the Oracle Unified Directory directory server can be complicated. Your first point of reference should be the Kerberos documentation.

For more help, use the following example procedure to get an idea of which steps to follow. Be aware, however, that this procedure is an example. You must modify the procedure to suit your own configuration and your own environment.

Additional information about configuring and using Kerberos in the Solaris OS can be found in *System Administration Guide: Security Services*. This guide is a part of the Solaris documentation set. You can also consult the man pages.

Information about this example and the steps used are as follows:

1. Assumptions for This Example
2. All Machines: Edit the Kerberos Client Configuration File
3. All Machines: Edit the Administration Server ACL Configuration File
4. KDC Machine: Edit the KDC Server Configuration File
5. KDC Machine: Create the KDC Database
6. KDC Machine: Create an Administration Principal and Keytab
7. KDC Machine: Start the Kerberos Daemons
8. KDC Machine: Add Host Principals for the KDC and Oracle Unified Directory Machines
9. KDC Machine: Add an LDAP Principal for the Directory Server
10. KDC Machine: Add a Test User to the KDC
11. Directory Server Machine: Install Oracle Unified Directory

12. Directory Server Machine: Configure the Directory Server to Enable GSSAPI
13. Directory Server Machine: Create and Configure the Directory Server LDAP
14. Directory Server Machine: Add a Test User to the Directory Server
15. Directory Server Machine: Obtain a Kerberos Ticket as the Test User
16. Client Machine: Authenticate to the Directory Server Through GSSAPI

#### 20.8.3.1 Assumptions for This Example

This example procedure describes the process of configuring one machine to operate as a Key Distribution Center (KDC), and a second machine to run the directory server. The result of this procedure is that users can perform Kerberos authentication through GSSAPI.

It is possible to run both the KDC and the directory server on the same machine. If you choose to run both on the same machine, use the same procedure, but omit the steps for the directory server machine that have already been done for the KDC machine.

This procedure makes a number of assumptions about the environment that is used. When using the example procedure, modify the values accordingly to suit your environment. These assumptions are:

- This system has a fresh installation of the Solaris 10 software with the latest recommended patch cluster installed. Kerberos authentication to the directory server can fail if the appropriate Solaris patches are not installed.
- The machine that is running the Kerberos daemons has the fully qualified domain name of `kdc.example.com`. The machine must be configured to use DNS as a naming service. This configuration is a requirement of Kerberos. Certain operations might fail if other naming services such as `file` are used instead.
- The machine that is running the directory server has the fully qualified domain name of `directory.example.com`. This machine must also be configured to use DNS as a naming service.
- The directory server machine serves as the client system for authenticating to the directory server through Kerberos. This authentication can be performed from any system that can communicate with both the directory server and Kerberos daemons. However, all of the necessary components for this example are provided with the Oracle Unified Directory directory server, and the authentication is performed from that system.
- Users in the directory server have DNs of the form `uid=`*username*`,ou=People,dc=example,dc=com`. The corresponding Kerberos principal is *username*`@EXAMPLE.COM`. If a different naming scheme is used, a different GSSAPI identity mapping must be used.

#### 20.8.3.2 All Machines: Edit the Kerberos Client Configuration File

The `/etc/krb5/krb5.conf` configuration file provides information that Kerberos clients require in order to communicate with the KDC.

Edit the `/etc/krb5/krb5.conf` configuration file on the KDC machine, the directory server machine, and any client machines that will authenticate to the directory server using Kerberos.

- Replace every occurrence of `"___default_realm___"` with `"EXAMPLE.COM"`.
- Replace every occurrence of `"___master_kdc___"` with `"kdc.example.com"`.

- Remove the lines that contain "___slave_kdcs___" as there will be only a single Kerberos server.
- Replace "___domain_mapping___" with ".example.com = EXAMPLE.COM" (note the initial period in.example.com).

The updated /etc/krb5/krb5.conf configuration file should look like the contents of the following example.

***Example 20–1 Edited Kerberos Client Configuration File*** /etc/krb5/krb5.conf

```
#pragma ident   "@(#)krb5.conf  1.2     99/07/20 SMI"
# Copyright (c) 1999, by Sun Microsystems, Inc.
# All rights reserved.
#
# krb5.conf template
# In order to complete this configuration file
# you will need to replace the __<name\>__ placeholders
# with appropriate values for your network.
#

[libdefaults]
        default_realm = EXAMPLE.COM
[realms]
        EXAMPLE.COM = {
                kdc = kdc.example.com
                admin_server = kdc.example.com
        }
[domain_realm]
        .example.com = EXAMPLE.COM
[logging]
        default = FILE:/var/krb5/kdc.log
        kdc = FILE:/var/krb5/kdc.log
        kdc_rotate = {

# How often to rotate kdc.log. Logs will get rotated no more
# often than the period, and less often if the KDC is not used
# frequently.
                period = 1d

# how many versions of kdc.log to keep around (kdc.log.0, kdc.log.1, ...)
                versions = 10
        }

[appdefaults]
        kinit = {
                renewable = true
                forwardable= true
        }
        gkadmin = {
                help_url =
 http://docs.sun.com:80/ab2/coll.384.1/SEAM/@AB2PageView/1195
        }
```

### 20.8.3.3 All Machines: Edit the Administration Server ACL Configuration File

Replace "___default_realm___" with "EXAMPLE.COM" in the /etc/krb5/kadm5.acl configuration file. The updated file should look like the following example.

***Example 20–2 Edited Administration Server ACL Configuration File***

```
#
# Copyright (c) 1998-2000 by Sun Microsystems, Inc.
# All rights reserved.
#
# pragma ident   "@(#)kadm5.acl  1.1     01/03/19 SMI"
*/admin@EXAMPLE.COM *
```

### 20.8.3.4 KDC Machine: Edit the KDC Server Configuration File

Edit the `/etc/krb5/kdc.conf` file to replace `"___default_realm___"` with `"EXAMPLE.COM"`. The updated file should look like the following example.

***Example 20–3 Edited KDC Server Configuration File*** `/etc/krb5/kdc.conf`

```
# Copyright 1998-2002 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
#ident  "@(#)kdc.conf   1.2     02/02/14 SMI"

[kdcdefaults]
        kdc_ports = 88,750

[realms]
        EXAMPLE.COM = {
                profile = /etc/krb5/krb5.conf
                database_name = /var/krb5/principal
                admin_keytab = /etc/krb5/kadm5.keytab
                acl_file = /etc/krb5/kadm5.acl
                kadmind_port = 749
                max_life = 8h 0m 0s
                max_renewable_life = 7d 0h 0m 0s
                default_principal_flags = +preauth
        }
```

### 20.8.3.5 KDC Machine: Create the KDC Database

```
$ /usr/sbin/kdb5_util create -r EXAMPLE.COM -s
Initializing database '/var/krb5/principal' for realm 'EXAMPLE.COM',
master key name 'K/M@EXAMPLE.COM'
You will be prompted for the database Master Password.
It is important that you NOT FORGET this password.
Enter KDC database master key: password
Re-enter KDC database master key to verify: password
$
```

### 20.8.3.6 KDC Machine: Create an Administration Principal and Keytab

Use the following command to create an administration user with a Principal of `kws/admin@EXAMPLE.COM` and service keys that will be used by the administration daemon.

```
$ /usr/sbin/kadmin.local
kadmin.local:  add_principal kws/admin
Enter password for principal "kws/admin@EXAMPLE.COM": secret
Re-enter password for principal "kws/admin@EXAMPLE.COM": secret
Principal "kws/admin@EXAMPLE.COM" created.
kadmin.local:  ktadd -k /etc/krb5/kadm5.keytab kadmin/kdc.example.com
Entry for principal kadmin/kdc.example.com with kvno 3, encryption type
```

```
 DES-CBC-CRC added to keytab WRFILE:/etc/krb5/kadm5.keytab.
kadmin.local:  ktadd -k /etc/krb5/kadm5.keytab changepw/kdc.example.com

Entry for principal changepw/kdc.example.com with kvno 3, encryption type
 DES-CBC-CRC added to keytab WRFILE:/etc/krb5/kadm5.keytab.
kadmin.local:  ktadd -k /etc/krb5/kadm5.keytab kadmin/changepw
Entry for principal kadmin/changepw with kvno 3, encryption type
 DES-CBC-CRC added to keytab WRFILE:/etc/krb5/kadm5.keytab.
kadmin.local:  quit$
```

#### 20.8.3.7 KDC Machine: Start the Kerberos Daemons

The Kerberos daemons are managed by the Service Management Facility (SMF) framework. Run the following commands to start the KDC and administration daemons:

```
$ /etc/init.d/kdc start
$ /etc/init.d/kdc.master start
$

$ svcadm disable network/security/krb5kdc
$ svcadm enable network/security/krb5kdc
$ svcadm disable network/security/kadmin
$ svcadm enable network/security/kadmin
$
```

The KDC process appears in the process list as `/usr/lib/krb5/krb5kdc`. The administration daemon appears as `/usr/lib/krb5/kadmind`.

#### 20.8.3.8 KDC Machine: Add Host Principals for the KDC and Oracle Unified Directory Machines

Use the following sequence of commands to add host Principals to the Kerberos database for the KDC and the directory server machines. The host Principal is used by certain Kerberos utilities such as `klist`.

```
$ /usr/sbin/kadmin -p kws/admin
Enter Password: secret
kadmin:  add_principal -randkey host/kdc.example.com
Principal "host/kdc.example.com@EXAMPLE.COM" created.
kadmin:  ktadd host/kdc.example.com
Entry for principal host/kdc.example.com with kvno 3, encryption type
 DES-CBC-CRC added to keytab WRFILE:/etc/krb5/krb5.keytab.
kadmin:  add_principal -randkey host/directory.example.com
Principal "host/directory.example.com@EXAMPLE.COM" created.
kadmin:  ktadd host/directory.example.com
Entry for principal host/directory.example.com with kvno 3, encryption type
 DES-CBC-CRC added to keytab WRFILE:/etc/krb5/krb5.keytab.
kadmin:  quit
$
```

#### 20.8.3.9 KDC Machine: Add an LDAP Principal for the Directory Server

For the directory server to be able to validate the Kerberos tickets that are held by authenticating users, the directory server must have its own Principal. Currently Oracle Unified Directory is hard coded to require a Principal of `ldap/fqdn@realm` where *fqdn* is the fully-qualified domain name of the directory server and *realm* is the Kerberos realm. The *fqdn* must match the fully qualified name that is provided when you install Oracle Unified Directory. In this case, the Principal for the directory server would be `ldap/directory.example.com@EXAMPLE.COM`.

Use the following sequence of commands to create an LDAP Principal for the directory server:

```
$ /usr/sbin/kadmin -p kws/admin
Enter Password: secret
kadmin: add_principal -randkey ldap/directory.example.com
Principal "ldap/directory.example.com@EXAMPLE.COM" created.
kadmin: quit
$
```

### 20.8.3.10 KDC Machine: Add a Test User to the KDC

To perform Kerberos authentication, the user authenticating must exist in the Kerberos database. In this example, the user has the user name `kerberos-test`, which means that the Kerberos Principal is `kerberos-test@EXAMPLE.COM`.

Create the user by using the command sequence in this example:

```
$ /usr/sbin/kadmin -p kws/admin
Enter Password: secret
kadmin:  add_principal kerberos-test
Enter password for principal "kerberos-test@EXAMPLE.COM": secret

Re-enter password for principal "kerberos-test@EXAMPLE.COM": secret

Principal "kerberos-test@EXAMPLE.COM" created.
kadmin:  quit
$
```

### 20.8.3.11 Directory Server Machine: Install Oracle Unified Directory

Install Oracle Unified Directory. The following table lists the installation settings that this section uses in examples.

| Variable Type | Example Value |
|---|---|
| Fully qualified directory server DNS name | `directory.example.com` |
| Server port | `389` |
| Suffix | `dc=example,dc=com` |
| Installation directory | `/asinst_1/oud` |
| Oracle Unified Directory server user | `oud` |
| Oracle Unified Directory server group | `oud` |
| Kerberos test principal | `kerberos-test` |
| Oracle Unified Directory keytab path | `/asinst_1/oud/config/oud.keytab` |

**Note:** The fully qualified directory server DNS name must resolve to the same IP address on all of the servers (the Oracle Unified Directory servers and the Kerberos Key Distribution Center (KDC) and client machines that expect to bind to the server using GSSAPI SASL).

### 20.8.3.12 Directory Server Machine: Create and Configure the Directory Server LDAP

As mentioned previously, to authenticate Kerberos users through GSSAPI, Oracle Unified Directory must have its own Principal in the KDC. The Principal information must reside in a Kerberos keytab on the directory server machine. This information must be in a file that is readable by the user account under which the directory server operates.

> **Note:** This step must be performed before the `GSSAPI SASL mechanism handler` is configured. The handler checks to make sure the keytab file exists before it will initialize.

Create a keytab file with the correct properties by using the following command sequence:

```
$ kadmin -p kws/admin@EXAMPLE.COM
kadmin:  addprinc -randkey ldap/directory.example.com
WARNING: no policy specified for ldap/directory.example.com@EXAMPLE.COM;
  defaulting to no policy
Principal "ldap/directory.example.com@EXAMPLE.COM" created.
kadmin:  ktadd -k asinst_1/oud/config/oud.keytab ldap/directory.example.com
Entry for principal ldap/directory.example.com with kvno 3,
  encryption type AES-128 CTS mode
  with 96-bit SHA-1 HMAC added to keytab WRFILE:asinst_1/oud/config/oud.keytab.
Entry for principal ldap/directory.example.com with kvno 3,
  encryption type Triple DES cbc mode
  with HMAC/sha1 added to keytab WRFILE:asinst_1/oud/config/oud.keytab.
Entry for principal ldap/directory.example.com with kvno 3,
  encryption type ArcFour with HMAC/md5
  added to keytab WRFILE:asinst_1/oud/config/oud.keytab.
Entry for principal ldap/directory.example.com with kvno 3,
  encryption type DES cbc mode with RSA-MD5
  added to keytab WRFILE:asinst_1/oud/config/oud.keytab.
kadmin:  quit
```

Change the permissions and ownership on this custom keytab. Make the keytab owned by the user account used to run the directory server and readable only by that user:

```
$ chown oud:oud asinst_1/oud/config/oud.keytab
$ chmod 600 asinst_1/oud/config/oud.keytab
```

To allow these changes to take effect, stop and restart the directory server.

### 20.8.3.13 Directory Server Machine: Configure the Directory Server to Enable GSSAPI

This step shows examples of managing the `GSSAPI SASL mechanism handler` on the directory server host `directory.example.com`.

Use the `dsconfig` command as shown in the following example to enable the GSSAPI SASL mechanism handler on the directory server host `directory.example.com` and configure it to use the `asinst_1/oud/config/oud.keytab`.

```
$ dsconfig -X -n -p 4444 -h directory.example.com \
  -D "cn=directory manager" -j pwd-file
  set-sasl-mechanism-handler-prop \
  --handler-name GSSAPI \
```

```
  --set enabled:true \
  --set keytab:asinst_1/oud/config/oud.keytab \
  --set server-fqdn:directory.example.com
```

The last line in this command sets the GSSAPI SASL mechanism property `server-fqdn` to `directory.example.com`. This is an optional parameter, which can be left out only if it is assured that a hostname lookup on the directory server host returns the exact hostname that was used in creating the LDAP principal. Setting this property explicitly assures that the two names are the same (in this example, `directory.example.com`).

Confirm that the configuration is correct by examining the properties of the GSSAPI SASL mechanism handler on the directory server host `directory.example.com`.

```
$ dsconfig -X -n -p 4444 -h directory.example.com \
  -D "cn=directory manager" -j pwd-file \
  get-sasl-mechanism-handler-prop \
  --handler-name GSSAPI
Property              : Value(s)
----------------------:----------------------
enabled               : true
identity-mapper       : Regular Expression
kdc-address           : -
keytab                : asinst_1/oud/config/oud.keytab
principal-name        : -
quality-of-protection : none
realm                 : -
server-fqdn           : directory.example.com
```

If necessary for troubleshooting, you can use `dsconfig` to list the status of all the SASL mechanism handlers on the directory server host `directory.example.com`.

```
$ dsconfig -X -n -p 4444 -h directory.example.com \
  -D "cn=directory manager" -j pwd-file \
  list-sasl-mechanism-handlers
SASL Mechanism Handler : Type       : enabled
-----------------------:------------:--------
ANONYMOUS              : anonymous  : false
CRAM-MD5               : cram-md5   : true
DIGEST-MD5             : digest-md5 : true
EXTERNAL               : external   : true
GSSAPI                 : gssapi     : true
PLAIN                  : plain      : true
```

If necessary, you can use `dsconfig` to disable the GSSAPI SASL mechanism handler on the directory server host `directory.example.com`.

```
$ dsconfig -X -n -p 4444 -h directory.example.com \
  -D "cn=directory manager" -j pwd-file \
  set-sasl-mechanism-handler-prop \
  --handler-name GSSAPI \
  --set enabled:false
```

### 20.8.3.14 Directory Server Machine: Add a Test User to the Directory Server

To authenticate a Kerberos user to the directory server, there must be a directory entry for the user that corresponds to the Kerberos Principal for that user.

In a previous step, a test user was added to the Kerberos database with a Principal of `kerberos-test@EXAMPLE.COM`. Because of the identity mapping configuration

added to the directory, the corresponding directory entry for that user must have a DN of `uid=kerberos-test,ou=People,dc=example,dc=com`.

Before you can add the user to the directory, you must create the file `testuser.ldif` with the following contents.

***Example 20–4 New `testuser.ldif` File***

```
dn: uid=kerberos-test,ou=People,dc=example,dc=com
changetype: add
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
uid: kerberos-test
givenName: Kerberos
sn: Test
cn: Kerberos Test
description: An account for testing Kerberos authentication through GSSAPI
```

Next, use `ldapmodify` to add this entry to the server:

```
$ ldapmodify -D "cn=Directory Manager" -w - -f testuser.ldif
adding new entry uid=kerberos-test,ou=People,dc=example,dc=com
$
```

#### 20.8.3.15 Directory Server Machine: Obtain a Kerberos Ticket as the Test User

The test user exists in the Kerberos database, the directory server, and the KDC. Therefore, it is now possible to authenticate as the test user to the directory server over Kerberos through GSSAPI.

First, use the `kinit` command to get a Kerberos ticket for the user, as shown in the following example:

```
$ kinit kerberos-test
Password for kerberos-test@EXAMPLE.COM: secret
$
```

Then, use the `klist` command to view information about this ticket:

```
$ klist
Kerberos 5 ticket cache: 'API:6'
Default principal: kerberos-test@EXAMPLE.COM
Valid Starting     Expires            Service Principal
03/23/09 12:35:05  03/23/09 20:35:05  krbtgt/EXAMPLE.COM@EXAMPLE.COM
renew until 03/30/09 12:34:15
$
```

#### 20.8.3.16 Client Machine: Authenticate to the Directory Server Through GSSAPI

The final step is to authenticate to the directory server by using GSSAPI. The `ldapsearch` utility provided with The directory server provides support for SASL authentication, including GSSAPI, DIGEST-MD5, and EXTERNAL mechanisms. However, to bind by using GSSAPI you must provide the client with the path to the SASL library. Provide the path by setting the `SASL_PATH` environment variable to the `lib/sasl` directory:

```
$ SASL_PATH=SASL-library
$ export SASL_PATH
$
```

To actually perform a Kerberos-based authentication to the directory server using `ldapsearch`, you must include the `-o mech=GSSAPI` and `-o authzid=`*principal* arguments.

You must also specify the fully qualified host name, shown here as `-h directory.example.com`, which must match the value of the `nsslapd-localhost` attribute on `cn=config` for the server. This use of the `-h` option is needed because the GSSAPI authentication process requires the host name provided by the client to match the host name provided by the server.

The following example retrieves the `dc=example,dc=com` entry while authenticated as the Kerberos test user account created previously:

```
$ ldapsearch -h directory.example.com -p 389 -o mech=GSSAPI \ -o
authzid="kerberos-test@EXAMPLE.COM" -b "dc=example,dc=com" -s base
"(objectClass=*)"
version: 1
dn: dc=example,dc=com
dc: example
objectClass: top
objectClass: domain
$
```

Check the directory server access log to confirm that the authentication was processed as expected:

```
$ tail -12 /local/ds/logs/access

[24/Jul/2004:00:30:47 -0500] conn=0 op=-1 msgId=-1 - fd=23 slot=23 LDAP
        connection from 1.1.1.8 to 1.1.1.8
[24/Jul/2004:00:30:47 -0500] conn=0 op=0 msgId=1 - BIND dn="" method=sasl
     version=3 mech=GSSAPI
[24/Jul/2004:00:30:47 -0500] conn=0 op=0 msgId=1 - RESULT err=14 tag=97
     nentries=0 etime=0, SASL bind in progress
[24/Jul/2004:00:30:47 -0500] conn=0 op=1 msgId=2 - BIND dn="" method=sasl
     version=3 mech=GSSAPI
[24/Jul/2004:00:30:47 -0500] conn=0 op=1 msgId=2 - RESULT err=14 tag=97
     nentries=0 etime=0, SASL bind in progress
[24/Jul/2004:00:30:47 -0500] conn=0 op=2 msgId=3 - BIND dn="" method=sasl
     version=3 mech=GSSAPI
[24/Jul/2004:00:30:47 -0500] conn=0 op=2 msgId=3 - RESULT err=0 tag=97
     nentries=0 etime=0 dn="uid=kerberos-test,ou=people,dc=example,dc=com"
[24/Jul/2004:00:30:47 -0500] conn=0 op=3 msgId=4 - SRCH base="dc=example,dc=com"
     scope=0 filter="(objectClass=*)" attrs=ALL
[24/Jul/2004:00:30:47 -0500] conn=0 op=3 msgId=4 - RESULT err=0 tag=101 nentries=1
     etime=0
[24/Jul/2004:00:30:47 -0500] conn=0 op=4 msgId=5 - UNBIND
[24/Jul/2004:00:30:47 -0500] conn=0 op=4 msgId=-1 - closing - U1
[24/Jul/2004:00:30:48 -0500] conn=0 op=-1 msgId=-1 - closed.
$
```

This example shows that the bind is a three-step process. The first two steps return LDAP result `14` (SASL bind in progress), and the third step shows that the bind was successful. The `method=sasl` and `mech=GSSAPI` tags show that the bind used the GSSAPI SASL mechanism. The `dn="uid=kerberos-test,ou=people,dc=example,dc=com"` at the end of the successful bind response shows that the bind was performed as the appropriate user.

### 20.8.4 Creating a Kerberos Workflow Element Using `dsconfig`

You can create a Kerberos workflow element by running the `dsconfig create-workflow-element` command:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
 create-workflow-element \
 --type KerberosAuthProviderWorkflowElement  \
 --element-name Kerberos_Test_WE  \
```

### 20.8.5 Troubleshooting Kerberos Configuration

If the Kerberos installation does not perform as expected, check the following conditions:

- Perform a successful `kinit` using the test principal from the directory server machine to make sure that the directory server can authenticate to the Kerberos KDC.
- Perform a successful `kinit` using the test principal from the client machines to make sure that the client machines can authenticate to the Kerberos KDC.
- Make sure that the directory server's keytab file exists and is readable by the directory server. That is, make sure that the keytab file's ownership and permission settings are correct.
- Make sure that the LDAP principal name in the keytab file matches the hostname that the directory server used when it was configured. The following example shows a configuration that fails:

  1. Configure GSSAPI as shown below. The value specified for the `server-fqdn` attribute, `bad.example.com`, does not match the value used in creating the keytab, `directory.example.com`.

     ```
     $ dsconfig -X -n -p 4444 -h directory.example.com \
       -D "cn=directory manager" -j pwd-file \
       set-sasl-mechanism-handler-prop \
       --handler-name GSSAPI \
       --set enabled:true \
       --set keytab:asinst_1/oud/config/oud.keytab \
       --set server-fqdn:bad.example.com
     ```

  2. From a client, attempt an ldapsearch authenticating using GSSAPI.

     ```
     $ ldapsearch -h directory.example.com \
       -o mech=GSSAPI -o authid=kerberos-test@EXAMPLE.COM \
       --searchScope base \
       -b "uid=kerberos-test,ou=people,dc=example,dc=com" "(objectclass=*)"
     An error occurred while attempting to perform GSSAPI authentication to
       the Directory Server: \
     PrivilegedActionException(AccessController.java:-2)
     Result Code:  82 (Local Error)
     ```

     The search fails as expected.

  3. To determine the cause of the search failure, inspect the directory server's access log:

     ```
     $ tail asinst_1/oud/logs/access
     [23/Mar/2009:13:12:59 -0500] CONNECT conn=14 from=129.150.33.77:65076
       to=192.168.0.199:1389 protocol=LDAP
     [23/Mar/2009:13:13:00 -0500] BIND REQ conn=14 op=0 msgID=1
       type=SASL mechanism=GSSAPI dn=""
     ```

```
[23/Mar/2009:13:13:00 -0500] BIND RES conn=14 op=0 msgID=1
  result=49 authFailureID=1310915   authFailureReason="An unexpected error
  occurred while trying to create an GSSAPI context:
  major code (13) No valid credentials provided,
  minor code (-1)   Failed to find any Kerberos Key" etime=253
[23/Mar/2009:13:13:00 -0500] DISCONNECT conn=14 reason="Client Disconnect"
```

The message in the minor code of the last record in the access log shows that the directory server could not find a match in the keytab file.

4. To fix the situation, disable the handler and then re-enable it with the correct information, as shown in the following example.

```
$ dsconfig -X -n -p 4444 -h directory.example.com \
  -D "cn=directory manager" -j pwd-file \
  set-sasl-mechanism-handler-prop \
  --handler-name GSSAPI \
  --set enabled:false
$ dsconfig -X -n -p 4444 -h directory.example.com
  -D "cn=directory manager" -j pwd-file \
  set-sasl-mechanism-handler-prop \
  --handler-name GSSAPI \
  --set enabled:true \
  --set keytab:asinst_1/oud/config/oud.keytab \
  --set server-fqdn:directory.example.com
$ ldapsearch -h directory.example.com \
  -o mech=GSSAPI \
  -o authid=kerberos-test@EXAMPLE.COM \
  --searchScope base \
  -b "uid=kerberos-test,ou=people,dc=example,dc=com" "(objectclass=*)"
dn: uid=kerberos-test,ou=People,dc=example,dc=com
changetype: add
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
uid: kerberos-test
givenName: Kerberos
sn: Test
cn: Kerberos Test
description: An account for testing Kerberos authentication through GSSAPI
```

## 20.9 Testing SSL, StartTLS, and SASL Authentication With ldapsearch

The ldapsearch utility included with the directory server is useful for testing that the server is properly configured to support SSL and StartTLS. This utility includes a number of options that are well-suited for testing in a number of different scenarios. This section describes how to use ldapsearch to test SSL and StartTLS communication, and SASL EXTERNAL authentication. The same process can be used with many of the other client tools provided with the directory server, including ldapmodify, ldapcompare, and ldapdelete.

### 20.9.1 ldapsearch Command Line Arguments Applicable To Security

The following command-line arguments are of particular interest when using the ldapsearch tool to communicate via SSL or StartTLS:

- `-h address` or `--hostname address` Specifies the address of the directory server to which you want to connect. If no value is specified, the IPv4 loopback address (`127.0.0.1`) is used.
- `-p port` or `--port port` Specifies the port number on which the directory server is listening for connections. If no value is specified, the standard unencrypted LDAP port (`389`) is used.
- `-Z` or `--useSSL` Indicates that the client should use SSL to secure communication with the directory server. If this option is used, the value specified for the port argument must be one on which the server is listening for SSL-based connections. The default LDAPS port is `636`.
- `-q` or `--startTLS` Indicates that the client should use the StartTLS extended operation to secure communication with the directory server. If this option is used, the value specified for the port argument must be the one on which the server is listening for clear-text LDAP connections. Note that the port argument is not required if the server is listening on the default LDAP port (`389`).
- `-r` or `--useSASLExternal` Indicates that the client should use SASL EXTERNAL authentication to authenticate to the directory server. If this option is used, you must also provide a keystore path.
- `-X` or `--trustAll` Indicates that the client should blindly trust any certificate that the directory server presents. This option should not be used in conjunction with the argument used to specify the trust store path.
- `-K path` or `--keyStorePath path` Specifies the path to the keystore that should be used if the client is to present a certificate to the directory server (for example, when using SASL EXTERNAL authentication). This should be the path to a JKS keystore.
- `-W password` or `--keyStorePassword password` Specifies the PIN required to access the contents of the key tore. This should not be used in conjunction with the keystore password file argument.
- `--keyStorePasswordFile path` Specifies the path to a file containing the PIN required to access the contents of the keystore. This should not be used in conjunction with the keystore password argument.
- `-N nickname` or `--certNickname nickname` Specifies the nickname, or alias, of the certificate that the client should present to the directory server. The keystore path argument must also be provided. If no nickname is given, then the client will pick the first acceptable client certificate that it finds in the keystore.
- `-P path` or `--trustStorePath path` Specifies the path to the JKS trust store file that the client should use when determining whether to trust the certificate presented by the directory server. If this argument is not given and the `trustAll` option is not given, then any certificate presented to the client will be displayed and the user will be prompted about whether to trust it.
- `--trustStorePassword password` Specifies the password needed to access the trust store contents. In most cases, no trust store password is required. This should not be used in conjunction with the trust store password file option.
- `--trustStorePasswordFile path` Specifies the path to a file containing the password needed to access the trust store contents. In most cases, no trust store password is required. This should not be used in conjunction with the trust store password option.
- `-E` or `--reportAuthzID` Indicates that the directory server should include the authorization identity of the authenticated user in the bind response. This is useful

when performing SASL authentication to determine the user to which the client certificate (or other form of SASL credentials if a mechanism other than EXTERNAL was used) was mapped.

## 20.9.2 Testing SSL

The following demonstrates the use of ldapsearch to communicate with a directory server using LDAP over SSL:

```
$ ldapsearch --hostname directory.example.com --port 1636 \
--useSSL --baseDN "" --searchScope base "(objectClass=*)"
```

In this case, no trust store was specified, and the --trustAll argument was also not given. Therefore, when the server presents its certificate to the client, the user will be prompted about whether that certificate should be trusted. The entire sequence might look something like:

```
$ ldapsearch --hostname directory.example.com --port 1636 \
--useSSL --baseDN "" --searchScope base "(objectClass=*)"

The server is using the following certificate:
Subject DN: CN=directory.example.com, O=Example Corp, C=US
Issuer DN: CN=directory.example.com, O=Example Corp, C=US
Validity: Fri Mar 02 16:48:17 CST 2007 through Thu might 31 17:48:17 CDT 2007
Do you want to trust this certificate and continue connecting to the server?
Please enter "yes" or "no":
dn:
objectClass: ds-rootDSE
objectClass: top
```

If the client simply wants to always trust any certificate that the server presents without being prompted, then the --trustAll argument might be provided. For example:

```
$ ldapsearch --hostname directory.example.com --port 1636 \
--useSSL --trustAll --baseDN "" --searchScope base \
"(objectClass=*)"
```

If the client has a trust store and wants to use that to determine whether to trust the server certificate, then the --trustStorePath argument might also be given. For example:

```
$ ldapsearch --hostname directory.example.com --port 1636 \
--useSSL --trustStorePath client.truststore --baseDN "" \
--searchScope base "(objectClass=*)"
```

## 20.9.3 Testing StartTLS

The process for using StartTLS with the ldapsearch utility is almost identical to the process for using SSL. The only differences are that you should use the port on which the server is listening for unencrypted LDAP requests and that you should indicate that StartTLS should be used instead of SSL (that is, use --useStartTLS instead of --useSSL). The following example is the equivalent of the first example given for using SSL with ldapsearch except that it uses StartTLS to secure the communication:

```
$ ldapsearch -h directory.example.com --port 1389 \
--useStartTLS --baseDN "" --searchScope base "(objectClass=*)"
```

This applies to all of the other examples given. Simply change the port number from the LDAPS port to the LDAP port, and replace the --useSSL option with --useStartTLS.

### 20.9.4 Testing SASL External Authentication

**Note:** SASL is not supported for use with a proxy server instance.

SASL EXTERNAL authentication might be used in conjunction with either SSL or StartTLS. The primary differences are that it will be necessary to provide a keystore that contains the client certificate, the PIN required to access the contents of that keystore, and a flag indicating that the client should use SASL EXTERNAL authentication. The following example demonstrates sample usage for such a command:

```
$ ldapsearch --hostname directory.example.com --port 1636 \
--useSSL --keyStorePath /path/to/client.keystore \
--keyStorePasswordFile /path/to/client.keystore.pin \
--useSASLExternal --certNickName nickname \
--baseDN "" --searchScope base \
"(objectClass=*)"
```

When using SASL EXTERNAL authentication, it is also often useful to ask the server to return the authorization identity to ensure that the authentication is being performed as the correct user. The following demonstrates an example of this process. (Note the value reported on the line beginning with the "#" character.)

```
$ ldapsearch --hostname directory.example.com --port 1636 \
--useSSL --keyStorePath /path/to/client.keystore \
--keyStorePasswordFile /path/to/client.keystore.pin \
--useSASLExternal --reportAuthzID --certNickName nickname \
--baseDN "" --searchScope base "(objectClass=*)"

# Bound with authorization ID dn:uid=test.user,dc=example,dc=com
dn:
objectClass: ds-rootDSE
objectClass: top
```

## 20.10 Debugging SSL Using OpenSSL s_client Test Utility

OpenSSL provides an extremely valuable and useful diagnostic tool, called s_client, to debug SSL servers. The command implements a generic SSL/TLS client which connects to a remote host using SSL/TLS.

This utility lets you test or debug servers that use SSL/TLS with a powerful command line utility. To test the secure connections to the Oracle Unified Directory server, type the following command on the command prompt:

```
openssl s_client -connect <host>:<port> [options]
```

Here:

s_client: It is an SSL/TLS test client, which is used to test secure servers. The test client can connect to a secure port, while providing a detailed log of the steps performed during the SSL/TLS handshake.

`hostname:port`: This specifies the host and optional port to connect to. If not specified then an attempt is made to connect to the local host on port `443`, since https uses port `443`.

If connected, you can manually type in several commands, such as "GET /" and "HEAD / HTTP/1.0" for secure servers. However, if the handshake fails then there are several possible causes. If you want to know the problem you are experiencing is related to the application, firewall, certificate trust, or so on then this section describes a way to eliminate SSL from your list of usual suspects.

This section contains the following topics:



## 20.10.1 Scenario 1- Connection Refused

You connect the SSL client over the designated SSL port, but the connection fails. Consider the following example to demonstrate this scenario:

```
openssl s_client -connect localhost:<ldaps_portnumber>
connect: Connection refused
connect:errno=146
```

### Solution

A possible solution is to check the correct value of LDAPS number in config.ldif file.

## 20.10.2 Scenario 2- Verify Return Code: 18 (Self Signed Certificate)

When you receive an error code 18, this implies your SSL client program failed to establish the secure connection (https) with the server due to certificate chain verification failure. The server that you using is a self-signed certificate, and you need to use a certificate chain.

Consider the following example to demonstrate this scenario:

```
openssl s_client -connect localhost:<ldaps-port-number>
CONNECTED(00000004)
depth=0 /C=ca/ST=California/L=SF/O=Oracle/OU=ldap/CN=server admin
verify error:num=18:self signed certificate
verify return:1
depth=0 /C=ca/ST=California/L=SF/O=Oracle/OU=ldap/CN=server admin
verify return:1
---
Certificate chain
 0 s:/C=ca/ST=California/L=SF/O=Oracle/OU=ldap/CN=server admin
   i:/C=ca/ST=California/L=SF/O=Oracle/OU=ldap/CN=server admin
---
Server certificate
-----BEGIN CERTIFICATE-----
MIIDBjCCAsSgAwIBAgIETxRMvTALBgcqhkjOOAQDBQAwZjELMAkGA1UEBhMCY2Ex
EzARBgNVBAgTCkNhbGlmb3JuaWExCzAJBgNVBAcTAlNGMQ8wDQYDVQQKEwZPcmFj
bGUxDTALBgNVBAsTBGxkYXAxFTATBgNVBAMTDHNlcnZlciBhZG1pbjAeFw0xMjAx
```

```
MTYxNjEzNDlaFw0xMjA0MTUxNjEzNDlaMGYxCzAJBgNVBAYTAmNhMRMwEQYDVQQI
EwpDYWxpZm9ybmlhMQswCQYDVQQHEwJTRjEPMA0GA1UEChMGT3JhY2xlMQ0wCwYD
VQQLEwRsZGFwMRUwEwYDVQQDEwxzZXJ2ZXIgYWRtaW4wggG4MIIBLAYHKoZIzjgE
ATCCAR8CgYEA/X9TgR11EilS30qcLuzk5/YRt1I870QAwx4/gLZRJmlFXUAiUftZ
PY1Y+r/F9bow9subVWzXgTuAHTRv8mZgt2uZUKWkn5/oBHsQIsJPu6nX/rfGG/g7
V+fGqKYVDwT7g/bTxR7DAjVUE1oWkTL2dfOuK2HXKu/yIgMZndFIAccCFQCXYFCP
FSMLzLKSuYKi64QL8Fgc9QKBgQD34aCF1ps93su8q1w2uFe5eZSvu/o66oL5V0wL
PQeCZ1FZV4661FlP5nEHEIGAtEkWcSPoTCgWE7fPCTKMyKbhPBZ6i1R8jSjgo64e
K7OmdZFuo38L+iE1YvH7YnoBJDvMpPG+qFGQiaiD3+Fa5Z8GkotmXoB7VSVkAUw7
/s9JKgOBhQACgYEAw+2EIpmwy0rqtHbNb6gxbEtW0hplXXQdHEQp24brde1jt1qv
LDz/c8KR+fVxqvTxAmurGt1qbrhjXcUxi1KdaLnLnLXTCoD+ZLQU+F6B/TNmfrxb
AJmHtmoZsFtNCBTC++FClXtconKyXjEWnKMw7fEb+gNY3eTUrcyIpa/YEbYwCwYH
KoZIzjgEAwUAAy8AMCwCFEtf5+J77Q/5fI6bZ7k3D1rdbw6UAhQkWGmp8VOiMdUg
5K4wK7Y7cC0wSQ==
-----END CERTIFICATE-----
subject=/C=ca/ST=California/L=SF/O=Oracle/OU=ldap/CN=server admin
issuer=/C=ca/ST=California/L=SF/O=Oracle/OU=ldap/CN=server admin
---
Acceptable client certificate CA names
/C=FR/ST=France/L=Grenoble/O=Oracle/OU=OUD/CN=CA Certificate
/C=ca/ST=California/L=SF/O=Oracle/OU=ldap/CN=user.41
/C=ca/ST=California/L=SF/O=Oracle/OU=ldap/CN=server admin
---
SSL handshake has read 1594 bytes and written 312 bytes
---
New, TLSv1/SSLv3, Cipher is EDH-DSS-DES-CBC3-SHA
Server public key is 1024 bit
SSL-Session:
    Protocol  : TLSv1
    Cipher    : EDH-DSS-DES-CBC3-SHA
    Session-ID: 4F16C3F27655013F71AE2120134A8D1AFE966A1D9233618507DEFE9C607417AA
    Session-ID-ctx:
    Master-Key:
57BDB7FCA9A293E65274AA7CDD0E7CC48AA227806FC2B54C9F9E36BB26D32943FC115CE4FF9A605B6B
6BD237026F3D0E
    Key-Arg   : None
    Start Time: 1326892018
    Timeout   : 300 (sec)
    Verify return code: 18 (self signed certificate)
```

#### Solution

You must import in the server key store, signed certificate reply, and CA certificate.

## 20.10.3 Scenario 3 - Verify Return Code: 0 (ok)

If a connection is successfully established with an SSL server, then you receive a return code `0`. This implies that any data received from the server is displayed and any key presses will be sent to the server. In addition, the certificate chain in use is also displayed.

Consider the following example to demonstrate a working session:

```
openssl s_client -connect localhost:8636 -verify 250 \
-key $SERVER_SSL/config/keystore   -CApath $CA_SSL -CAfile ca-cert.pem

-key is specifying the path to teh server keystore
-CAPath/-CAfile allows to locate CA certificate (pem format)

verify depth is 250
```

```
CONNECTED(00000004)
depth=1 /C=FR/ST=France/L=Grenoble/O=Oracle/OU=OUD/CN=CA Certificate
verify return:1
depth=0 /C=ca/ST=California/L=SF/O=Oracle/OU=ldap/CN=server admin
verify return:1
---
Certificate chain
 0 s:/C=ca/ST=California/L=SF/O=Oracle/OU=ldap/CN=server admin
   i:/C=FR/ST=France/L=Grenoble/O=Oracle/OU=OUD/CN=CA Certificate
 1 s:/C=FR/ST=France/L=Grenoble/O=Oracle/OU=OUD/CN=CA Certificate
   i:/C=FR/ST=France/L=Grenoble/O=Oracle/OU=OUD/CN=CA Certificate
---
Server certificate
-----BEGIN CERTIFICATE-----
MIIDYDCCAsmgAwIBAgIFAJbW4rkwDQYJKoZIhvcNAQEFBQAwaTELMAkGA1UEBhMC
RlIxDzANBgNVBAgTBkZyYW5jZTERMA8GA1UEBxMIR3Jlbm9ibGUxDzANBgNVBAoT
Bk9yYWNsZTEMMAoGA1UECxMDT1VEMRcwFQYDVQQDEw5DQSBDZXJ0aWZpY2F0ZTAe
Fw0xMjAxMTcxMDQ5MjdaFw0xMjA0MTcxMDQ5MjdaMGYxCzAJBgNVBAYTAmNhMRMw
EQYDVQQIEwpDYWxpZm9ybmlhMQswCQYDVQQHEwJTRjEPMA0GA1UEChMGT3JhY2xl
MQ0wCwYDVQQLEwRsZGFwMRUwEwYDVQQDEwxzZXJ2ZXIgYWRtaW4wggG3MIIBLAYH
KoZIzjgEATCCAR8CgYEA/X9TgR11EilS30qcLuzk5/YRt1I870QAwx4/gLZRJmlF
XUAiUftZPY1Y+r/F9bow9subVWzXgTuAHTRv8mZgt2uZUKWkn5/oBHsQIsJPu6nX
/rfGG/g7V+fGqKYVDwT7g/bTxR7DAjVUE1oWkTL2dfOuK2HXKu/yIgMZndFIAccC
FQCXYFCPFSMLzLKSuYKi64QL8Fgc9QKBgQD34aCF1ps93su8q1w2uFe5eZSvu/o6
6oL5V0wLPQeCZ1FZV4661Flp5nEHEIGAtEkWcSPoTCgWE7fPCTKMyKbhPBZ6i1R8
jSjgo64eK7OmdZFuo38L+iE1YvH7YnoBJDvMpPG+qFGQiaiD3+Fa5Z8GkotmXoB7
VSVkAUw7/s9JKgOBhAACgYA8N/yzB5rrvNOPhOrea1RNCRePn0bMvXkDpfUs8dpH
z1qQog4soloAhojIYJYA3OGqKr3ryNnfB0B8lePQ1ZaJgkURqOjiVKF6xv5FmnuM
C1uwiTfr/9IKijiy8oCKKKSLTB5lY3Rk0o03D+LrqgLp27A41WvvhGo4djBqXse1
OTANBgkqhkiG9w0BAQUFAAOBgQBzTpgFc1YCpo8QKeoDBRag4tn2y8BzkeLeLMgy
gQAYCGNjJjrV0ChYKMJnqLPCrP9+/Otyj9ZByn9+T1Jx9/khuh9oNXCwF5FUE5VE
gkn3kPo1LdLBqKpfUSeFcYNJDQDhtThVwEq05Ifm+JuCCM4J3BbFuZpJM5xnbcIZ
mjcn5w==
-----END CERTIFICATE-----
subject=/C=ca/ST=California/L=SF/O=Oracle/OU=ldap/CN=server admin
issuer=/C=FR/ST=France/L=Grenoble/O=Oracle/OU=OUD/CN=CA Certificate
---Verify return code: 0 (ok)
Acceptable client certificate CA names
/C=FR/ST=France/L=Grenoble/O=Oracle/OU=OUD/CN=CA Certificate
/C=fr/ST=Isere/L=Montbonnot/O=Oracle/OU=ldap/CN=server_8839
---
SSL handshake has read 2179 bytes and written 312 bytes
---
New, TLSv1/SSLv3, Cipher is EDH-DSS-DES-CBC3-SHA
Server public key is 1024 bit
SSL-Session:
    Protocol  : TLSv1
    Cipher    : EDH-DSS-DES-CBC3-SHA
    Session-ID: 4F16C59B172D329E44AF199B4E49B14E54163AAF783A68FBD48556FCB06A9238
    Session-ID-ctx:
    Master-Key:
21CC1BBF638FFDAF16E5DBAB337728D029F0125D483636EF7590BE3005DDA96AEAF60DE88172DE9258
06F633EB09ACBE
    Key-Arg   : None
    Start Time: 1326892443
    Timeout   : 300 (sec)
    Verify return code: 0 (ok)
```

### 20.10.4 Scenario 4 - SSLHandshakeException

When you try to establish a server secure connection, the following error message is issued by the ldapsearch:

```
ldapsearch  -p 7636 -D "cn=Directory Manager" -w secret12  -P config/truststore -Z
-b dc=example,dc=com uid=user.0 Cannot send the simple bind request:
SSLHandshakeException(sun.security.validator.ValidatorException: PKIX path
building failed: sun.security.provider.certpath.SunCertPathBuilderException:
unable to find valid certification path to requested target)
```

This error appears because the server certificate is self signed certificate and not a certificate chain. You will receive an error code `18`.

The following demonstrates an example of this process.

```
openssl s_client -connect localhost:7636
CONNECTED(00000004)
depth=0 /C=ca/ST=California/L=SF/O=Oracle/OU=ldap/CN=server admin
verify error:num=18:self signed certificate
verify return:1
depth=0 /C=ca/ST=California/L=SF/O=Oracle/OU=ldap/CN=server admin
verify return:1
---
Certificate chain
 0 s:/C=ca/ST=California/L=SF/O=Oracle/OU=ldap/CN=server admin
   i:/C=ca/ST=California/L=SF/O=Oracle/OU=ldap/CN=server admin
---
Server certificate
-----BEGIN CERTIFICATE-----
MIIDBjCCAsSgAwIBAgIETxRMvTALBgcqhkjOOAQDBQAwZjELMAkGA1UEBhMCY2Ex
EzARBgNVBAgTCkNhbGlmb3JuaWExCzAJBgNVBAcTAlNGMQ8wDQYDVQQKEwZPcmFj
bGUxDTALBgNVBAsTBGxkYXAxFTATBgNVBAMTDHNlcnZlciBhZG1pbjAeFw0xMjAx
MTYxNjEzNDlaFw0xMjA0MTUxNjEzNDlaMGYxCzAJBgNVBAYTAmNhMRMwEQYDVQQI
EwpDYWxpZm9ybmlhMQswCQYDVQQHEwJTRjEPMA0GA1UEChMGT3JhY2xlMQ0wCwYD
VQQLEwRsZGFwMRUwEwYDVQQDEwxzZXJ2ZXIgYWRtaW4wggG4MIIBLAYHKoZIzjgE
ATCCAR8CgYEA/X9TgR11EilS30qcLuzk5/YRt1I870QAwx4/gLZRJmlFXUAiUftZ
PY1Y+r/F9bow9subVWzXgTuAHTRv8mZgt2uZUKWkn5/oBHsQIsJPu6nX/rfGG/g7
V+fGqKYVDwT7g/bTxR7DAjVUE1oWkTL2dfOuK2HXKu/yIgMZndFIAccCFQCXYFCP
FSMLzLKSuYKi64QL8Fgc9QKBgQD34aCF1ps93su8q1w2uFe5eZSvu/o66oL5V0wL
PQeCZ1FZV4661FlP5nEHEIGAtEkWcSPoTCgWE7fPCTKMyKbhPBZ6i1R8jSjgo64e
K7OmdZFuo38L+iE1YvH7YnoBJDvMpPG+qFGQiaiD3+Fa5Z8GkotmXoB7VSVkAUw7
/s9JKgOBhQACgYEAw+2EIpmwy0rqtHbNb6gxbEtW0hplXXQdHEQp24brde1jt1qv
LDz/c8KR+fVxqvTxAmurGt1qbrhjXcUxi1KdaLnLnLXTCoD+ZLQU+F6B/TNmfrxb
AJmHtmoZsFtNCBTC++FClXtconKyXjEWnKMw7fEb+gNY3eTUrcyIpa/YEbYwCwYH
KoZIzjgEAwUAAy8AMCwCFEtf5+J77Q/5fI6bZ7k3D1rdbw6UAhQkWGmp8VOiMdUg
5K4wK7Y7cC0wSQ==
-----END CERTIFICATE-----
subject=/C=ca/ST=California/L=SF/O=Oracle/OU=ldap/CN=server admin
issuer=/C=ca/ST=California/L=SF/O=Oracle/OU=ldap/CN=server admin
---
Acceptable client certificate CA names
/C=FR/ST=France/L=Grenoble/O=Oracle/OU=OUD/CN=CA Certificate
/C=ca/ST=California/L=SF/O=Oracle/OU=ldap/CN=user.41
/C=ca/ST=California/L=SF/O=Oracle/OU=ldap/CN=server admin
---
SSL handshake has read 1594 bytes and written 312 bytes
---
New, TLSv1/SSLv3, Cipher is EDH-DSS-DES-CBC3-SHA
Server public key is 1024 bit
SSL-Session:
```

```
    Protocol  : TLSv1
    Cipher    : EDH-DSS-DES-CBC3-SHA
    Session-ID: 4F16C3F27655013F71AE2120134A8D1AFE966A1D9233618507DEFE9C607417AA
    Session-ID-ctx:
    Master-Key:
57BDB7FCA9A293E65274AA7CDD0E7CC48AA227806FC2B54C9F9E36BB26D32943FC115CE4FF9A605B6B
6BD237026F3D0E
    Key-Arg   : None
    Start Time: 1326892018
    Timeout   : 300 (sec)
    Verify return code: 18 (self signed certificate)
```

**Solution**

Perform the following steps to fix the issue:

1. Import the CA certificate into the server keystore.

```
keytool -importcert -alias ca-cert -keystore config/keystore -storetype JKS
-file $CA_SSL/ca-cert.pem
Enter keystore password:
Owner: CN=CA Certificate, OU=OUD, O=Oracle, L=Grenoble, ST=France, C=FR
Issuer: CN=CA Certificate, OU=OUD, O=Oracle, L=Grenoble, ST=France, C=FR
Serial number: 96b69e65
Valid from: Wed Jan 04 15:51:37 MET 2012 until: Mon Sep 04 16:51:37 MEST 2428
Certificate fingerprints:
         MD5:  D0:5B:C8:2A:3D:3B:09:07:5A:29:62:E3:27:99:4E:D4
         SHA1: E4:C9:BB:B7:5B:49:C7:7E:BF:8B:C3:C3:DC:DF:29:E7:74:A0:66:03
         Signature algorithm name: SHA1withRSA
         Version: 3
Trust this certificate? [no]:  yes
Certificate was added to keystore
```

2. Import the signed server certificate reply into the server keystore.

```
keytool -importcert -trustcacerts -alias server-cert -keystore config/keystore
-storetype JKS -file server-cert.pem Enter keystore password:
Certificate reply was installed in keystore
```

3. List certificates in the LDAP server keystore.

```
keytool -list -keystore config/keystore -storepass secret12 -v

Keystore type: JKS
Keystore provider: SUN

Your keystore contains 2 entries

Alias name: ca-cert
Creation date: Jan 18, 2012
Entry type: trustedCertEntry

Owner: CN=CA Certificate, OU=OUD, O=Oracle, L=Grenoble, ST=France, C=FR
Issuer: CN=CA Certificate, OU=OUD, O=Oracle, L=Grenoble, ST=France, C=FR
Serial number: 96b69e65
Valid from: Wed Jan 04 15:51:37 MET 2012 until: Mon Sep 04 16:51:37 MEST 2428
Certificate fingerprints:
         MD5:  D0:5B:C8:2A:3D:3B:09:07:5A:29:62:E3:27:99:4E:D4
         SHA1: E4:C9:BB:B7:5B:49:C7:7E:BF:8B:C3:C3:DC:DF:29:E7:74:A0:66:03
         Signature algorithm name: SHA1withRSA
         Version: 3
```

4. Verify the connection with a ldapsearch request over SSL.

```
ldapsearch  -p 7636 -D "cn=Directory Manager" -w secret12  -P config/truststore
-Z -b dc=example,dc=com uid=user.0
dn: uid=user.0,ou=People,dc=example,dc=com
postalAddress: Aaccf Amar$01251 Chestnut Street$Panama City, DE  50369
postalCode: 50369
uid: user.0
description: This is the description for Aaccf Amar.
userPassword: {SSHA}vVIy4fjEUyt0L8GSVzX+VrJKEgGASLkeCvL1ng==
employeeNumber: 0
initials: ASA
givenName: Aaccf
objectClass: person
objectClass: inetorgperson
objectClass: organizationalperson
objectClass: top
pager: +1 779 041 6341
mobile: +1 010 154 3228
cn: Aaccf Amar
telephoneNumber: +1 685 622 6202
sn: Amar
street: 01251 Chestnut Street
homePhone: +1 225 216 5900
mail: user.0@maildomain.net
l: Panama City
st: DE
```

5. Access the log.

```
[18/Jan/2012:16:39:24 +0100] CONNECT conn=1 from=127.0.0.1:46726
to=127.0.0.1:7636 protocol=LDAPS
[18/Jan/2012:16:39:24 +0100] BIND REQ conn=1 op=0 msgID=1 type=SIMPLE
dn="cn=Directory Manager"
[18/Jan/2012:16:39:24 +0100] BIND RES conn=1 op=0 msgID=1 result=0
authDN="cn=Directory Manager,cn=Root DNs,cn=config" etime=31
[18/Jan/2012:16:39:24 +0100] SEARCH REQ conn=1 op=1 msgID=2
base="dc=example,dc=com" scope=wholeSubtree filter="(uid=user.0)" attrs="ALL"
[18/Jan/2012:16:39:24 +0100] SEARCH RES conn=1 op=1 msgID=2 result=0 nentries=1
etime=18
[18/Jan/2012:16:39:24 +0100] UNBIND REQ conn=1 op=2 msgID=3
[18/Jan/2012:16:39:24 +0100] DISCONNECT conn=1 reason="Client Disconnect"
```

## 20.10.5 Scenario 5 - SASL EXTERNAL Bind Request Could Not Be Processed

When you try to perform OUD SASL client external authentication over SSL the following error message appears:

```
ldapsearch -p 7636 -Z -K /export/home/oud/security/client/config/keystore -W
secret12 -P /export/home/oud/security/client/config/truststore
--trustStorePassword secret12 -N user.41-cert --useSASLExternal -b
dc=example,dc=com uid=user.0
The SASL EXTERNAL bind attempt failed
Result Code:  49 (Invalid Credentials)
```

When you view the access log, then the following message is shown:

```
CONNECT conn=2 from=127.0.0.1:46763 to=127.0.0.1:7636 protocol=LDAPS
[18/Jan/2012:17:48:44 +0100] BIND REQ conn=2 op=0 msgID=1 type=SASL
mechanism=EXTERNAL dn=""
```

```
[18/Jan/2012:17:48:44 +0100] BIND RES conn=2 op=0 msgID=1 result=49
authFailureID=1245310 authFailureReason="The SASL EXTERNAL bind request could not
be processed because the client did not present a certificate chain during SSL/TLS
negotiation" etime=6
[18/Jan/2012:17:48:44 +0100] DISCONNECT conn=2 reason="Client Disconnect"
```

This error appears because the client certificate is not a valid certificate chain.

**Solution**

Perform the following steps to fix this issue:

1. Import the CA certificate into the client keystore.

```
keytool -importcert -alias ca-cert -keystore config/keystore \
-storetype JKS -file $CA_SSL/ca-cert.pem
Enter keystore password:
Owner: CN=CA Certificate, OU=OUD, O=Oracle, L=Grenoble, ST=France, C=FR
Issuer: CN=CA Certificate, OU=OUD, O=Oracle, L=Grenoble, ST=France, C=FR
Serial number: 96b69e65
Valid from: Wed Jan 04 15:51:37 MET 2012 until: Mon Sep 04 16:51:37 MEST 2428
Certificate fingerprints:
         MD5:  D0:5B:C8:2A:3D:3B:09:07:5A:29:62:E3:27:99:4E:D4
         SHA1: E4:C9:BB:B7:5B:49:C7:7E:BF:8B:C3:C3:DC:DF:29:E7:74:A0:66:03
         Signature algorithm name: SHA1withRSA
         Version: 3
Trust this certificate? [no]:  yes
Certificate was added to keystore
```

2. Import the user signed reply certificate into the client keystore.

```
keytool -importcert -trustcacerts -alias user.41-cert -keystore config/keystore
-storetype JKS -file user.41-cert.pem -storepass secret12
Certificate reply was installed in keystore
```

3. Run the ldap command.

```
ldapsearch -p 7636 -Z -K /export/home/oud/security/client/config/keystore -W
secret12 -P /export/home/oud/security/client/config/truststore
--trustStorePassword secret12 -N user.41-cert --useSASLExternal -b
dc=example,dc=com uid=user.0
dn: uid=user.0,ou=People,dc=example,dc=com
postalAddress: Aaccf Amar$01251 Chestnut Street$Panama City, DE  50369
postalCode: 50369
uid: user.0
description: This is the description for Aaccf Amar.
employeeNumber: 0
initials: ASA
givenName: Aaccf
objectClass: person
objectClass: inetorgperson
objectClass: organizationalperson
objectClass: top
pager: +1 779 041 6341
mobile: +1 010 154 3228
cn: Aaccf Amar
telephoneNumber: +1 685 622 6202
sn: Amar
street: 01251 Chestnut Street
homePhone: +1 225 216 5900
mail: user.0@maildomain.net
l: Panama City
```

```
st: DE
```

4. Validate the log.

```
[18/Jan/2012:18:04:49 +0100] CONNECT conn=3 from=127.0.0.1:46777
to=127.0.0.1:7636 protocol=LDAPS
[18/Jan/2012:18:04:49 +0100] BIND REQ conn=3 op=0 msgID=1 type=SASL
mechanism=EXTERNAL dn=""
[18/Jan/2012:18:04:49 +0100] BIND RES conn=3 op=0 msgID=1 result=0
authDN="uid=user.41,ou=People,dc=example,dc=com" etime=37
[18/Jan/2012:18:04:49 +0100] SEARCH REQ conn=3 op=1 msgID=2
base="dc=example,dc=com" scope=wholeSubtree filter="(uid=user.0)" attrs="ALL"
[18/Jan/2012:18:04:49 +0100] SEARCH RES conn=3 op=1 msgID=2 result=0 nentries=1
etime=15
[18/Jan/2012:18:04:49 +0100] UNBIND REQ conn=3 op=2 msgID=3
[18/Jan/2012:18:04:49 +0100] DISCONNECT conn=3 reason="Client Disconnect"
```

## 20.11 Debugging SSL or TLS Using Java Debug Information

You can troubleshoot network Traffic for SSL or TLS connections using Java debug information.

There are situations when the only way to analyze SSL is to trace network access. Oracle Unified Directory allows you to debug SSL by adding `-Djavax.net.debug=all` option to the server in the *config/java.properties* file.

A sample debug output is as follows:

```
server.core.DirectoryServer (alert type org.opends.server.DirectoryServerStarted,
alert ID 458887):  The Directory Server has started successfully
***
found key for : server-cert
chain [0] = [
[
  Version: V3
  Subject: CN=server admin, OU=ldap, O=mycompany, L=City1, ST=Country1, C=ca
  Signature Algorithm: SHA1withRSA, OID = 1.2.840.113549.1.1.5

  Key:  SunPKCS11-Solaris DSA public key, 1024 bits (id 22714576, session object)
  y:
137585178298829672773992262717400783035252676067753730258902133887472765737596162 8
650688647577510816321283250872882407370491996059918680923417848100018238935577641 0
228200735673010501146203945919143729329772551286385346818351986257750540195836208 6
546885405080570540575677103845462467633475547155894544465662390
  p:
178011905478542266528237562450159990145232156369120674273274450314442865788737020 7
706126952521234630795671567847784664499706507709207278570500096683881440341297452 2
117181850604723115003930107995935806739534871706631980226201971496652413506094591 3
707594956514672855690606794135837542707371727429551343320695239
  q: 864205495604807476120572616017955259175325408501
  g:
174068207532402095185811980123523436538604490794561350978495831040599953488455823 1
478515974089409507253077970949157594923683005742524387610370844734671801488761181 0
308304375498519098347260155049469132948808339549231385000036164648264460849230407 8
721818959999056496097769368017749273708962006689187956744210730
  Validity: [From: Mon Jan 16 17:15:45 MET 2012,
             To: Mon Apr 16 18:15:45 MEST 2012]
  Issuer: CN=CA Certificate, OU=OUD, O=mycompany, L=City2, ST=Country2, C=FR
  SerialNumber: [96d4f0dc]
```

```
]
  Algorithm: [SHA1withRSA]
  Signature:
0000: 72 F6 7E 93 2B 87 B9 C7   39 51 4C D2 A7 B0 AA 36  r...+...9QL....6
0010: B8 0F BA C4 6E 43 70 72   81 50 09 7A 88 05 16 A2  ....nCpr.P.z....
0020: 1C 96 C2 49 B3 0A F9 AB   2B 4B 8D 59 4C BA 58 C9  ...I....+K.YL.X.
0030: EF B9 48 58 A7 C5 BB B5   0E 64 51 CF BC 58 DA 71  ..HX.....dQ..X.q
0040: E1 F7 2E A4 1D 1B FC D5   4F B2 70 B0 78 F4 FB E6  ........O.p.x...
0050: C4 6A 6A E0 DE B0 F5 98   7B 09 A9 A4 9D 17 4C F5  .jj...........L.
0060: 9F 06 07 E1 09 81 77 9E   41 3C 02 4C FB D8 94 ED  ......w.A<.L....
0070: 36 6A 65 5A 96 2C AE A4   86 83 66 63 BC 3C 8C 47  6jeZ.,....fc.<.G

]
```

The preceding information is provided in addition to the Oracle Unified Directory debug log text.

This section describes how to work with SSL debug recording and contains the following topics:

- Section 20.11.1, "Enabling SSL Debug Recording"
- Section 20.11.2, "Disabling SSL Debug Recording"

### 20.11.1 Enabling SSL Debug Recording

Perform the following steps to enable SSL debug recording:

1. Update the `start-ds.java-args` property in the *config/java.properties* file with:

   ```
   start-ds.java-args=-server -Djavax.net.debug=all
   ```

2. Run the `dsjavaproperties` command as described in Section A.2.5, "dsjavaproperties."
3. Stop the server instance using the `stop-ds` command.
4. Restart the server instance using the `start-ds` command.

> **Note:** The SSL debug information is logged in the *logs/server.out* file.

### 20.11.2 Disabling SSL Debug Recording

Perform the following steps to disable SSL debug recording:

1. Delete the `-Djavax.net.debug=all` property from *java.properties* file.

   ```
   start-ds.java-args=-server
   ```

2. Run the `dsjavaproperties` command as described in Section A.2.5, "dsjavaproperties."
3. Stop the server instance using the `stop-ds` command.
4. Restart the server instance using the `start-ds` command.

## 20.12 Controlling Connection Access Using Allowed and Denied Rules

You can use connection handler allowed and denied client rules to control which hosts can make TCP connections to the server. Connection handlers are responsible for accepting connections to the server.

The different types of connection handlers and their configuration properties are presented in this section and include the following:

- `allowed-client`. Specifies a set of host names or address masks that determine the clients that are allowed to establish connections to this Connection Handler. Valid values include a host name, a fully qualified domain name, a domain name, an IP address, or a subnetwork with subnetwork mask.
- `denied-client`. Specifies a set of host names or address masks that determine the clients that are not allowed to establish connections to this Connection Handler. Valid values include a host name, a fully qualified domain name, a domain name, an IP address, or a subnetwork with subnetwork mask. If both allowed and denied client masks are defined and a client connection matches one or more masks in both lists, then the connection is denied. If only a denied list is specified, then any client not matching a mask in that list is allowed.

**Note:** Both IPv4 and IPv6 addresses are supported.

## 20.12.1 Property Syntax of Allowed and Denied Client Rules

The `allowed-client` and `denied-client` properties share the same syntax to perform pattern matching against IP (IPv4 or IPv6) addresses and host names.

The following syntaxes are supported:

- IP address - The IP address of the clients to be allowed or denied can be specified in the rule. For example:

```
ds-cfg-denied-client:   192.168.5.6
ds-cfg-allowed-client:  2001:fecd:ba23:cd1f:dcb1:1010:9234:4088
```

- IP address with CIDR notation - A range of IP addresses can be allowed or denied by specifying an IP address using CIDR notation. For example:

```
ds-cfg-denied-client:   192.168.5.6/28
ds-cfg-allowed-client:  2001:0db8:1234::/48
```

  The first denies clients in the range `192.168.5.0 - 192.168.5.15` and the second allows clients in the range `2001:0db8:1234:0000:0000:0000:0000:0000 - 2001:0db8:1234:ffff:ffff:ffff:ffff:ffff`.

- IP address with '*' notation - A range of IP addresses (IPv4 only) can be allowed or denied by specifying an IP address with a '*' character to match parts of the IP address. For example:

```
ds-cfg-denied-client:   192.168.5.*
ds-cfg-allowed-client:   129.45.*.*
```

  The first example denies clients with IP addresses starting with 192.168.5 and the second allows clients with IP address starting with 129.45. Notice that the second example uses multiple match characters. To allow all IP addresses to match, the rule would look like:

```
ds-cfg-denied-client:   *.*.*.*
```

- DNS names - Clients can be restricted by DNS name. For example to restrict clients with the host name `foo.example.com`, enter:

```
ds-cfg-denied-client:   foo.example.com
```

- DNS names with pattern matching - This is similar to IP address pattern matching. The property can specify the '*' character to match parts of the DN name:

  ```
  ds-cfg-allowed-client: foo.*.test.com
  ```

  The property allows clients with DN names such as: `foo.bar.test.com` or `foo.foobar.test.com`. To only match DNS names ending in a suffix the property would be:

  ```
  ds-cfg-allowed-client: .example.com
  ```

  This property allows clients with DNS names such as: `test.example.com` or `test.me.example.com`.

  **Note:** Be careful when you use the DNS properties because the host name resolution depends on the server name service configuration.

## 20.12.2 Configuring Allowed and Denied Client Rules

Each connection handler needs to have its own set of rules. For example:

```
dn: cn=LDAP Connection Handler,cn=Connection Handlers,cn=config
objectClass: top
objectClass: ds-cfg-connection-handler
objectClass: ds-cfg-ldap-connection-handler
cn: LDAP Connection Handler
ds-cfg-java-class: org.opends.server.protocols.ldap.LDAPConnectionHandler
ds-cfg-enabled: true
ds-cfg-listen-address: 0.0.0.0
ds-cfg-listen-port: 389
ds-cfg-accept-backlog: 128
ds-cfg-allow-ldap-v2: true
ds-cfg-keep-stats: true
ds-cfg-use-tcp-keep-alive: true
ds-cfg-use-tcp-no-delay: true
ds-cfg-allow-tcp-reuse-address: true
ds-cfg-send-rejection-notice: true
ds-cfg-max-request-size: 5 megabytes
ds-cfg-max-blocked-write-time-limit: 2 minutes
ds-cfg-num-request-handlers: 2
ds-cfg-allow-start-tls: false
ds-cfg-use-ssl: false
ds-cfg-ssl-client-auth-policy: optional
ds-cfg-ssl-cert-nickname: server-cert
ds-cfg-denied-client: *.example.com
ds-cfg-denied-client:  129.45.*.*
ds-cfg-denied-client:   192.168.5.6

dn: cn=LDAPS Connection Handler,cn=Connection Handlers,cn=config
objectClass: top
objectClass: ds-cfg-connection-handler
objectClass: ds-cfg-ldap-connection-handler
cn: LDAPS Connection Handler
ds-cfg-java-class: org.opends.server.protocols.ldap.LDAPConnectionHandler
ds-cfg-enabled: true
ds-cfg-listen-address: 0.0.0.0
ds-cfg-listen-port: 636
```

```
ds-cfg-accept-backlog: 128
ds-cfg-allow-ldap-v2: true
ds-cfg-keep-stats: true
ds-cfg-use-tcp-keep-alive: true
ds-cfg-use-tcp-no-delay: true
ds-cfg-allow-tcp-reuse-address: true
ds-cfg-send-rejection-notice: true
ds-cfg-max-request-size: 5 megabytes
ds-cfg-max-blocked-write-time-limit: 2 minutes
ds-cfg-num-request-handlers: 2
ds-cfg-allow-start-tls: false
ds-cfg-use-ssl: true
ds-cfg-ssl-client-auth-policy: optional
ds-cfg-ssl-cert-nickname: server-cert
ds-cfg-key-manager-provider: cn=JKS,cn=Key Manager Providers,cn=config
ds-cfg-trust-manager-provider: cn=JKS,cn=Trust Manager Providers,cn=config
ds-cfg-allowed-client: .example.com
ds-cfg-allowed-client: foo.*.test.com
ds-cfg-allowed-client: 192.168.6.7/22
```

Use the `dsconfig` command to manage the allowed and denied properties for each connection handler. For example:

```
$ dsconfig -n -X -p 4444 -D "cn=directory manager" -j pwd-file \
  set-connection-handler-prop   --handler-name "LDAPS Connection Handler" \
  --set denied-client:.example.com \
  --set allowed-client:192.168.1.6/17
```

**Note:** Denied rules are applied before the allowed rules.

## 20.13 Configuring Unlimited Strength Cryptography

To configure unlimited strength cryptography, you must download the Java Cryptography Extension Unlimited Strength Jurisdiction policy files for missing cryptography support. Perform the following steps to download and install the policy file for configuring unlimited strength cryptography:

1. Download the Java Cryptography Extension Unlimited Strength Jurisdiction policy files from the following Web page

   http://www.oracle.com/technetwork/java/javase/downloads/index.html

2. Perform the installation instructions described in the README.txt file that is part of the downloaded zip.

   Java Cryptography Extension Unlimited Strength Jurisdiction policy files are now installed.

3. Stop the Oracle Unified Directory server, and then restart.

# 21

# Configuring Security Between the Proxy and the Data Source

Security configuration between the proxy and the remote LDAP servers can be configured as follows:

- During installation of the proxy by using the `oud-proxy-setup` GUI. For more information, see "Setting Up the Proxy Server by Using the GUI" in *Oracle Fusion Middleware Installation Guide for Oracle Unified Directory*.
- After the proxy installation, by using the `dsconfig` command in interactive mode. For general information about using the `dsconfig` command, see Section 14.1, "Managing the Server Configuration With `dsconfig`".

For security management, network groups can be enabled to classify incoming client connections. You can use network groups to restrict operations that can be performed, based on how the connection has been classified. Use this functionality, for example, to restrict access to clients that connect from a specified IP address only. For more information, see Section 14.1.6, "Configuring Network Groups With `dsconfig`".

For secure client authentication between the proxy and remote LDAP servers, the certificate of the proxy must be imported into the truststore of each remote LDAP server. In this case, you must configure a keystore manually. For details, see Section 20.2, "Configuring Key Manager Providers".

The proxy security does not bypass the back-end ACI.

This chapter covers the following topics:

- Section 21.1, "How the Proxy Manages Secure Connections"
- Section 21.2, "Modes of Secure Connection"
- Section 21.3, "Configuring Security Between the Proxy and Data Source Using `dsconfig`"

## 21.1 How the Proxy Manages Secure Connections

The proxy manages the security with the client and with the directory server, and supports both SSL and StartTLS.

When you configure security, you must specify how the proxy connects to the remote LDAP server by indicating if the proxy should use SSL `always`, `never`, or `user`. If you specify `always`, the connection with the remote LDAP server will always be secured using SSL, regardless of how the client connects to the proxy. If you specify `never`, the connection between the proxy and the remote LDAP directory server will not be secured, regardless of whether the client connects to the proxy with a secure

connection. If specify `user`, the security between the proxy and the remote LDAP directory servers will be the same as the security between the client and the proxy. For example, if the client connects over SSL, the connection with the remote LDAP server will also use SSL. One notable exception is if the client connects using StartTLS, in which case the proxy will connect to the remote LDAP servers using SSL.

For more information see Modes of Secure Connection.

## 21.2 Modes of Secure Connection

The proxy handles connections to the remote LDAP servers in three SSL security modes:

- `always`
- `never`
- `user`

You can view or edit these settings using the `dsconfig --advanced` command. Choose Extension from the main menu.

The `remote-ldap-server-ssl-policy` property manages the three SSL security modes.

When the `remote-ldap-server-ssl-policy` property is set to `always` or user, the proxy needs to trust the remote LDAP servers. To achieve this, you need to manually import the certificates of each remote LDAP server into the proxy's truststore.

### 21.2.1 The `always` Secure Mode

With the `remote-ldap-server-ssl-policy` property set to `always`, all connections made from the proxy to the remote LDAP servers are fully secure SSL connections, regardless how the client connects to the proxy.

In this mode, the pool size refers to one type of connection pool: secure LDAPS connections.

In the `always` secure mode, the certificate of each remote LDAP server must be imported into the proxy's truststore. If there is a large number of back-end LDAP servers that are not Oracle Unified Directory servers, and if certificates were not managed during installation, importing certificates into the truststore of the proxy can be a constraint. For test environment purposes, you can speed up this process by using the `ssl-trust-all` parameter. This parameter requests the proxy to trust all remote LDAP servers.

### 21.2.2 The `never` Secure Mode

With the `remote-ldap-server-ssl-policy` property set to `never`, none of the connections from the proxy to the remote LDAP servers are secure SSL connections.

In this mode, the monitoring connection by the proxy of the remote LDAP servers is never secure.

In this mode, the pool size refers to one type of connection pool: unsecure LDAP connections.

### 21.2.3 The `user` Secure Mode

With the `remote-ldap-server-ssl-policy` property set to `user`, incoming requests from clients to the proxy dictate whether the connection between the proxy and remote LDAP servers should be secure, regardless of how the client connects to the proxy.

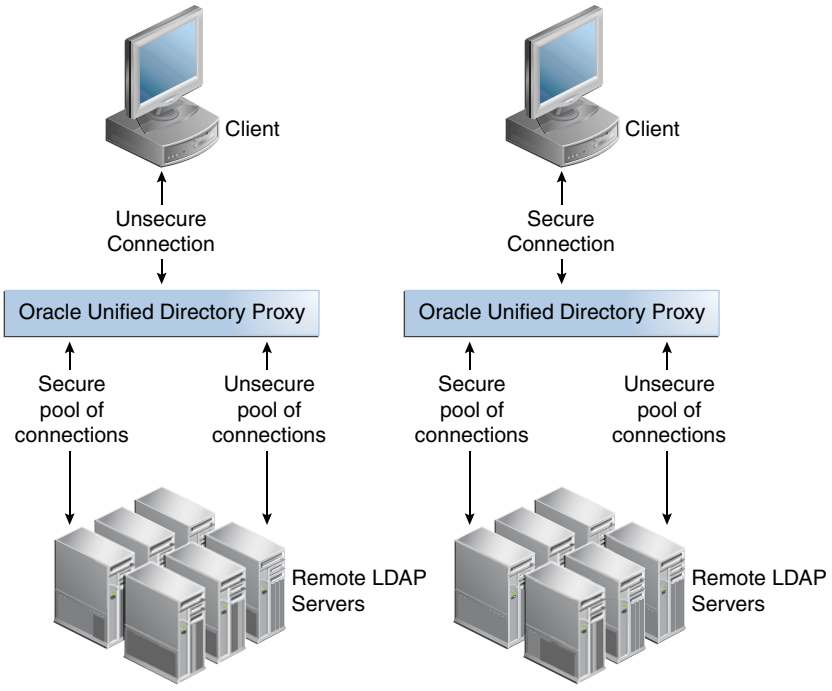
If the incoming client request is secure, whether SSL or StartTLS, the connection from the proxy to the remote LDAP servers is a secure SSL connection.

If the incoming client request is not secure, the connection from the proxy to the remote LDAP servers is not a secure SSL connection.

In this mode, the monitoring connection between the proxy and the remote LDAP servers is never secure.

Two pools of connections are created, one secure and one unsecure. This is shown in Figure 21–1. In the scenario on the left, the client connects to the proxy using an unsecure connection, and the unsecure pool of connections from the proxy to the remote LDAP servers is used. In the scenario on the right, the client connects to the proxy using a secure connection, whether SSL or StartTLS, and the secure SSL pool of connections from the proxy to the remote LDAP servers is used.

***Figure 21–1 Connections in the `user` Secure Mode***



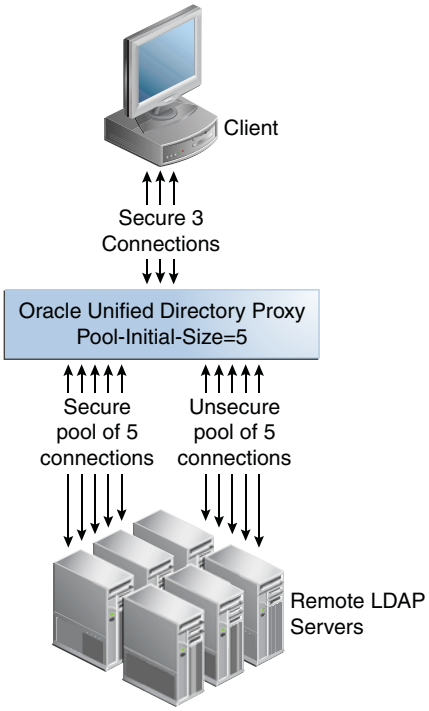

In the `user` mode, the certificate of each remote LDAP server must be imported into the proxy's truststore. If there is a large number of remote LDAP servers that are not Oracle Unified Directory servers, and if certificates were not managed during installation, importing certificates into the truststore of the proxy can be a constraint. In a test environment, you can speed up this process by using the `ssl-trust-all` parameter. This parameter requests the proxy to trust all remote LDAP servers.

When the `remote-ldap-server-ssl-policy` property is set to `user`, the pool size refers to two types of connection pools: unsecure LDAP connections and secure LDAPS connections. If for example the `pool-initial-size` is set to 5 connections, as shown in Figure 21–2, then when the LDAP Extension is initialized, there will be

one pool of 5 LDAP connections and one pool of 5 LDAPS connections, or a total of 10 connections. Each pool evolves separately after this initialization, based on parameters set for that pool.

**Note:** By default, `pool-initial-size` is set to 10 connections.

***Figure 21–2 Multiple Pools of Connections***




## 21.3 Configuring Security Between the Proxy and Data Source Using `dsconfig`

The `dsconfig` tool accesses the server over a secured connection with certificate authentication. If you run `dsconfig` in non-interactive mode, as `dsconfig -n`, specification of the trust store parameters depends on whether you run the command locally or remotely. For more information on running the command locally or remotely, see Overview of the `dsconfig` Command.

### 21.3.1 To Configure Security Between the Proxy and Directory Servers Using `dsconfig`

This task highlights the main steps required to configure security for connections to remote LDAP servers. Where the process is similar to that provided for configuring security between the proxy and the client, pointers are given to the related procedure.

1. If the remote LDAP servers do not require client authentication to be passed from the proxy, proceed directly to step 2.

   If the remote LDAP servers require client authentication to be passed from the proxy, perform the following sub-steps:

   a. Configure a keystore for remote LDAP server connections.

To do this, use the Java keytool command to generate a certificate on the proxy server. The keystore must be configured manually. For details, see Configuring Key Manager Providers.

Self-sign the certificate or have the certificate signed by an external certificate authority. For details, see Configuring Key Manager Providers.

**b.** Configure a key manager provider on the proxy for the keystore for remote LDAP server connections.

For details, see Configuring Key Manager Providers. This key manager provider can be separate to that used for handling secure connections to clients.

**c.** If the remote LDAP servers require client authentication, the certificate of the proxy must be imported into the truststore of each remote LDAP server.

For information about importing and exporting certificates on Oracle Unified Directory, see Configuring Key Manager Providers.

**2.** For the proxy to establish secure connections with the remote LDAP servers, configure a truststore.

All remote LDAP servers requiring a secure connection need to have their certificates imported into the proxy truststore. All of these remote LDAP server certificates can be imported into a single proxy truststore or distributed among multiple proxy truststores. You can have as many proxy truststores as there are remote LDAP server certificates to be imported.

An LDAP proxy extension targeting a secured connection to a remote LDAP data source must reference in its configuration the appropriate truststore manager. This enables the LDAP proxy extension to access the imported remote LDAP server certificate, to accept the secure connection.

**3.** Each truststore requires a proxy trust manager provider.

To list the proxy trust manager providers, use the dsconfig list-trust-manager-providers command. For example:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
list-trust-manager-providers
```

To create a proxy trust manager provider, use the dsconfig create-trust-manager-provider command. For example:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager"  -j pwd-file -X -n \
create-trust-manager-provider \
--provider-name Backend\ Servers \
--type file-based --set enabled:true \
--set trust-store-file:/localhost/config/backend-servers-truststore \
--set trust-store-type:JKS \
--set trust-store-pin-file:/installPath/config/backend-servers-truststore.pin
```

**4.** Import the certificates of the remote LDAP servers into the proxy truststore.

## 21.3.2 Configurable LDAP Extension Properties Relevant to Security

When managing connections to remote LDAP servers using dsconfig, a number of configurable LDAP Extension security connection properties are available. For information about managing LDAP extensions, see Configuring Communication With Remote LDAP Servers. Configurable properties that either directly or indirectly relate to security considerations include the following:

`remote-ldap-server-ssl-policy`
This important value governs the overall security mode of the connections between the proxy and remote LDAP servers. Its use is covered in the section Modes of Secure Connection.

`pool-increment`
If the `remote-ldap-server-ssl-policy` property is set to `user`, two pools of connections are created and the incremental change of size of each pool is set to `pool-increment`. For more information on this property, see To Modify the Advanced Properties of an LDAP Server Extension.

`pool-initial-size`
If the `remote-ldap-server-ssl-policy` property is set to `user`, two pools of connections are created and the initial size, and minimum size, of each pool is set to `pool-initial-size`. In this case, therefore, there will initially be twice the total number of connections indicated in `pool-initial-size`. For details, see To Modify the Advanced Properties of an LDAP Server Extension.

`pool-max-size`
If the `remote-ldap-server-ssl-policy` property is set to `user`, two pools of connections are created and the maximum size of each pool is set to `pool-max-size`.

The default value is 1000 connections. For more information on this property, see To Modify the Advanced Properties of an LDAP Server Extension.

`remote-ldap-server-ssl-port`
The port number for SSL connections from the proxy to the remote LDAP server.

`ssl-client-alias`
When a keystore is created for client authentication, several keys can be stored in it. Use this property to specify which key to use. For more information about keystores, see Getting SSL Up and Running Quickly. See also Configuring Key Manager Providers.

`ssl-key-manager-provider`
Specifies a key manager provider to use for the LDAP Server Extension. The key manager provider is not mandatory and can be used if the remote LDAP server is configured for client authentication. The referenced key manager provider must be enabled. For more information about key manager providers, see Configuring Key Manager Providers.

`ssl-trust-all`
If this parameter is set to `true`, all remote LDAP servers are trusted. The default value is `false`. Setting this value to `true` avoids having to import certificates from remote LDAP servers but is insecure.

Note that although the interactive `dsconfig --advanced` command offers Blind Trust as a possible trust manager provider, Blind Trust is not supported for the proxy server. Instead, if you want to avoid the import of certificates, set the `ssl-trust-all` parameter to true. This presents an insecure deployment and is not recommended for production environments, only for testing purposes.

If the `remote-ldap-server-ssl-policy` is set to `never`, then the value of the `ssl-trust-all` parameter is irrelevant. All connections between the proxy will be insecure (unencrypted) in this case. For more information on the `remote-ldap-server-ssl-policy`, see Modes of Secure Connection.

`ssl-trust-manager-provider`
Specifies which trust manager provider to use for the LDAP Server Extension. The trust manager provider is mandatory unless the `ssl-trust-all` parameter is set to `true`. The referenced trust manager provider must be enabled.

# 22

# Controlling Access To Data

Controlling access to directory contents is an integral part of creating a secure directory service. Access to data is managed with access control instructions (ACIs) that specify the access right to individual entries, all sub-entries below an entry, or all entries on a global basis.

Numerous or complicated ACIs require greater processing resources than a few simple ACIs. You can significantly reduce the performance of your directory by specifying a large number of ACIs or extremely complicated ACIs.

Oracle Unified Directory includes the ability to view the effective rights of a given user for a given entry. This feature simplifies the administration of the complex and powerful access control mechanism.

For an overview of the ACI model, see Chapter 8, "Understanding the Oracle Unified Directory Access Control Model".

The following sections describe how to create ACIs to control access to data:



## 22.1 Managing Global ACIs With `dsconfig`

Global ACIs control access to the root of the DIT instead of to a particular sub-tree. Global ACIs apply to all entries in the directory. You can set, reset, and delete global ACIs with the `dsconfig` command and with the `ldapmodify` command. `dsconfig` accesses the server configuration over SSL, using the administration connector. For more information about `dsconfig`, see Section 14.1, "Managing the Server Configuration With `dsconfig`".

You cannot use `dsconfig` to manage ACIs that are applied to entries in sub-trees. To manage non-global ACIs, see Section 22.2, "Managing ACIs With `ldapmodify`".

### 22.1.1 Default Global ACIs

When you install Oracle Unified Directory, nine default global ACIs are defined. The effect of all the default global ACIs is to allow the following:

- Anyone has read access to certain controls and extended operations.
- Anyone has access to search, compare, and read attributes at the rootDSE level. Certain attributes require explicit access.
- Authenticated users can modify a subset of the attributes in their own entries in the directory. Users are unable to delete their own entries.
- Anyone has access to key operational attributes including many in the root DSE and cn=schema, as well as other attributes that show up in entries throughout the server.

The proxy does not evaluate global ACIs. The proxy forwards LDAP requests to the remote LDAP server, and the remote LDAP server evaluates the ACIs.

### 22.1.2 To Display the Global ACIs

The global ACIs are all values of the global-aci property of the access control handler. You can use dsconfig to display the global ACIs currently configured on the server by viewing the global-aci property.

Run the dsconfig command as follows (output reformatted for readability).

```
$ dsconfig -h localhost -p 4444 -D cn="Directory Manager" -j pwd-file -X -n \
  get-access-control-handler-prop --property global-aci
Property   : Value(s)
-----------:---------------------------------------------------------------------------
global-aci : (extop="1.3.6.1.4.1.26027.1.6.1 || 1.3.6.1.4.1.26027.1.6.3 ||
           :   1.3.6.1.4.1.4203.1.11.1 || 1.3.6.1.4.1.1466.20037 ||
           :   1.3.6.1.4.1.4203.1.11.3") (version 3.0; acl "Anonymous extended operation access";
           :   allow(read) userdn="ldap:///anyone";),
           : "(target="ldap:///")(targetscope="base")(targetattr="objectClass||namingContexts||
           :   supportedAuthPasswordSchemes||supportedControl||supportedExtension||
           :   supportedFeatures||supportedLDAPVersion||supportedSASLMechanisms||vendorName||
           :   vendorVersion")(version 3.0; acl "User-Visible Root DSE Operational Attributes";
           :   allow (read,search,compare) userdn="ldap:///anyone";)",
           : (target="ldap:///cn=changelog")(targetattr="*")(version 3.0; acl "External changelog
           :   access"; deny (all) userdn="ldap:///anyone";),
           : "(target="ldap:///cn=schema")(targetscope="base")(targetattr="objectClass||
           :   attributeTypes||dITContentRules||dITStructureRules||ldapSyntaxes||matchingRules||
           :   matchingRuleUse||nameForms||objectClasses")(version 3.0; acl "User-Visible Schema
           :   Operational Attributes"; allow (read,search,compare) userdn="ldap:///anyone";)",
           : (target="ldap:///dc=replicationchanges")(targetattr="*")(version3.0; acl
           :   "Replication backend access"; deny (all) userdn="ldap:///anyone";),
           : (targetattr="audio||authPassword||description||displayName||givenName||homePhone||
           :   homePostalAddress||initials||jpegPhoto||labeledURI||mobile||pager||postalAddress||
           :   postalCode||preferredLanguage||telephoneNumber||userPassword")(version 3.0; acl
           :   "Self entry modification"; allow (write) userdn="ldap:///self";),
           : "(targetattr="createTimestamp||creatorsName||modifiersName||modifyTimestamp||entryDN||
           :   entryUUID||subschemaSubentry||orclguid")(version 3.0; acl "User-Visible Operational
           :   Attributes"; allow (read,search,compare) userdn="ldap:///anyone";)",
           : "(targetattr="userPassword||authPassword")(version 3.0; acl "Self entry read";
           :   allow (read,search,compare) userdn="ldap:///self";)",
           : (targetcontrol="1.3.6.1.1.12 || 1.3.6.1.1.13.1 || 1.3.6.1.1.13.2||
           :   1.2.840.113556.1.4.319 || 1.2.826.0.1.3344810.2.3 || 2.16.840.1.113730.3.4.18 ||
           :   2.16.840.1.113730.3.4.9 ||1.2.840.113556.1.4.473 || 1.3.6.1.4.1.42.2.27.9.5.9")
           :  (version 3.0; acl "Authenticated users control access"; allow(read)
```

```
:   userdn="ldap:///all";), (targetcontrol="2.16.840.1.113730.3.4.2 ||
:   2.16.840.1.113730.3.4.17 || 2.16.840.1.113730.3.4.19 ||1.3.6.1.4.1.4203.1.10.2 ||
:   1.3.6.1.4.1.42.2.27.8.5.1 ||2.16.840.1.113730.3.4.16 || 2.16.840.1.113894.1.8.31")
:  (version 3.0; acl "Anonymous control access"; allow(read) userdn="ldap:///anyone";)
```

### 22.1.3 To Delete a Global ACI

The easiest way to delete a global ACI is to use `dsconfig` in interactive mode. Interactive mode walks you through the ACI configuration, and is therefore not documented here. If you delete global ACIs in non-interactive mode, make sure that you escape all special characters in the ACI specification as required by your command line shell.

This example deletes the global ACI that allows anonymous access by using `dsconfig` in non-interactive mode.

Run the `dsconfig` command as follows.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  set-access-control-handler-prop \
  --remove global-aci:\(targetattr!=\"userPassword\|\|authPassword\"\) \
  \(version\ 3.0\;\ acl\ \"Anonymous\ read\ access\"\;\ allow\(read,search,compare\) \
  \ userdn=\"ldap:///anyone\"\;\)
```

### 22.1.4 To Add a Global ACI

When you add a global ACI, make sure that you escape all special characters in the ACI specification as required by your command-line shell.

The following example adds the global ACI that was removed in the previous procedure, using `dsconfig` in non-interactive mode:

Run the `dsconfig` command as follows.

```
$ dsconfig -h localhost -p 4444 -D cn="Directory Manager" -j pwd-file -n \
  set-access-control-handler-prop \
  --add global-aci:\(targetattr!=\"userPassword\|\|authPassword\"\) \
  \(version\ 3.0\;\ acl\ \"Anonymous\ read\ access\"\;\ allow\
\(read,search,compare\)
  \ userdn=\"ldap:///anyone\"\;\)
```

## 22.2 Managing ACIs With `ldapmodify`

You can create access control instructions (ACIs) manually using LDIF statements, and add them to your directory by using the `ldapmodify` command. Because ACI values can be very complex, it is useful to view existing values and copy them to help create new ones.

For additional sample ACIs to the ones illustrated here, see Section 22.5, "Access Control Usage Examples".

### 22.2.1 To View ACI Attribute Values

ACIs are stored as one or more values of the `aci` attribute on an entry. The `aci` attribute is a multivalued operational attribute that can be read and modified by directory users, and should itself be protected by ACIs.

Administrative users are usually given full access to the `aci` attribute.

View the values of the `aci` attribute by running the following `ldapsearch` command:

```
$ ldapsearch -h host -p port -D "cn=Directory Manager" -j pwd-file \
  -b entryDN -s base "(objectclass=*)" aci
```

The result is LDIF text that you can copy into a new LDIF ACI definition for editing. Because the value of an ACI is a long string, the output from the `ldapsearch` operation is likely to be displayed over several lines, with the first space being a continuation marker. Take this into account when copying and pasting the LDIF output.

To view the effect of an ACI value, in terms of the permissions that it grants or denies, see Section 22.7, "Viewing Effective Rights".

## 22.2.2 To Add an ACI

You can add an ACI by specifying the ACI in an LDIF file and then applying the LDIF file with the `ldapmodify` command. The LDIF file must contain one or more `aci` attributes, each of which is composed of the `aci:` prefix followed by the ACI specification. For more information, see Section 8.2, "ACI Syntax".

1. Create the ACI in an LDIF file.

   The following sample LDIF file (`aci.ldif`) adds an ACI that grants a particular user (`csmith`) full access rights to the directory:

   ```
   dn: ou=people,dc=example,dc=com
   changetype: modify
   add: aci
   aci: (targetattr="*")(version 3.0; acl "give csmith full rights"; allow(all)
   userdn =  "ldap:///uid=csmith,ou=People,dc=example,dc=com";)
   ```

2. Use the `ldapmodify` command to apply the ACI to the directory.

   The following command applies the ACI contained in the `aci.ldif` file to the directory:

   ```
   $ ldapmodify -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
     --filename aci.ldif
   Processing MODIFY request for ou=people,dc=example,dc=com
   MODIFY operation successful for DN ou=people,dc=example,dc=com
   ```

## 22.2.3 To Remove an ACI

You can remove an ACI by specifying its value in an LDIF file, and then removing the value with the `ldapmodify` command.

1. Remove the ACI in an LDIF file.

   The following sample LDIF file (`remove-aci.ldif`) removes the ACI that was added in the previous procedure:

   ```
   dn: ou=people,dc=example,dc=com
   changetype: modify
   delete: aci
   aci: (targetattr="*")(version 3.0; acl "give csmith full rights"; allow(all)
   userdn =  "ldap:///uid=csmith,ou=People,dc=example,dc=com";)
   ```

2. Use the `ldapmodify` command to apply the change to the directory.

   The following command applies the changes contained in the `remove-aci.ldif` file to the directory:

   ```
   $ ldapmodify -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
   --filename remove-aci.ldif
   ```

```
Processing MODIFY request for ou=people,dc=example,dc=com
MODIFY operation successful for DN ou=people,dc=example,dc=com
```

## 22.3 Managing Access Control With Oracle Directory Services Manager

You can use ODSM to view the existing ACIs that are configured in the server, to create new access control points, and to create new ACIs in a user-friendly interface. The following topics described how to manage access control by using ODSM.

### 22.3.1 Display the Configured ACIs

Oracle Unified Directory supports several preconfigured ACIs, by default. You can display all ACIs that are configured in the server by using ODSM, as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager".
2. Select the **Security** tab.
3. Expand the **Directory ACLs** element.
4. All configured ACIs are listed under the access control point in which the ACI is defined. Expand the access control point to view the ACIs. For example, to display the list of ACIs that apply to the Root entry, expand the Root entry.
5. Select an ACI to view its properties in the right hand pane.

### 22.3.2 Create an Access Control Point

An access control point is the entry in which an ACI is defined (in other words, the entry that contains the corresponding `aci` attribute.

You can define a new access control point by using ODSM, as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager".
2. Select the **Security** tab.
3. Expand the **Directory ACLs** element.
4. Click the **Create** icon.
5. In the **Location** field, enter the DN of the entry that will be the new access control point, or click **Select** to select the entry from the directory.
6. To add one or more ACIs to the access control point, click **Create ACI**.
7. Enter the ACI details. For more information about these fields, see Add an ACI.
8. When you have added the required ACIs to the access control point, click **Create**.

### 22.3.3 Create an Access Control Point Based on an Existing Access Control Point

You can define a new access control point that is based on an existing access control point by using ODSM, as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager".
2. Select the **Security** tab.
3. Expand the **Directory ACLs** element.

4. Select the access control point on which you want to base the new access control point.
5. Click the **Create like** icon.
6. In the **Location** field, enter the DN of the entry that will be the new access control point, or click **Select** to select the entry from the directory.
7. The new access control point is automatically created with the same ACL as the access control point on which it was based.
8. To add, remove, or edit the existing ACIs on the new access control point, click **Create**, **Edit** or **Delete**.
9. To add or edit an ACI, enter the required details. For more information about these fields, see Section 22.3.5, "Add an ACI".
10. When you have modified the ACIs for the new access control point, click **Create**.

### 22.3.4 Delete an Access Control Point

You can delete an access control point by using ODSM, as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager".
2. Select the **Security** tab.
3. Expand the **Directory ACLs** element.
4. Select the access control point that you want to delete and click the **Delete** icon.
5. Click **OK** to confirm the deletion.

### 22.3.5 Add an ACI

You can add an ACI to an existing access control point, by using ODSM as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager".
2. Select the **Security** tab.
3. Expand the **Directory ACLs** element.
4. Expand the access control point to which you want to add the new ACI.
5. Select one of the ACIs in the access control list.
6. Click the **Add** icon.
7. To build the ACI in a user friendly interface, select the **Detail View** tab.
8. Select the **Scope** of the ACI.

   Usually an ACI has subtree scope. You can restrict the scope of the ACI by selecting one of the following values:

   - **Base.** The ACI applies to the target resource only.
   - **One.** The ACI applies to the target resource's first-generation children.
   - **Subtree.** The ACI applies to the target resource and the subtree below it.
   - **Subordinate.** The ACI applies only to the subtree below the target resource.
9. In the **Targets** field, select each element of the ACI and click **Edit** to define its properties.

For more information about defining ACI targets, see Section 8.2.2, "Defining Targets".

10. In the **Permissions** field, click the **Add** icon to define permissions and bind rules.

    For more information about defining ACI permissions, see Section 8.2.3, "Defining Permissions".

    For more information about defining bind rules, see Section 8.3, "Bind Rules."

    Perform the following steps to define the bind rules:

    a. From **Bind Rule Type** list, select the desired bind rule.

    b. Click the **User Attribute** tab to create user attribute bind rule.

       Perform the following steps:

       – For **User Attribute Operator** property, select the desired value.

       – For **Entry Selection** property, select **Target Entry and its Subtree.**

       – From the **Inheritance Levels** list, select the desired inheritance level value.

       – In the **User Attribute** field, enter an attribute or alternatively click **Select** to search an entry.

       – For **User Attribute Type** property, click **Bind Type Format.**

       – From the **Bind Type Value** list, select the bind type value.

       – Click **OK.**

11. If you would rather define the ACI manually, click the **Text Editor View** tab and enter the details of the ACI.

    Click **Validate** to check that the ACI conforms to the ACI syntax.

    You can also use this view to copy and paste existing ACIs.

12. When you have completed the ACI definition, click *Create*.

### 22.3.6 Add an ACI Based on an Existing ACI

You can add an ACI that is based on an existing ACI, by using ODSM as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager".
2. Select the **Security** tab.
3. Expand the **Directory ACLs** element.
4. Expand the access control point that contains the ACI that you want to copy.
5. Select the ACI that you want to copy.
6. Click the **Add like** icon.
7. Edit the elements of the ACI that you want to change, either in **Text Editor View** or in **Detail View**.
8. When you have completed the ACI definition, click **Create**.

### 22.3.7 Modify an ACI

You can modify an existing ACI, by using ODSM as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager".
2. Select the **Security** tab.
3. Expand the **Directory ACLs** element.
4. Expand the access control point that contains the ACI that you want to change
5. Select the ACI that you want to change.
6. Edit the elements of the ACI, either in **Text Editor View** or in **Detail View**.
7. When you have completed your changes, click **Apply**.

## 22.4 Managing Macro ACIs With Oracle Directory Services Manager

You can use ODSM to enter macro expressions in `target, targetFilter, userDn, groupDN,` and `userAttr` attributes. For more information about Macro ACIs, see Section 8.6, "Using Macro ACIs for Advanced Access Control."

This section contains the following topics:

- Section 22.4.1, "Editing a Target"
- Section 22.4.2, "Editing a Target Filter"
- Section 22.4.3, "Editing Bind Rules for User DN or Group DN"
- Section 22.4.4, "Editing Bind Rules for User Attributes"

### 22.4.1 Editing a Target

This section describes how to enter a macro ACI in the target.

To edit a target to enter a macro ACI:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager."
2. Select the **Security** tab.
3. From the **Directory ACLs** list, select the ACI that you want to edit.
4. From the Targets table, select the **Target** row.
5. Click **Edit.**
6. In the **Target** field, enter the macro expression.
7. Click **OK.**

### 22.4.2 Editing a Target Filter

This section describes how to enter a macro ACI in the target filter.

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager."
2. Select the **Security** tab.
3. From the **Directory ACLs** list, select the ACI that you want to edit.
4. From the Targets table, select the **Target Filter** row**.**
5. Click **Edit.**

6. In the **Target** field, enter the filter with the macro expression.
7. Click **OK.**

### 22.4.3 Editing Bind Rules for User DN or Group DN

This section describes how to define access for a targeted resource to specific user or a specific group.

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager."
2. Select the **Security** tab.
3. From the **Directory ACLs** list, select the ACI that you want to edit.
4. From the Permissions table, select the **Bind Rules** row.
5. Click **Edit.**
6. From Bind Rule Type list, select the desired bind rule.
7. On the **Access To** tab, from the **User DN** list, select **Specify Users.**
8. Perform the following steps in the User table:
   a. Click **Add.**
   b. Enter the macro expression to define user access or alternatively click **Select** to search the entry and add the macro expression in the selected entry.
9. Perform the following steps to specify access to a specific group for a targeted resource in the Group DN Operator table:
   a. Click **Add.**
   b. Enter the macro expression to define group access or alternatively click **Select** to search the entry and add the macro expression in the selected entry.
10. Click **OK.**

---

**Note:** You can edit an individual bind rule as well. You need to select the required bind rule in the Permissions table, and then click **Edit** for modifying the bind rule.

---

### 22.4.4 Editing Bind Rules for User Attributes

The section describes how to edit bind rules for a user attribute.

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager."
2. Select the **Security** tab.
3. From the **Directory ACLs** list, select the ACI that you want to edit.
4. From the Permissions table, select the **Bind Rules** row.
5. Click **Edit.**
6. Click the **User Attribute** tab.
7. From Bind Rule Type list, select the desired bind rule.
8. For Entry Selection property, select **Specific Entry.**

9. In the Entry Base DN field, enter the base DN or alternatively click **Select** to search an entry and add the macro expression in the selected entry.
10. In the User Attribute field, enter an attribute name or alternatively click **Select** to search an attribute name from the list of attribute names.
11. Click **Bind Type Format.**
12. From the Bind Type Value list, select the desired value.
13. Click **OK.**

**Note:** You can edit an individual bind rule as well. You need to select the required bind rule in the Permissions table, and then click **Edit** for modifying the bind rule.

## 22.5 Access Control Usage Examples

This section provides several sample ACIs that can be used to implement an access control policy.

### 22.5.1 Disabling Anonymous Access

The directory server allows anonymous access by default. There might be situations in which you want to disable anonymous access, particularly to sensitive data within your directory.

The following default ACI allows anonymous read access to all user attributes except for the `userpassword` and `authPassword` attributes:

```
aci: (targetattr!="userPassword||authPassword")(version 3.0; acl "
Anonymous read access"; allow (read,search,compare) userdn="ldap:///anyone";)
```

To disable anonymous access, remove this ACI from the default access control handler, as shown in the following example:

```
$ dsconfig -h localhost -p 4444 -D cn="Directory Manager" -j pwd-file -n \
  set-access-control-handler-prop \
  --remove global-aci:'(targetattr!="userPassword||authPassword") \
  (version 3.0; acl "Anonymous read access"; \
  allow (read,search,compare) userdn="ldap:///anyone";)'
```

Depending on your shell, you might need to escape any quotations in the ACI itself.

### 22.5.2 Granting Write Access to Personal Entries

The default global ACIs allow write access to a limited subset of the attributes of a user's own entry. These attributes include the following:

- `audio`
- `authPassword`
- `description`
- `displayName`
- `givenName`
- `homePhone`
- `homePostalAddress`

- initials
- jpegPhoto
- labeledURI
- mobile
- pager
- postalAddress
- postalCode
- preferredLanguage
- telephoneNumber
- userPassword

Use this procedures in this section to grant users write access to additional attributes of their own entries.

#### 22.5.2.1 Granting Write Access Based on DNS

The following example ACI enables users internal to example.com to change their own business category and room number.

Remember, by allowing write access, you also grant users the right to delete attribute values.

```
aci: (targetattr="businessCategory || roomNumber")
(version 3.0; acl "Write example.com"; allow (write)
userdn="ldap:///self" and dns="*.example.com";)
```

This example assumes that the ACI is added to the ou=People,dc=example,dc=com entry.

#### 22.5.2.2 Granting Write Access Based on Authentication Method

The following example enables any user to update all of his own personal information in the example.com tree provided that he establish an SSL connection to the directory.

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

```
aci: (targetattr="*")
(version 3.0; acl "Write SSL"; allow (write)
userdn= "ldap:///self" and authmethod="ssl";)
```

This example assumes that the aci is added to the ou=subscribers,dc=example,dc=com entry.

### 22.5.3 Granting a Group Full Access to a Suffix

Most directories have a group that is used to identify certain corporate functions. These groups can be given full access to all or part of the directory. By applying the access rights to the group, you can avoid setting the access rights for each member individually. Instead, you grant users these access rights by adding them to the group.

The following sample ACI allows a group named the HRgroup full access to the ou=People branch of the directory so that they can update employee information:

```
aci: (targetattr="*") (version 3.0; acl "HR"; allow (all)
```

```
groupdn= "ldap:///cn=HRgroup,ou=People,dc=example,dc=com";)
```

This example assumes that the ACI is added to the `ou=People,dc=example,dc=com` entry.

## 22.5.4 Granting Rights to Add and Delete Group Entries

Some organizations want to allow employees to create entries in the tree if it can increase their efficiency, or if it can contribute to the corporate dynamics. The following examples assume that `example.com` has a social committee that is organized into various clubs (tennis, swimming, skiing, and so on).

### 22.5.4.1 Creating a `"Create Group"` ACI

This sample ACI allows any `example.com` employee to create a group entry representing a new club, under the `ou=social committee` branch.

```
aci: (target = "ldap:///dc=ou=social committee,dc=example,dc=com")
(targetfilter="(|(objectClass=groupOfNames)(objectClass=top))")
(version 3.0; acl"Create Group"; allow (search,read,add) (userdn = @
"ldap:///uid=*,ou=People,dc=example,dc=com" and dns = "*.example.com");)
```

This example assumes that the ACI is added to the `ou=social committee, dc=example,dc=com` entry.

> **Note:** This ACI does not grant write permission, which means that the entry creator cannot modify the entry. Because the server adds the value `top` behind the scenes, you must specify `objectClass=top` in the `targattrfilters`.

### 22.5.4.2 Creating a `"Delete Group"` ACI

This sample ACI ensures that only the group owner can modify or delete a group entry under the ou=Social Committee branch.

```
aci: (target="ou=social committee,dc=example,dc=com")
(targetattr = "*")
(targattrfilters="del=objectClass:(objectClass=groupOfNames)")
(version 3.0; acl "Delete Group"; allow (write,delete)
userattr="owner#GROUPDN";)
```

This example assumes that the ACI is added to the `ou=social committee,dc=example,dc=com` entry.

## 22.5.5 Allowing Users to Add or Remove Themselves From a Group

Many directories set ACIs that allow users to add or remove themselves from groups. This is useful, for example, for allowing users to add and remove themselves from mailing lists. The following sample ACI enables all employees to add themselves to any group entry under the `ou=social committee` subtree:

```
aci: (targettattr="member")(version 3.0; acl "Group Members";
allow (selfwrite)
(userdn= "ldap:///uid=*,ou=People,dc=example,dc=com") ;)
```

This example assumes that the ACI is added to the `ou=social committee, dc=example,dc=com` entry.

### 22.5.6 Granting Conditional Access to a Group

In many cases, when you grant a group privileged access to the directory, you want to ensure that those privileges are protected from intruders trying to impersonate the privileged users. Therefore, in many cases, access control rules that grant critical access to a group or role are often associated with a number of conditions.

The following sample ACI grants the Directory Administrators group full access to the corporate clients branch of the directory tree, provided the following conditions are fulfilled:

- The connection is authenticated using a certificate over SSL
- Access is requested between 08:00 and 18:00, Monday through Thursday
- Access is requested from a specified IP address

```
aci: (target="ou=corporate-clients,dc=example,dc=com")
(targetattr = "*") (version 3.0; acl "corporate-clients"; allow (all)
(groupdn="ldap:///cn=DirectoryAdmin,ou=corporate-clients,dc=example,dc=com")
and (authmethod="ssl") and (dayofweek="Mon,Tue,Wed,Thu") and
(timeofday >= "0800" and timeofday <= "1800") and (ip="255.255.123.234"); )
```

This example assumes that the ACI is added to the `ou=corporate-clients,dc=example,dc=com` entry.

### 22.5.7 Denying Access

If your directory holds business-critical information, you might specifically want to deny access to it. The following sample ACIs allow users to read certain "billing information", such as connection time and account balance, under their own entries, but prohibits them from changing this information.

This ACI allows users to read the information. The example assumes that the relevant attributes have been created in the schema.

```
aci: (targetattr="connectionTime || accountBalance")
(version 3.0; acl "Billing Info Read"; allow (search,read)
userdn="ldap:///self";)
```

This ACI prevents users from changing the information. The example assumes that the relevant attributes have been created in the schema.

```
aci: (targetattr="connectionTime || accountBalance")
(version 3.0; acl "Billing Info Deny";
deny (write) userdn="ldap:///self";)
```

### 22.5.8 Defining Permissions for DNs That Contain a Comma

DNs that contain commas require special treatment within LDIF ACI statements. In the target and bind rule portions of the ACI statement, commas must be escaped by a single backslash (\). The following example illustrates this syntax:

```
dn: o=example.com Bolivia\, S.A.
objectClass: top
objectClass: organization
aci: (target="ldap:///o=example.com Bolivia\,S.A.")
(targetattr="*") (version 3.0; acl "aci 2"; allow (all)
groupdn = "ldap:///cn=Directory Administrators,
o=example.com Bolivia\, S.A.";)
```

## 22.6 Proxy Authorization ACIs

The proxy authorization method is a special form of authentication: a user that binds to the directory using his own identity is granted the rights of another user, through proxy authorization.

This example makes the following assumptions:

- The client application's bind DN is `uid=MoneyWizAcctSoftware,ou=Applications,dc=example,dc=com`.
- The targeted subtree to which the client application is requesting access is `ou=Accounting,dc=example,dc=com`.
- An Accounting Administrator with access permissions to the `ou=Accounting,dc=example,dc=com` subtree exists in the directory.

For the client application to gain access to the Accounting subtree (using the same access permissions as the Accounting Administrator), the application requires the following rights and controls:

- The Accounting Administrator must have access permissions to the `ou=Accounting,dc=example,dc=com` subtree. The following ACI grants all rights to the Accounting Administrator entry:

```
aci: (target="ldap:///ou=Accounting,dc=example,dc=com")
(targetattr="*") (version 3.0; acl "allow All-AcctAdmin"; allow
(all) userdn="ldap:///uid=AcctAdministrator,ou=Administrators,
dc=example,dc=com";)
```

- The client application must have proxy rights. The following ACI grants proxy rights to the client application:

```
aci: (target="ldap:///ou=Accounting,dc=example,dc=com")
(targetattr="*") (version 3.0; acl "allow proxy-
accounting software"; allow (proxy) userdn=
"ldap:///uid=MoneyWizAcctSoftware,ou=Applications,
dc=example,dc=com";)
```

- The client application must be allowed to use the proxy authorization control. The following ACI allows the client application to use the proxy authorization control:

```
aci: (targetcontrol = "2.16.840.1.113730.3.4.18")
(version 3.0; acl "allow proxy auth - accounting software";
allow (all) userdn="ldap:///uid=MoneyWizAcctSoftware,ou=Applications,
dc=example,dc=com";)
```

With these ACIs in place, the `MoneyWizAcctSoftware` client application can bind to the directory and send an LDAP command such as `ldapsearch` or `ldapmodify` that requires the access rights of the proxy DN.

In the previous example, if the client wanted to perform an `ldapsearch` command, the command would include the following controls:

```
$ ldapsearch -D "uid=MoneyWizAcctSoftware,ou=Applications,dc=example,dc=com" \
  -j pwd-file -Y "dn:uid=AcctAdministrator,ou=Administrators,dc=example,dc=com" \
  -b "ou=Accounting,dc=example,dc=com"  "objectclass=*"\
...
```

The base of the search must match the target of the ACIs. The client binds as itself but is granted the privileges of the proxy entry. The client does not need the password of the proxy entry.

For more information, see Section 17.5.3.13, "Searching Using the Proxied Authorization Control".

## 22.7 Viewing Effective Rights

When you maintain the access control policy on the entries of a directory, it is useful to know the effects on security of the ACIs that you define. The directory server enables you to evaluate existing ACIs and report the effective rights that they grant for a given user on a given entry.

### 22.7.1 The Get Effective Rights Control

The directory server responds to the Get Effective Rights control, which can be included in a search operation. The response to this control is to return the effective rights information about the entries and attributes in the search results. This extra information includes read and write permissions for each entry and for each attribute in each entry. The permissions may be requested for the bind DN used for the search or for an arbitrary DN, allowing administrators to test the permissions of directory users.

Effective rights functionality relies on an LDAP control. To view the effective rights when going through a proxy server, you must enable this control in the proxy chaining policy. You must also ensure that the proxy identity used to bind to the remote server is also allowed to access the effective rights attributes.

### 22.7.2 Using the Get Effective Rights Control

The behavior of the Get Effective Rights Control differs from the Internet draft Get Effective Rights Control (`http://tools.ietf.org/html/draft-ietf-ldapext-acl-model-08`) in the following ways:

- There is no response control returned with the search results. Instead, the rights information is added to the result entries. Also, the format of the rights information is completely different from the draft and is described below.
- The request control only takes an `authzid`.

There are two ways to specify the Get Effective Rights control with the `ldapsearch` command:

1. Use the `-J "1.3.6.1.4.1.42.2.27.9.5.2"` option or simply `-J effectiverights`. If you specify a NULL value for the Get Effective Rights Control's `authzid` value, the bind user is used as the `authzid` and the rights for the attributes and entries being returned with the current `ldapsearch` operation are retrieved.
2. The simpler and preferred method is to use the `-g` option with or without the `-e` option:
   - `-g "dn: DN`"--The search results will show the effective rights of the user binding with the given *DN*. This option allows an administrator to check the effective rights of another user. The option `-g "dn:"` will show the effective rights for anonymous authentication.
   - `-e attributeName1 -e attributeName2` --The search results will also include the effective rights on the named attributes. This option can be used to specify attributes that would not appear in the search results for the entry. For

example, this option can be used to determine if a user has permission to add an attribute that does not currently exist in an entry.

---

**Note:** The `-e` option requires the `-g` option and should not be used with the `-J` option.

If you use the `-g` option, do not use the `-J` option with the OID of the Get Effective Rights control.

---

Besides using one of these two ways to specify the Get Effective Rights Control, you must specify the type of information you want to view, either the simple rights or the more detailed logging information that explains how those rights are granted or denied. The type of information is determined by adding either `aclRights` or `aclRightsInfo`, respectively, as an attribute to return in the search results. You can request both attributes to receive all effective rights information, although the simple rights are redundant with the information in the detailed logging information.

---

**Note:** The `aclRights` and `aclRightsInfo` attributes have the behavior of virtual operational attributes. They are not stored in the directory, and they will not be returned unless explicitly requested. These attributes are generated by the directory server in response to the Get Effective Rights Control. For this reason, neither of these attributes can be used in filters or search operations of any kind.

---

The effective rights feature inherits other parameters that affect access control (such as time of day, authentication method, machine address, and machine name) from the user initiating the search operation.

The following example shows how a user, Carla Fuente, can view her rights in the directory. In the results, a `1` means that permission is granted, and a `0` means that permission is denied.

```
$ ldapsearch -J effectiverights -h rousseau.example.com -p 1389 \
  -D "uid=cfuente,ou=People,dc=example,dc=com" -j pwd-file \
  -b "dc=example,dc=com" "(objectclass=*)" aclRights
dn: dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
dn: ou=Groups, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
dn: ou=People, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
dn: cn=Accounting Managers,ou=groups,dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
dn: cn=HR Managers,ou=groups,dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
dn: uid=bjensen,ou=People, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
dn: uid=cfuente, ou=People, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:1,proxy:0
```

This result shows Carla Fuente the entries in the directory where she has at least read permission and that she can modify her own entry. The effective rights control does not bypass normal access permissions, so a user will never see the entries for which they do not have read permission. In the following example, the Directory Manager can see the entries to which Carla Fuente does not have read permission:

```
$ ldapsearch -h rousseau.example.com -p 1389 -D "cn=Directory Manager" \
  -j pwd-file -g "dn: uid=cfuente,ou=People,dc=example,dc=com" \
  -b "dc=example,dc=com" "(objectclass=*)" aclRights
dn: dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
dn: ou=Groups, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
dn: cn=Directory Administrators, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:0,write:0,proxy:0
dn: ou=Special Users,dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:0,write:0,proxy:0
dn: ou=People, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
dn: cn=Accounting Managers,ou=groups,dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
dn: cn=HR Managers,ou=groups,dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
dn: uid=bjensen,ou=People, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
dn: uid=cfuente, ou=People, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:1,proxy:0
```

In the output above, the directory manager can see that Carla Fuente cannot even view the Special Users nor the Directory Administrators branches of the directory tree. In the following example, the Directory Administrator can see that Carla Fuente cannot modify the `mail` and `manager` attributes in her own entry:

```
$ ldapsearch -h rousseau.example.com -p 1389 -D "cn=Directory Manager" \
  -j pwd-file -g "dn: uid=cfuente,ou=People,dc=example,dc=com" \
  -b "dc=example,dc=com" "(uid=cfuente)" aclRights "*"
version: 1
dn: uid=cfuente, ou=People, dc=example,dc=com
aclRights;attributeLevel;mail: search:1,read:1,compare:1,
write:0,selfwrite_add:0,selfwrite_delete:0,proxy:0
mail: cfuente@example.com
aclRights;attributeLevel;uid: search:1,read:1,compare:1,
write:1,selfwrite_add:1,selfwrite_delete:1,proxy:0
uid: cfuente
aclRights;attributeLevel;givenName: search:1,read:1,compare:1,
write:1,selfwrite_add:1,selfwrite_delete:1,proxy:0
givenName: Carla
aclRights;attributeLevel;sn: search:1,read:1,compare:1,
write:1,selfwrite_add:1,selfwrite_delete:1,proxy:0
sn: Fuente
aclRights;attributeLevel;cn: search:1,read:1,compare:1,
write:1,selfwrite_add:1,selfwrite_delete:1,proxy:0
cn: Carla Fuente
aclRights;attributeLevel;userPassword: search:0,read:0,
compare:0,write:1,selfwrite_add:1,selfwrite_delete:1,proxy:0
userPassword: {SSHA}wnbWHIq2HPiY/5ECwe6MWBGx2KMiZ8JmjF80Ow==
aclRights;attributeLevel;manager: search:1,read:1,compare:1,
write:0,selfwrite_add:0,selfwrite_delete:0,proxy:0
manager: uid=bjensen,ou=People,dc=example,dc=com
aclRights;attributeLevel;telephoneNumber: search:1,read:1,compare:1,
write:1,selfwrite_add:1,selfwrite_delete:1,proxy:0
telephoneNumber: (234) 555-7898
aclRights;attributeLevel;objectClass: search:1,read:1,compare:1,
write:1,selfwrite_add:1,selfwrite_delete:1,proxy:0
objectClass: top
objectClass: person
```

```
objectClass: organizationalPerson
objectClass: inetorgperson
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
```

## 22.7.3 Understanding Effective Rights Results

Depending on the options specified, an effective rights request returns the following information:

- Rights Information
- `write`, `selfwrite_add`, and `selfwrite_delete` Permissions
- Logging Information

### 22.7.3.1 Rights Information

The effective rights information is presented according to the following subtypes:

`aclRights;entrylevel` - Presents entry-level rights information

`aclRights;attributelevel` - Presents attribute-level rights information

`aclRightsInfo;entrylevel` - Presents entry-level logging information

`aclRightsInfo;attributelevel` - Presents attribute-level logging information

The format of the `aclRights` string is as follows:

`aclRights;entryLevel:` *permission*:*value*(*permission*:*value*)*

and

`aclRights;attributeLevel:` *permission*:*value*(*permission*:*value*)*

The possible entry-level permissions are `add`, `delete`, `read`, `write`, and `proxy`. The possible values for each permission are `0` (permission not granted) and `1` (permission granted).

| Entry-level Permission | Explanation |
|---|---|
| `add` and `delete` | The ability of a user to add and delete the entire entry. |
| `read` | The ability of a user to read and search attributes in the entry. |
| `write` | The ability of a user to add, delete, and replace attribute values in the entry. |
| `proxy` | The ability of a user to access the directory with the rights of the entry. |

**Note:** For information about assigning these permissions in an ACI, see Section 8.2, "ACI Syntax".

The possible attribute-level permissions are `read`, `search`, `compare`, `write`, `selfwrite_add`, `selfwrite_delete`, and `proxy`. The possible values for each permission are `0` (permission not granted) and `1` (permission granted). For the case of the write permission, the value of "`?`" is also permitted.

| Attribute-level Permission | Explanation |
|---|---|
| `read` | The ability of a user to read the attribute value in the entry. |
| `search` | The ability of a user to search the attribute value in the entry. |
| `compare` | The ability of a user to compare the attribute value in the entry with a value that is provided by the user. |
| `write` | The ability of a user to add, delete, and replace the attribute value in the entry. This applies to all attributes except the `authorization dn`. |
| `selfwrite_add` | The ability of a user to add the attribute, `authorization dn`. |
| `selfwrite_delete` | The ability of a user to delete the attribute, `authorization dn`. |
| `proxy` | The ability of a user to access the directory with the rights of the attribute in the entry. |

**Note:** The `write`, `selfwrite_add`, and `selfwrite_delete` permissions are particularly complex. If you see a "`?`", consult the logging information to establish why the permissions will or will not be granted. For more information, see Table 22–1.

The format of the `aclRightsInfo` string is as follows:

```
aclRightsInfo;logs;entryLevel;permission:
acl_summary(main):permission-string
```

and

```
aclRightsInfo;logs;attributeLevel;permission;attribute:
acl_summary(main):permission-string
```

The entry-level and attribute-level permissions are described in the preceding section.

The *permission-string* contains detailed information about the effective rights at the entry-level and attribute-levels.

#### 22.7.3.2 `write`, `selfwrite_add`, and `selfwrite_delete` Permissions

The attribute-level permission for `write` can be either `0`, `1`, or "`?`". Only `write` attribute-level permissions can have a value of "`?`", which usually results from a `targattrfilters` ACI component. For add and delete permissions, the entries that can be modified depend on the values of the attributes in the entry. Only the permission, `0` or `1`, is returned on the entries as they are returned with the `ldapsearch` operation.

For all attribute values except the `authorization dn`, if the value for a `write` permission is `1`, the permission is granted for both add and delete. Similarly, for all attribute values except the `authorization dn`, a value of `0` for a write permission means that the permission is not granted for either add or delete `ldapmodify` operations. The permission in force for the value of the `authorization dn` is returned explicitly in one of the `selfwrite` permissions, that is, either `selfwrite_add` or `selfwrite_delete`.

Although selfwrite_add and selfwrite_delete attribute-level permissions do not exist in the context of ACIs, a set of ACIs can grant a user selfwrite permission for just the add or just the delete part of a modify operation. For selfwrite permissions, the value of the attribute being modified is the authorization dn. The same distinction is not made for write permissions because the value of the attribute being modified for a write permission is undefined.

When the effective permission depends on a targattrfilters ACI, the "?" value indicates that the logging information should be consulted for more permission detail. The interdependencies between the write, selfwrite_add, and selfwrite_delete permissions are fairly complex and are outlined in the following table.

***Table 22–1 Effective Rights Permission Interdependencies***

| write | selfwrite_add | selfwrite_delete | Effective Rights Explanation |
|---|---|---|---|
| 0 | 0 | 0 | Cannot add or delete any values of this attribute. |
| 0 | 0 | 1 | Can only delete the value of the authorization dn. |
| 0 | 1 | 0 | Can only add the value of the authorization dn. |
| 0 | 1 | 1 | Can only add or delete the value of the authorization dn. |
| 1 | 0 | 0 | Can add or delete all values except the authorization dn. |
| 1 | 0 | 1 | Can delete all values including the authorization dn and can add all values excluding the authorization dn. |
| 1 | 1 | 0 | Can add all values including the authorization dn and can delete all values excluding the authorization dn. |
| 1 | 1 | 1 | Can add or delete all values of this attribute. |
| ? | 0 | 0 | Cannot add or delete the authorization dn value, but might be able to add or delete other values. Refer to logging information for further details regarding the write permission. |
| ? | 0 | 1 | Can delete but cannot add the value of the authorization dn, and might be able to add or delete other values. Refer to logging information for further details regarding the write permission. |
| ? | 1 | 0 | Can add but cannot delete the value of the authorization dn and might be able to add or delete other values. Refer to logging information for further details regarding the write permission. |
| 1 | ? | 1 | Can add and delete the value of the authorization dn and might be able to modify add, modify, or delete other values. Refer to logging information for further details regarding the write permission. |

### 22.7.3.3 Logging Information

The effective rights logging information enables you to understand and debug access control difficulties. The logging information contains an access control summary statement, called the `acl_summary`, that indicates why access control has been allowed or denied. The access control summary statement includes the following information:

- Whether access was allowed or denied
- The permissions granted
- The target entry of the permissions
- The name of the target attribute
- The subject of the rights being requested
- Whether or not the request was made by proxy, and if so, the proxy authentication DN
- The reason for allowing or denying access (important for debugging purposes as explained in the following table)

The following table lists the effective rights logging information reasons and their explanations.

***Table 22–2 Effective Rights Logging Information Reasons and Their Explanations***

| Logging Information Reason | Explanation |
|---|---|
| `no reason available` | No reason available to explain why access was allowed or denied. |
| `no allow acis` | No `allow` ACIs exist, which results in denied access. |
| `result cached deny` | Cached information was used to determine the access denied decision. |
| `result cached allow` | Cached information was used to determine the access allowed decision. |
| `evaluated allow` | An ACI was evaluated to determine the access allowed decision. The name of the ACI is included in the log information. |
| `evaluated deny` | An ACI was evaluated to determine the access denied decision. The name of the ACI is included in the log information. |
| `no acis matched the resource` | No ACIs match the resource or target, which results in denied access. |
| `no acis matched the subject` | No ACIs match the subject requesting access control, which results in denied access. |
| `allow anyone aci matched anon user` | An ACI with a `userdn = "ldap:///anyone"` subject allowed access to the anonymous user. |
| `no matching anyone aci for anon user` | No ACI with a `userdn= "ldap:///anyone"` subject was found, so access for the anonymous user was denied. |
| `user root` | The user is root DN and is allowed access. |

**Note:** Write permissions for virtual attributes are not provided, nor is any associated logging evaluation information, because virtual attributes cannot be updated.

### 22.7.4 Restricting Access to the Get Effective Rights Control

Viewing effective rights is itself a directory operation that should be protected and appropriately restricted.

The default ACI does not allow read access to the `aclRights` and `aclRightsInfo` operational attributes used to return effective rights. Create a new ACI for these attributes to enable access by directory users to this information.

For example, the following ACI allows members of the Directory Administrators group to get effective rights:

```
aci: (targetattr = "aclRights||aclRightsInfo")(version 3.0; acl
"getEffectiveRights";
allow(all) groupdn = "ldap:///cn=Directory
Administrators,ou=Groups,dc=example,dc=com";)
```

In addition, access is needed to use the Get Effective Rights Control.

To enable access by directory users to the Get Effective Rights Control, create a new ACI target by using the OID (`1.3.6.1.4.1.42.2.27.9.5.2`) for this control. For additional ACI syntax information, see Section 8.2.2, "Defining Targets".

For example, the following ACI allows members of the Directory Administrators group to use the Get Effective Rights control:

```
aci: (targetcontrol = "1.3.6.1.4.1.42.2.27.9.5.2")(version 3.0;
acl "getEffectiveRights control access";
allow(all) groupdn = "ldap:///cn=Directory
Administrators,ou=Groups,dc=example,dc=com";)
```

# 23

# Managing Administrative Users

Oracle Unified Directory provides a flexible Privilege Subsystem that allows you to configure root users, Global Administrators, and administrators for your server. You can configure multiple root users and assign different root privileges to each administrator. For administrative domains, you can also configure multiple Global Administrators to manage administrative domains in your network or in a replicated environment.

The topics in this section describe the management of multiple root users and the privilege subsystem. The topics also provide instructions on how to configure and maintain the various user accounts required to administer your server securely.

Before you start with the procedures outlined here, determine the following guidelines for your server:

- Number of root users, their privileges, and resource limits, if any.
- Number of administrators, their privileges, and resource limits, if any.
- Guidelines for user accounts on your system.
- Password policies for the server and for specific groups of users.

This chapter covers the following topics:



## 23.1 Working With Multiple Root Users

Oracle Unified Directory provides one default root DN or root user, `"cn=Directory Manager"`. The default root DN is a user entry assigned with specialized privileges with full read and write access to all data in the server. Comparable to a Unix root user or superuser, the root DN can bypass access controls to carry out tasks on the server. The root user is defined below the `"cn=Root DNs,cn=config"` branch of the server at `cn=Directory Manager,cn=Root DNs,cn=config`.

The server supports multiple root users who have their own entries and their own set of credentials on the server. This allows you to assign privileges to a user who might need root access for a particular task but might not need the full set of root user

privileges. With each entry, you can assign strong authentication such as the GSSAPI SASL mechanism, password policies, or add resource limits (if your schema allows it) to one root user while having a completely different configuration for another root user.

Root users differ from regular user entries in the following ways:

- **Configuration**. Root users are the only user accounts that can exist in the server configuration (`cn=config`).
- **Privilege inheritance**. Root users automatically inherit the set of default root user privileges. Regular users do not automatically receive any privileges unless explicitly granted. You can grant privileges using real, virtual root-privilege-name attributes, or both in the entry.
- **Lockdown mode**. Root users are the *only* users who can cause the server to enter or leave lockdown mode and only over the loopback interface.

The Privilege Subsystem supports the configuration of multiple root users.

## 23.2 Root Users and the Privilege Subsystem

The Privilege Subsystem allows you to assign refined privileges to users who might require only a specific set of root user access privileges. Root users are automatically granted a set of privileges defined in the `default-root-privilege-name` attribute in the `"cn=Root DNs,cn=config"` subtree.

The Privilege Subsystem is independent from the Access Control Subsystem, but some operations might be subject to access controls.

The following set of privileges are automatically assigned to the root user.

| Privilege | Description |
|---|---|
| `bypass-acl` | Allows the user to bypass access control evaluation. |
| `modify-acl` | Allows the user to make changes to access control instructions defined in the server. |
| `config-read` | Allows the user to have read access to the server configuration. |
| `config-write` | Allows the user to have write access to the server configuration. |
| `ldif-import` | Allows the user to request the LDIF import task. |
| `ldif-export` | Allows the user to request the LDIF export task. |
| `backend-backup` | Allows the user to request the back-end backup task. |
| `backend-restore` | Allows the user to request the back-end restore task. |
| `server-shutdown` | Allows the user to request the server shutdown task. |
| `server-restart` | Allows the user to request the server restart task. |
| `disconnect-client` | Allows the user to terminate arbitrary client connections. |

| Privilege | Description |
|---|---|
| cancel-request | Allows the user to cancel arbitrary client requests. |
| unindexed-search | Allows the user to request unindexed search operations. |
| password-reset | Allows the user to reset the user passwords. |
| update-schema | Allows the user to update the server schema. |
| privilege-change | Allows the user to change the set of privileges assigned to a user, or to change the set of default root privileges. |

The following privileges can be assigned to the root user.

| Privilege | Description |
|---|---|
| jmx-read | Allows the user to read JMX attribute values. |
| jmx-write | Allows the user to update JMX attribute values. |
| jmx-notify | Allows the user to subscribe to JMX notifications. |
| proxied-auth | Allows the user to use the proxied authorization control or to request an alternate SASL authorization ID. |

## 23.3 Managing Root Users With dsconfig

Use the dsconfig command to manage root users. For more information, see Section 14.1, "Managing the Server Configuration With dsconfig".

### 23.3.1 To View the Default Root User Privileges

The default root user has a number of privileges, which are stored as values of the default-root-privilege-name property.

1. View the default root user privileges by running the following dsconfig command:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -n \
  get-root-dn-prop

Property                    : Value(s)
----------------------------:-----------------------------------------------
default-root-privilege-name : backend-backup, backend-restore, bypass-acl,
                                     : cancel-request, config-read,
config-write,
                                     : disconnect-client, ldif-export,
ldif-import,
                                     : modify-acl, password-reset,
privilege-change,
                                     : server-restart, server-shutdown,
                                     : unindexed-search, update-schema
```

### 23.3.2 To Edit the Default Root User Privileges

The easiest way to manage root user privileges is to use dsconfig in interactive mode. Interactive mode walks you through the root user configuration, and is therefore not documented here.

To add or remove privileges for the default root user, add or remove the values of the default-root-privilege-name property. This property can hold the following values:

- backend-backup
- backend-restore
- bypass-acl
- cancel-request
- config-read
- config-write
- disconnect-client
- jmx-notify
- jmx-read
- jmx-write
- ldif-export
- ldif-import
- modify-acl
- password-reset
- privilege-change
- proxied-auth
- server-restart
- server-shutdown
- unindexed-search
- update-schema

This example adds the jmx-notify privilege to the default root user, by using dsconfig in non-interactive mode.

1. Run the dsconfig command as follows:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -n \
  set-root-dn-prop --add default-root-privilege-name:jmx-notify
```

### 23.3.3 To Create a Root User

Root users are stored below the entry cn=Root DNs,cn=config. To create a new root user, create the entry in LDIF and add it by using the ldapmodify command. Note that the cn=config suffix is available only through the administration connector, and must therefore be accessed over SSL, via the administration port.

Root users automatically inherit the set of default root user privileges on the server. For information about adding or removing privileges for a specific root user, see To Change a Root User's Privileges.

1. Create a root user entry below the cn=Root DNs,cn=config entry.

   The following LDIF file represents a new root user named "Administration Manager". The entry is saved in a file named add-root-user.ldif.

```
dn: cn=MyRootUser,cn=Root DNs,cn=config
objectClass: inetOrgPerson
objectClass: person
objectClass: top
objectClass: ds-cfg-root-dn-user
objectClass: organizationalPerson
userPassword: password
cn: MyRootUser
sn: MyRootUser
ds-cfg-alternate-bind-dn: cn=MyRootUser
givenName: Directory
```

2. Use the ldapmodify command to add the entry.

```
$ ldapmodify -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file \
  --useSSL --defaultAdd --filename "add-root-user.ldif"
Processing ADD request for cn=MyRootUser,cn=Root DNs,cn=config
ADD operation successful for DN cn=MyRootUser,cn=Root DNs,cn=config
```

3. Use the ldapsearch command to display all the root users defined in the server.

```
$ ldapsearch -p 4444 -b "cn=root DNs,cn=config" -D "cn=directory manager" -j
pwd-file \
--useSSL "objectclass=*" dn
dn: cn=Root DNs,cn=config

dn: cn=MyRootUser,cn=Root DNs,cn=config

dn: cn=Directory Manager,cn=Root DNs,cn=config
```

### 23.3.4 To Change a Root User's Password

1. Create a password in a secure file.
2. Use ldappasswordmodify to change the password.

```
$ ldappasswordmodify -h localhost -p 4444 -D "cn=MyRootUser" -j pwd-file \
  --useSSL --newPasswordFile rootuser_pwd.txt
The LDAP password modify operation was successful
```

### 23.3.5 To Change a Root User's Privileges

If you want to have a different set of privileges for a specific root user, add the ds-privilege-name attribute to that root user's entry.

The following example gives the root user "cn=MyRootUser,cn=Root DNs,cn=config" the ability to use proxied authorization. The example removes the ability to change user privileges or access the configuration. The minus sign before the privilege indicates that the privilege is being removed rather than granted.

1. Apply the following LDIF statement to the root user's entry:

```
dn: cn=MyRootUser,cn=Root DNs,cn=config
changetype: modify
add: ds-privilege-name
ds-privilege-name: proxied-auth
ds-privilege-name: -config-read
ds-privilege-name: -config-write
```

In this example, the root user `"cn=MyRootUser,cn=Root DNs,cn=config"` would inherit all privileges automatically granted to root users with the exception of the `config-read` and `config-write` privileges. The user would also be given the `proxied-auth` privilege.

## 23.4 Setting Root User Resource Limits

You can set resource limits on the server for search operations by using the operational attributes on the client application that is binding to the server. The following resource limits are available:

- **Look-through limit**. Specify the maximum number of entries that can be examined during a single search operation. Use the `ds-rlim-lookthrough-limit` operational attribute.
- **Size limit**. Specify the maximum number of entries that can be returned in a single search operation. Use the `ds-rlim-size-limit` operational attribute.
- **Time limit**. Specify the maximum length of time in seconds that the server can spend processing a search operation. Use the `ds-rlim-time-limit` operational attribute.

The following LDIF update statement sets resource limits for the new root user created in the previous section. This statement should be applied to the root user's entry.

```
dn: cn=MyRootUser,cn=Root DNs,cn=config
changetype: modify
add: ds-rlim-lookthrough-limit
ds-rlim-lookthrough-limit: 1000
-
add: ds-rlim-size-limit
ds-rlim-size-limit: 500
-
add: ds-rlim-time-limit
ds-rlim-time-limit: 300
```

To set a particular resource limit to *unlimited*, set the value of the corresponding attribute to `0` (zero).

## 23.5 Managing Administrators

An administrator generally has broader rights and permissions than most users. You can create a number of administrators, with different access controls and resource limits.

### 23.5.1 To Create a New Administrator

1. Import the administrator data using `import-ldif`.

   For this example, the administrator being added has `uid=Admin.Lab`.

   a. Alternatively, you can use an existing user.

2. Create a group of administrators with `cn=Administrators`.

   Since the group of administrators should have only a few users, you can create a static group. For more information, see Defining Static Groups.

```
dn: cn=Administrators,ou=People,dc=example,dc=com
objectClass: top
objectClass: groupOfNames
member: uid=Admin.Lab,ou=People,dc=example,dc=com
cn: Administrator
```

3. Set the privileges of the administrator by using the `ldapmodify` command.

   For example, the following command would give the administrator the rights to perform backup and restore on the back end.

```
ldapmodify -h localhost -p 1389 -j pwd-file -D "cn=directory manager"
dn: uid=Admin.Lab,ou=People,dc=example,dc=com
changetype: modify
add: ds-privilege-name
ds-privilege-name: backend-backup
ds-privilege-name: backend-restore
```

4. Set resource limits, if required.

   The procedure is similar to setting the resource limits for a root user. See Setting Root User Resource Limits.

## 23.5.2 To Create an Administrator with Root User Privileges

You can assign root user privileges to an administration or user.

1. Create an administrator, or use an existing user.

   See To Create a New Administrator.

2. Modify the privileges using the `ldapmodify` command.

   For example, the command below changes the privileges for the administrator named `Admin.Lab` to have the same default privileges as a root user.

```
ldapmodify -h localhost -p 1389 -j pwd-file -D "cn=directory manager"
dn: uid=Admin.Lab,cn=Administrators,ou=People,dc=example,dc=com
changetype: modify
add: ds-privilege-name
ds-privilege-name: bypass-acl
ds-privilege-name: modify-acl
ds-privilege-name: config-read
ds-privilege-name: config-write
ds-privilege-name: ldif-import
ds-privilege-name: ldif-export
ds-privilege-name: backend-backup
ds-privilege-name: backend-restore
ds-privilege-name: server-shutdown
ds-privilege-name: server-restart
ds-privilege-name: disconnect-client
ds-privilege-name: cancel-request
ds-privilege-name: password-reset
ds-privilege-name: update-schema
ds-privilege-name: privilege-change
ds-privilege-name: unindexed-search
```

**Note:** The privileges on the access controls `bypass-acl` and `modify-acl` should only be assigned to a restricted number of people. Assigning the rights to bypass or modify access controls to inexperienced users can be risky.

## 23.6 Managing Global Administrators

When you set up replication servers using the graphical installer or the `dsreplication` command, you are prompted to set a user name and password for the Global Administrator. The Global Administrator is responsible for managing and maintaining administrative server domains in replicated environments.

The Global Administrator exists in the `cn=Administrators,cn=admin data` subtree. To view the Global Administrator entry, run the following `ldapsearch` command:

```
$ ldapsearch -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file \
  --useSSL -b "cn=Administrators,cn=admin data" -s sub "(objectclass=*)"
dn: cn=Administrators,cn=admin data
objectClass: top
objectClass: groupofurls
description: Group of identities which have full access.
cn: Administrators
memberURL: ldap:///cn=Administrators,cn=admin data??one?(objectclass=*)
dn: cn=admin,cn=Administrators,cn=admin data
objectClass: person
objectClass: top
userPassword: {SSHA}+ed1wbhcWjxtv2zJ6OHEA2TuE9n1qIJGnuR94w==
description: The Administrator that can manage all the OUD instances.
cn: admin
sn: admin
```

# 24

# Managing Password Policies

A *password policy* is a set of rules governing the use of passwords in the system and is an integral component of any security strategy employed for your directory.

Oracle Unified Directory includes a default password policy for general users and a default root users password policy. These default password policies reside in the directory server's configuration and can be modified.

Oracle Unified Directory supports multiple password policies, so you can create and configure specialized password policies for a specific set of users in addition to using the default password policies. Customized password policies can be defined as LDAP subentries, and stored with the user data, which allows them to be replicated across servers.

This chapter outlines the components of password policies and provides procedures to configure and manage password policies. The chapter covers the following topics:



## 24.1 Password Policy Components

All password policies involve the following configurable components:

- **Password complexity requirements**. Specifies the composition of the password and its required number of characters. Typically, you would specify the minimum number of characters used in a password, the type of characters allowed, and the required number of numeric characters. For example, many institutions require a minimum of seven or eight characters, one numeral, one special character, as well as a mix of uppercase and lowercase letters.
- **Password history**. Determines the number of unique passwords a user must use before an old password can be reused.
- **Maximum password age**. Determines how long a password can be used before the user is allowed or required to change it.

- **Minimum password age**. Determines how long a new password must be kept before the user can change it.
- **First Login**. Determines if the user will be required to change his password upon first logging in to the system.
- **Authorized password change**. Refers to the conditions under which a user can change his password. For example, before a user can change his password, the server can be configured to require the user to enter his current password to authenticate his identity before entering a new password.
- **Account lockout**. Determines the conditions under which an account is disabled for access by the user. For example, if a user fails to properly authenticate after three attempts, then the server can be configured to lock the account on the fourth attempt. The administrator will be required to manually unlock the account for user.
- **Password storage scheme**. Determines how the password is to be encrypted and stored on the server. You can configure storage schemes for certain accounts on the server. For example, root user passwords require strong encryption due to the importance of the account and its privileges. Thus, you can configure the use the SSHA-512 storage scheme to store root user passwords.

Password validation is not handled directly in the password policy, but by specific password validator entries, the DNs of which are present in the password policy. For more information, see Section 24.6, "Password Validators".

## 24.2 The Default Password Policy

The Default Password Policy includes a number of configurable properties. These are listed in the following table.

| Property | Description |
|---|---|
| `account-status-notification-handler` | The account status notification handler is used to send messages when events occur during the course of password policy processing. This property specifies the DNs of the account status notification handlers that should be used for this password policy. |
| `allow-expired-password-changes` | Not recommended. Indicates whether users are allowed to change their passwords after the passwords have expired. The user needs to issue the request anonymously and include the current password in the request. If this property is enabled, this feature uses the Password Modify Extended Operation, which is enabled by default at initial configuration. |
| `allow-user-password-changes` | Indicates whether users are allowed to change their own passwords if they have access control rights to do so. |
| `default-password-storage-scheme` | Specifies the DNs for the password storage schemes that are used to encode clear-text passwords for this password policy. |

| Property | Description |
|---|---|
| deprecated-password-storage-scheme | Specifies the DNs for password storage schemes that are considered deprecated for this password policy. If a user with this password policy authenticates to the server and his password is encoded with any deprecated schemes, those values are removed and replaced with values encoded using the default password storage scheme. |
| expire-password-without-warning | Indicates whether user passwords are allowed to expire even if the user has not yet seen a password expiration warning. If this is set to false, the user is always guaranteed to see at least one warning message even if the password expiration time has passed. The expiration time will be reset to the current time plus the warning interval (ds-cfg-password-expiration-warning-interval). |
| force-change-on-add | Indicates whether users are required to change their passwords the first time they use their accounts and before they are allowed to perform any other operation. |
| force-change-on-reset | Indicates whether users are required to change their passwords after an administrative password reset and before they are allowed to perform any other operation. |
| grace-login-count | Specifies the maximum number of grace login that a user should be given. A grace login makes it possible for a user to authenticate to the server even after the password has expired, but the user is not allowed to do anything else until he has changed his password. |
| idle-lockout-interval | Specifies the maximum length of time that a user account can remain idle (that is, that the user may go without authenticating to the directory) before the server locks the account. This action is enforced if last login time tracking is enabled and if the idle lockout interval is set to a nonzero value. |
| last-login-time-attribute | Specifies the name of the attribute in the user's entry that is used to hold the last login time for the user. If this is provided, the specified attribute must either be defined as an operational attribute in the server schema, or it must be allowed by at least one of the object classes in the user's entry. The ds-pwp-last-login operational attribute has been defined for this purpose. Last login time tracking is only enabled if the ds-cfg-last-login-time-attribute and ds-cfg-last-login-time-format attributes have been configured for the password policy. |

| Property | Description |
|---|---|
| last-login-time-format | Specifies the format string that should be used to generate the last login time values. This can be any valid format string that can be used in conjunction with the java.text.SimpleDateFormat class. Note that for performance reasons, it might be desirable to configure this attribute so that it only stores the date (format: yyyyMMdd) and not the time of the last login. Then, it only needs to be updated once per day, rather than each time the user may authenticate. Last login time tracking is only enabled if the ds-cfg-last-login-time-attribute and ds-cfg-last-login-time-format attributes have been configured for the password policy. |
| lockout-duration | Specifies the length of time that a user account should remain locked due to failed authentication attempts before it is automatically unlocked. A value of "0 seconds" indicates that any locked accounts are not automatically unlocked and must be reset by an administrator. |
| lockout-failure-count | Specifies the number of authentication failures required to lock a user account, either temporarily or permanently. A value of zero indicates that automatic lockout is not enabled. |
| lockout-failure-expiration-interval | Specifies the maximum length of time that a previously failed authentication attempt should be counted toward a lockout failure. Note that the record of all previous failed attempts is always cleared upon a successful authentication. A value of "0 seconds" indicates that failed attempts are never automatically expired. |
| max-password-age | Specifies the maximum length of time that a user is allowed to keep the same password before choosing a new one. This is often known as the *password expiration interval*. A value of "0 seconds" indicates that passwords never expire. If the ds-cfg-expire-passwords-without-warning attribute is set to false, the effective password expiration time is recalculated to be the time at which the first warning is received, plus the warning interval (ds-cfg-password-expiration-warning-interval). This behavior ensures that a user always has the full configured warning interval to change his password. |

| Property | Description |
|---|---|
| max-password-reset-age | Specifies the maximum length of time that users are allowed to change their passwords after they have been administratively reset and before they are locked out. This is only applicable if the ds-cfg-force-change-on-reset attribute is set to true. A value of "0 seconds" indicates that there are no limits on the length of time that users have to change their passwords after administrative resets. |
| min-password-age | Specifies the minimum length of time that a user is required to have a password value before it can be changed again. Providing a nonzero value ensures that users are not allowed to repeatedly change their passwords in order to flush their previous password from the history so it can be reused. |
| password-attribute | Specifies the attribute in the user's entry that holds the encoded passwords for the user. The specified attribute must be defined in the server schema, and it must have either the user password syntax or the authentication password syntax. Typically, you enter "userPassword" for the User Password syntax (OID: 1.3.6.1.4.1.26027.1.3.1). You can also specify, if your server supports it, the value authPassword for the authenticated password syntax (OID: 1.3.6.1.4.1.4203.1.1.2). |
| password-change-requires-current-password | Indicates whether users are required to provide their current password when setting a new password. If this is set to true, then users are required to provide their current password when changing their existing password. This may be done using the password modify extended operation, or using a standard LDAP modify operation by deleting the existing password value and adding the new password value in the same modify operation. |
| password-expiration-warning-interval | Specifies the length of time before the password expires that the users should start to receive notification that it is about to expire. This must be given a nonzero value if the ds-cfg-expire-passwords-without-warning attribute is set to false. |
| password-generator | Specifies the DN for the password generator that should be used in conjunction with this password policy. The password generator is used in conjunction with the password modify extended operation to provide a new password for cases in which the client did not include one in the request. If no password generator DN is specified, then the password modify extended operation does not automatically generate passwords for users. |

| Property | Description |
|---|---|
| password-history-count | Specifies the maximum number of password values that should be maintained in the password history. Whenever a user's password is changed, the server checks the proposed new password against the current password and all passwords stored in the history. If a match is found, then the user is not allowed to use that new password. A value of zero indicates either that the server should not maintain a password history (that is, the password history duration has a value of "0 seconds") or that the password history list should be based entirely on duration and no maximum count should be enforced (that is, the password history duration has a value other than "0 seconds"). Note that if an administrator reduces the configured password history count to a smaller (but still nonzero) value, each user entry containing password history state information is not impacted until a password change is processed for that user. At that time, any excess history state values is purged from the entry. If the history count is reduced to zero and the password history duration is also set to "0 seconds," any state information in the user's entry is retained in case the feature is re-enabled. |
| password-history-duration | Specifies the maximum length of time that a formerly used password should remain in effect in the user's password history. Whenever a user's password is changed, the server checks the proposed new password against the current password and all passwords stored in the history. If a match is found, the user is not allowed to use that new password. A value of "0 seconds" indicates either that the server should not maintain a password history (that is, the password history count has a value of "0") or that the password history list should be based entirely on count and no maximum duration should be enforced (that is, the password history count has a value other than "0"). |
| password-validator | Specifies the DNs for password validators that should be used in conjunction with this password policy. The password validators are invoked whenever a user attempts to provide a new password in order to determine whether that new password is acceptable. |
| previous-last-login-time-format | Specifies the format string that was used in the past for older last login time values. This value is not necessary unless the last login time feature is enabled and the format in which the values are stored has been changed. |
| require-change-by-time | Specifies a time by which all users with this password policy are required to change their passwords. This option works independently of password expiration (that is, force all users to change their passwords at some point even if password expiration is disabled). |

| Property | Description |
|---|---|
| require-secure-authentication | Indicates whether users with this password policy are required to authenticate in a secure manner using a secure communication mechanism like SSL, or a secure SASL mechanism like DIGEST-MD5, EXTERNAL, or GSSAPI that does not expose the password in the clear. |
| require-secure-password-changes | Indicates whether users with this password policy are required to make password changes in a secure manner, such as over a secure communication channel like SSL. |

### 24.2.1 To View the Properties of the Default Password Policy

You can view the properties of the default password policy by using the `dsconfig` command, or by using ODSM.

To view the properties by using `dsconfig`, run the following command:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  get-password-policy-prop --policy-name "Default Password Policy"

Property                                 : Value(s)
-----------------------------------------:-------------------------
account-status-notification-handler      : -
allow-expired-password-changes           : false
allow-user-password-changes              : true
default-password-storage-scheme          : Salted SHA-1
deprecated-password-storage-scheme       : -
expire-passwords-without-warning         : false
force-change-on-add                      : false
force-change-on-reset                    : false
grace-login-count                        : 0
idle-lockout-interval                    : 0 s
last-login-time-attribute                : -
last-login-time-format                   : -
lockout-duration                         : 0 s
lockout-failure-count                    : 0
lockout-failure-expiration-interval      : 0 s
max-password-age                         : 0 s
max-password-reset-age                   : 0 s
min-password-age                         : 0 s
password-attribute                       : userpassword
password-change-requires-current-password : false
password-expiration-warning-interval     : 5 d
password-generator                       : Random Password Generator
password-history-count                   : 0
password-history-duration                : 0 s
password-validator                       : -
previous-last-login-time-format          : -
require-change-by-time                   : -
require-secure-authentication            : false
require-secure-password-changes          : false
```

To view any advanced properties, include the `--advanced` option, as follows:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  get-password-policy-prop --policy-name "Default Password Policy" --advanced
```

To view the properties of the default password policy by using ODSM, do the following:

- Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager".
- Select the **Security** tab.
- Expand the **Password Policy** element.
- Select **Default Password Policy**.

The password policy properties, and their values, are displayed in the right-hand pane.

### 24.2.2 To Modify the Default Password Policy

You can modify the properties of the default password policy by using the `dsconfig` command, or by using ODSM.

To modify the properties by using `dsconfig`, run the following command:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  set-password-policy-prop --policy-name "Default Password Policy" \
  --set allow-expired-password-changes:true
```

To modify the properties of the default password policy by using ODSM, do the following:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager".
2. Select the **Security** tab.
3. Expand the **Password Policy** element.
4. Select **Default Password Policy**.

   The password policy properties, and their values, are displayed in the right-hand pane.
5. Modify the required property and click **Apply**.

You cannot display or modify advanced properties by using ODSM.

## 24.3 Password Policies in a Replicated Environment

The password policies that reside in the directory server configuration (under `cn=config`) are not replicated. Configuration information in general is not replicated and is specific to each directory server instance. If you modify the default password policy, you must make the same changes on each directory server instance in a replicated topology. Similarly, specialized password policies under `cn=config` are not replicated to other directory servers.

Password policies that are created as subentries (that is, as part of the data) are replicated. For information about creating password policies as subentries, see Section 24.4.7, "To Define a Password Policy as an LDAP Subentry".

Additional considerations for using password policies in replicated environments include the following:

- The directory server replicates all password information (current password, password history, password expiration) that is stored in the user entry.

- If a user changes his password, the new password might take a while to be updated on all replicas.
- A user might receive multiple password expiration warnings, one from each replicated server.

## 24.4 Configuring Password Policies by Using the Command Line

The easiest way to configure a password policy is by using the `dsconfig` command to manage the existing password policies and to modify the password policy properties.

This section covers the following topics:



### 24.4.1 Configuring the Default Password Policy

The following examples use `dsconfig` to modify various properties of the default password policy.

***Example 24–1 Configuring Account Lockout***

The following account lockout features can be configured:

- **Lockout failure count.** Specifies the number of authentication failures required to lock a user account.
- **Lockout duration.** Determines the length of time that the account is in a locked state after failed authentication attempts. After the duration time, the account is automatically unlocked. A value of zero indicates that the account is not be automatically unlocked.
- **Lockout failure expiration interval.** Determines the maximum length of time that a previously failed authentication attempt should be counted toward a lockout failure. A value of zero indicates that failed attempts never automatically expire.
- **Idle lockout interval.** Specifies the maximum length of time that a user account can go without authenticating to the directory before the server locks the account. This property is enforced if the `last-login-time` is enabled and `idle-lockout-interval` is set to a nonzero value.

The following command sets the account lockout properties for the default password policy.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -n \
  set-password-policy-prop \
  --policy-name "Default Password Policy" --set "lockout-failure-count:3" \
  --set "lockout-duration:15 minutes" --set "idle-lockout-interval:90 days" \
  --set "lockout-failure-expiration-interval:10 minutes"
```

***Example 24–2 Configuring Last Login***

*Last login* is a basic security feature that helps the user to keep track of the login history. The directory server provides an operational attribute, `ds-pwp-last-login`, that holds the user's last login time. If you specify another attribute, the operational attribute must be defined in the server schema, or it must be allowed by at least one of the object classes in the user's entry.

The `last-login-time-format` property determines the time format. If the time format has changed and last login is enabled, the `previous-last-login-time-format` property is used.

The following command sets the last login properties for the default password policy.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -n \
  set-password-policy-prop \
  --policy-name "Default Password Policy" \
  --set "last-login-time-attribute:ds-pwp-last-login-time" \
  --set "last-login-time-format:yyyyMMdd" \
  --set "previous-last-login-time-format:yyyyMMdd"
```

***Example 24–3 Configuring Password History Count and Duration***

The `password-history-count` property specifies the number of past passwords that should be maintained in the history. A value of zero indicates that the server does not maintain a password history.

The `password-history-duration` property specifies the maximum length of time that a previously used password should remain in the user's password history. A value of `0 seconds` indicates that the server should not maintain a password history.

The following command configures password history count and duration for the default password policy.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -n \
  set-password-policy-prop \
  --policy-name "Default Password Policy" --set "password-history-count:3" \
  --set "password-history-duration:5 seconds"
```

## 24.4.2 To Create a New Password Policy

You can configure and store multiple password policies with different configuration options. When you set up a directory server instance, the instance uses the default password policy and applies it to all user entries, except root users (for example, the `cn=Directory Manager` account).

You can change the default password policy or you can create new password policies for specific groups in your directory. If a specific property is not present in a password policy, the server reads that property from the default password policy, in other words, all password policies inherit their default values from the default password policy.

The following command creates a new password policy and sets the `default-password-storage-scheme`, `lockout-duration`, `lockout-failure-count`, and `password-change-requires-current-password` properties. The remaining properties are inherited from the default Password Policy.

Use the `dsconfig` command to create a new password policy, as follows:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  create-password-policy \
```

```
  --policy-name "Temp Password Policy" --set password-attribute:userPassword \
  --set default-password-storage-scheme:"Salted SHA-1" \
  --set lockout-duration:300s --set lockout-failure-count:3 \
  --set password-change-requires-current-password:true
```

### 24.4.3 To Create a First Login Password Policy

The First Login Password Policy is a specialized password policy that requires a user to change his password when first logging in to the system. Typically, an administrator sets up a new temporary password for newly created accounts, and the user is required to create his password after first logging in with the temporary password.

Use the `dsconfig` command to create a first login password policy.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  create-password-policy --policy-name "First Login Password Policy" \
  --set password-attribute:userpassword \
  --set default-password-storage-scheme:"Salted SHA-1" \
  --set allow-user-password-changes:true --set force-change-on-add:true \
  --set force-change-on-reset:true \
  --set expire-password-without-expiration:false \
  --set password-expiration-warning-interval:86400 \
  --set min-password-age:0 --set max-password-age:259200 \
  --set lockout-duration:3600 --set lockout-failure-count:3 \
  --set password-change-requires-current-password:true
```

### 24.4.4 To Assign a Password Policy to an Individual Account

You can assign a password policy to an individual by adding the `ds-pwp-password-policy-dn` attribute to the user's entry. The server then uses the configured password policy for that user.

1. Use `ldapmodify` to add the `ds-pwp-password-policy-dn` attribute.

   ```
   $ ldapmodify --h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
   dn: uid=mgarcia,ou=Contractors,dc=example,dc=com
   changetype: modify
   add: ds-pwp-password-policy-dn
   ds-pwp-password-policy-dn: cn=Temp Password Policy,cn=Password
   Policies,cn=config
   ```

2. Verify the entry by using `ldapsearch`.

   ```
   $ ldapsearch -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
     -b "dc=example,dc=com" -s sub "(uid=mgarcia)" ds-pwp-password-policy-dn
   ```

### 24.4.5 To Prevent Password Policy Modifications

To prevent users from modifying their password policy, you must add an ACI to the root entry.

Use the `ldapmodify` command with the specific ACI.

```
$ ldapmodify -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
dn: dc=example,dc=com
changetype: modify
add: aci
aci: (targetattr != "passwordPolicySubentry")(version 3.0; acl "Allow self
modification except for passwordPolicySubentry";
allow (write) (userdn = "ldap:///self");)
```

### 24.4.6 To Assign a Password Policy to a Group of Users

You can assign a password policy to a group of users by adding a virtual attribute that automatically assigns the `ds-pwp-password-policy-dn` attribute to all user entries that match the criteria associated with that virtual attribute. The criteria can be based entirely or in part on the group membership for a user.

Use `dsconfig` to create a virtual attribute that adds a password policy to a group of users.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  create-virtual-attribute \
  --name "Add PWPolicy to Admins" --type user-defined --set enabled:true \
  --set attribute-type:ds-pwp-password-policy-dn \
  --set group-dn:cn=Admins,ou=Groups,dc=example,dc=com \
  --set conflict-behavior:real-overrides-virtual \
  --set value:"cn=Admins PWPolicy,cn=Password Policies,cn=config"
```

### 24.4.7 To Define a Password Policy as an LDAP Subentry

LDAP subentries are special entries that hold operational data for the server. They are similar to operational attributes in that they are not returned to clients unless explicitly requested by including a Subentries Control request control.

You can define a password policy as an LDAP subentry, which means that the password policy is stored along with the user data, and can therefore be replicated.

Subentry password policies override the default password policy that is defined in the configuration. Settings that are not included in the subentry password policy are inherited from the default password policy.

When more than one password policy is defined under the same parent node with overlapping scope, the election of the password policy subentry that will apply to an entry within that scope cannot be determined. You must therefore ensure that the password policies are defined in such a way that they do not conflict with each other.

Subentry password policies must rely on standard password policy properties only. A subentry password policy cannot contain password policy extension that are specific to Oracle Unified Directory.

For subentry password policies, password validators and password generators are always inherited from the default server password policy. You cannot define password validators or password generators for individual password policy subentries.

To define a subentry password policy, create the password policy in an LDIF file, and add it to the data by using `ldapmodify`. You can specify the entries to which the password policy should be applied by including an LDAP filter in the subentry subtree specification.

The following example creates a password policy that applies only to a group of administrators. This password policy specifies the following:

- The user's account will be locked after a three successive failed password attempts.
- A failure interval of 300 seconds, after which a previously failed authentication attempt is no longer counted toward a lockout failure.
- A lockout duration of 300 seconds, after which it is automatically unlocked.
- Users to which this password policy applies can change their own passwords.

- Users with this password policy must change their password in a secure manner that does not expose the credentials.

1. Create an LDIF file (`admin-pwp.ldif`) that includes the entry specifying the password policy.

```
dn: cn=Admins Password Policy,dc=example,dc=com
objectClass: top
objectClass: subentry
objectClass: pwdPolicy
cn: Admins Password Policy
pwdAttribute: userPassword
pwdLockout: TRUE
pwdMaxFailure: 3
pwdFailureCountInterval: 300
pwdLockoutDuration: 300
pwdAllowUserChange: TRUE
pwdSafeModify: TRUE
subtreeSpecification: {relativeBase "ou=people", specificationFilter
  "(isMemberOf=cn=Admins,ou=Groups,dc=example,dc=com)" }
```

2. Use the `ldapmodify` command to add the entry to the directory.

```
$ ldapmodify -h localhost -p 1389 -D "cn=Directory Manager" -w password \
  --defaultAdd --filename admin-pwp.ldif
Processing ADD request for cn=Admins Password Policy,dc=example,dc=com
ADD operation successful for DN cn=Admins Password Policy,dc=example,dc=com
```

### 24.4.8 To Delete a Password Policy

You can delete any password policy, except the Default Password Policy and the Default Root User Policy, from the directory when it is no longer needed.

In practice, first check the users who have the password policy you plan to delete, move them to a new password policy, and then remove the old password policy. If a password policy is deleted, any users who have a deleted password policy continue to have the `ds-pwd-password-policy-dn` pointing to the old password policy. The server returns an error when any requests to access the entry occur.

Use `dsconfig` to delete a password policy.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -n \
  delete-password-policy --policy-name "Temp Password Policy"
```

## 24.5 Configuring Password Policies by Using Oracle Directory Services Manager

You can use ODSM to manage password policies, as described in the following sections.

### 24.5.1 List the Configured Password Policy Subentries

You can display all password policy subentries that are configured in the server by using ODSM, as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager".
2. Select the **Security** tab.
3. Expand the **Password Policy Subentry** element.

The DNs of all password policy subentries are listed.

4. To display the details of a password policy subentry, select its DN.

   The password policy subentry properties are displayed in the right hand pane.

5. To modify any aspect of the password policy subentry, change the required value and click **Apply**.

For a description of all possible properties, and their values, see "Password Policy" in the *Oracle Unified Directory Configuration Reference*.

### 24.5.2 Create a Password Policy Subentry

You can create a new password policy subentry by using ODSM, as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager".
2. Select the **Security** tab.
3. Expand the **Password Policy Subentry** element.
4. Click the **Add** icon.

   The password policy subentry properties are displayed in the right hand pane.

5. On the **Create new password policy subentry** screen, complete the required fields.

   For a description of all possible properties, and their values, see "Password Policy" in the *Oracle Unified Directory Configuration Reference*.

6. When you have completed configuring the password policy subentry, click **Create**.

### 24.5.3 Create a Password Policy Subentry Based on an Existing Password Policy Subentry

You can create a new password policy subentry that is based on an existing password policy subentry by using ODSM, as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager".
2. Select the **Security** tab.
3. Expand the **Password Policy Subentry** element.
4. Select the password policy subentry on which you want to base the new subentry.
5. Click the **Add like** icon.

   The properties of the original password policy subentry are displayed in the right hand pane.

6. Modify the required values.

   For a description of all possible properties, and their values, see "Password Policy" in the *Oracle Unified Directory Configuration Reference*.

7. When you have completed configuring the new password policy subentry, click **Create**.

### 24.5.4 Delete a Password Policy Subentry

You can delete a password policy subentry by using ODSM, as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager".
2. Select the **Security** tab.
3. Expand the **Password Policy Subentry** element.
4. Select the password policy subentry that you want to deleted.
5. Click the **Delete** icon.

   You are prompted to confirm the deletion. Click **OK**.

### 24.5.5 Display the Configured Password Policies

You can display the list of password policies by using ODSM, as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager".
2. Select the **Security** tab.
3. Expand the **Password Policy** element.

   The list of configured password policies is displayed.
4. Select a password policy to display its properties in the right hand pane.

For a description of all possible properties, and their values, see "Password Policy" in the *Oracle Unified Directory Configuration Reference*.

### 24.5.6 Modify a Password Policy

You can modify a configured password policy by using ODSM, as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager".
2. Select the **Security** tab.
3. Expand the **Password Policy** element.

   The list of configured password policies is displayed.
4. Select the password policy whose properties you want to modify.

For a description of all possible properties, and their values, see "Password Policy" in the *Oracle Unified Directory Configuration Reference*.

### 24.5.7 Create a Password Policy

You can create a new password policy by using ODSM, as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager".
2. Select the **Security** tab.
3. Expand the **Password Policy** element.
4. Click the **Add** icon.
5. On the **Create New Password Policy** screen, configure the required properties.

For a description of all possible properties, and their values, see "Password Policy" in the *Oracle Unified Directory Configuration Reference*.

6. When you have configured the new password policy, click **Create**.

### 24.5.8 Create a Password Policy Based on an Existing Password Policy

You can create a new password policy that is based on an existing password policy by using ODSM, as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager".
2. Select the **Security** tab.
3. Expand the **Password Policy** element.
4. Select the password policy on which you want to base the new policy.
5. Click the **Add like** icon.
6. On the **Create New Password Policy** screen, modify the properties to create the new policy.

   For a description of all possible properties, and their values, see "Password Policy" in the *Oracle Unified Directory Configuration Reference*.
7. When you have configured the new password policy, click **Create**.

### 24.5.9 Delete a Password Policy

You can delete a password policy by using ODSM, as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager".
2. Select the **Security** tab.
3. Expand the **Password Policy** element.
4. Select the password policy that you want to delete.
5. Click the **Delete** icon.
6. Click **OK** to confirm the deletion.

### 24.5.10 Display the Supported Password Storage Schemes

A password storage scheme provides a mechanism for encoding user passwords for storage in the server. In most cases, the password is encoded in a manner that prevents users from determining what the clear-text password is, while still allowing the server to determine whether the user-supplied password is correct. Oracle Unified Directory supports a number of password storage schemes. For more information, see Section D.15.9, "password storage scheme".

You can use ODSM to display the list of password storage schemes, as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager".
2. Select the **Security** tab.
3. Expand the **Password Storage** element.
4. The list of password storage schemes is displayed.

### 24.5.11 Enable or Disable a Password Storage Scheme

You can use ODSM to enable or disable a password storage scheme, as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager".
2. Select the **Security** tab.
3. Expand the **Password Storage** element.
4. Select the password storage scheme that you want to enable or disable.
5. In the right hand pane, check or uncheck the **Enabled** box, as required.
6. Click **Apply** to save your changes.

## 24.6 Password Validators

Password validators provide a mechanism to determine whether a provided plain text password is acceptable for use. Validation prevents users from choosing trivial passwords that are weak and might be easily guessed. Types of validation that might be performed include:

- Ensuring that a password has at least a specified minimum number of characters.
- Ensuring that a password has no more than a specified maximum number of characters.
- Ensuring that a password contains at least a specified number of characters from different character sets (for example, lowercase letters, uppercase letters, numeric digits, and symbols).
- Ensuring that a user is not allowed to re-use a password that has been previously used by that user (that is, that the password is not contained in a history of previous passwords).
- Ensuring that a user is not allowed to choose a password that matches the value of another attribute in the user's entry.
- Ensuring a password is not contained in a specified dictionary.

The password policy for a user specifies the set of password validators that should be used whenever that user provides a new password. To activate a password validator, you must enable the corresponding configuration entry, and include the DN of that entry in the `password-validator` attribute of the password policy in which you want that validator active.

The following password validators are available in the server by default:

- **Attribute Value Password Validator**

  This validator attempts to determine whether a proposed password is acceptable for use by determining whether that password is contained in any attribute within the user's entry.

  The validator can be configured to look in all attributes or in a specified subset of attributes.

- **Character Set Password Validator**

  This validator determines whether a proposed password is acceptable by checking whether it contains a sufficient number of characters from one or more user-defined character sets.

For example, the validator can ensure that passwords must have at least one lowercase letter, one uppercase letter, one digit, and one symbol..

- **Dictionary Password Validator**

  This validator determines whether a proposed password is acceptable based on whether the password value appears in a provided dictionary file.

  A large dictionary file is provided with the server, but you can supply an alternate dictionary. In this case, the dictionary must be a plain-text file with one word per line.

- **Length Based Password Validator**

  This validator determines whether a proposed password is acceptable based on whether the number of characters it contains falls within an acceptable range of values.

  Both upper and lower bounds can be defined.

- **Repeated Characters Password Validator**

  This validator determines whether a proposed password is acceptable based on the number of times any character appears consecutively in a password value.

  It ensures that user passwords do not contain strings of the same character repeated several times, like "aaaaaa" or "aaabbb"..

- **Similarity Based Password Validator**

  This validator determines whether a proposed password is acceptable by measuring how similar it is to the user's current password.

  In particular, it uses the Levenshtein Distance algorithm to determine the minimum number of changes (where a change may be inserting, deleting, or replacing a character) to transform one string into the other. It can be used to prevent users from making only minor changes to their current password when setting a new password. Note that for this password validator to be effective, it is necessary to have access to the user's current password. Therefore, if this password validator is to be enabled, the `password-change-requires-current-password` property in the password policy configuration must also be set to `true`.

- Unique Characters Password Validator

  This validator determines whether a proposed password is acceptable based on the number of unique characters that it contains.

  It can be used to prevent simple passwords that contain only a few characters like "aabbcc" or "abcabc".

## 24.6.1 Managing Password Validators

You can manage password validators by using the `dsconfig` command or by using the ODSM interface, as described in the following sections:

- Section 24.6.1.1, "To Display the Available Password Validators"
- Section 24.6.1.2, "To Display the Properties of a Password Validator"
- Section 24.6.1.3, "To Enable or Disable a Password Validator"
- Section 24.6.1.4, "To Configure the Values of a Password Validator"
- Section 24.6.1.5, "To Associate a Password Validator With a Password Policy"

#### 24.6.1.1 To Display the Available Password Validators

Use the `dsconfig` command to list the password validators that are available, as follows:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  list-password-validators
Password Validator                   : Type                : enabled
-------------------------------------:---------------------:--------
Attribute Value                      : attribute-value     : true
Character Set                        : character-set       : true
Dictionary                           : dictionary          : false
Length-Based Password Validator      : length-based        : true
Repeated Characters                  : repeated-characters : true
Similarity-Based Password Validator  : similarity-based    : true
Unique Characters                    : unique-characters   : true
```

To display the available password validators by using ODSM, do the following:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager".
2. Select the **Security** tab.
3. Expand the **Password Validator** element.

   The available password validators are displayed.

#### 24.6.1.2 To Display the Properties of a Password Validator

Use the `dsconfig` command to display the properties of a particular password validator, as follows:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  get-password-validator-prop --validator-name "Length-Based Password Validator"
Property            : Value(s)
--------------------:---------
enabled             : true
max-password-length : 0
min-password-length : 6
```

To display the properties of a password validator by using ODSM, do the following:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager".
2. Select the **Security** tab.
3. Expand the **Password Validator** element.

   The available password validators are displayed.
4. Click on a password validator to display its properties in the right hand pane.

#### 24.6.1.3 To Enable or Disable a Password Validator

All of the password validators, except the Dictionary validator, are enabled by default. A validator must be enabled before it can be associated with a specific password policy.

Use the `dsconfig` command to set the `enabled` property to `true` or `false`. For example, to disable the Length-Based password validator, set the `enabled` property as follows:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
```

```
set-password-validator-prop --validator-name "Length-Based Password Validator" \
--set enabled:false
```

To enable or disable a password validator by using ODSM, do the following:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager".
2. Select the **Security** tab.
3. Expand the **Password Validator** element.

   The available password validators are displayed.
4. Click on a password validator to display its properties in the right hand pane.
5. Select the **Enabled** check box to enable the validator, or deselect this check box to disable the validator.
6. Click **Apply** to save the configuration changes.

#### 24.6.1.4 To Configure the Values of a Password Validator

Use the `dsconfig` command to configure properties of a password validator. For example, to specify that passwords must be at least eight characters long, set the `min-password-length` property as follows:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  set-password-validator-prop --validator-name "Length-Based Password Validator" \
  --set min-password-length:6
```

To display the properties of a password validator by using ODSM, do the following:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager".
2. Select the **Security** tab.
3. Expand the **Password Validator** element.

   The available password validators are displayed.
4. Click on a password validator to display its properties in the right hand pane.
5. Configure any required properties and click **Apply** to save the configuration change.

#### 24.6.1.5 To Associate a Password Validator With a Password Policy

A password validator is only taken into account when it is associated with a specific password policy.

To associate a password validator with a password policy by using `dsconfig`, set the `password-validator` property of the password policy.

For example, to specify that the default password policy should check whether passwords conform to a specific number of characters, set the `password-validator` property of the default password policy as follows:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  set-password-policy-prop --policy-name "Default Password Policy" \
  --set password-validator:"Length-Based Password Validator"
```

To associate a password validator with a password policy by using ODSM, do the following:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager".
2. Select the **Security** tab.
3. Expand the **Password Policy** element.

   The available password policies are displayed.
4. Click on a password policy to display its properties in the right hand pane.
5. Expand the Syntax element in the right hand pane.
6. From the **Password Validator** list, select the password validators that you want to associate with this password policy.
7. Click **Apply** to save the configuration changes.

## 24.7 Password Generators

Password generators are used to generate passwords for user accounts. A password generator is used in conjunction with the password modify extended operation to provide a new password for cases in which the client did not include a password in its request. If no password generator is associated with the password policy that is in force, the password modify extended operation does not automatically generate passwords for users.

The passwords that are created by a password generator are not subject to validation. You should configure password generators so that the passwords they create are in-line with the requirements of the associated password validators.

By default one password generator is configured on a directory server instance - the random password generator. The following sections describe how to manage password generators by using `dsconfig`.

### 24.7.1 To Display the Configured Password Generators

Use the `dsconfig` command to list the configured password generators, as follows:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  list-password-generators
Password Generator        : Type   : enabled
--------------------------:--------:--------
Random Password Generator : random : true
```

### 24.7.2 To Display the Properties of a Password Generator

Use the `dsconfig` command to display the properties of a password generator, as follows:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  get-password-generator-prop --generator-name "Random Password Generator"
Property               : Value(s)
-----------------------:-------------------------------------------------------
enabled                : true
password-character-set : alpha:abcdefghijklmnopqrstuvwxyz, numeric:0123456789
password-format        : "alpha:3,numeric:2,alpha:3"
```

The password character set is a multi-valued property, with each value defining a different character set. The format of the character set is the name of the set followed by a colon and the characters that are in that set. For example, the value

"alpha:abcdefghijklmnopqrstuvwxyz" defines a character set named "alpha" containing all of the lower-case ASCII alphabetic characters.

The password format is a comma-delimited list of elements in which each of those elements is comprised of the name of a character set defined in the `password-character-set` property, a colon, and the number of characters to include from that set. For example, the default value of "`alpha:3,numeric:2,alpha:3`" generates an 8-character password in which the first three characters are from the "alpha" set, the next two are from the "numeric" set, and the final three are from the "alpha" set.

### 24.7.3 To Enable or Disable a Password Generator

The random password generator is enabled by default. A validator must be enabled before it can be associated with a specific password policy.

Use the `dsconfig` command to set the `enabled` property to `true` or `false`. For example, to disable the random password generator, set the `enabled` property as follows:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  set-password-generator-prop --generator-name "Random Password Generator" \
  --set enabled:false
```

### 24.7.4 To Configure the Values of a Password Generator

Use the `dsconfig` command to configure properties of a password generator. For example, to specify that passwords generated by the random password generator must be of the form, three letters, three numbers, and two defined special characters, set the corresponding properties as follows:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  set-password-generator-prop --generator-name "Random Password Generator" \
  --add password-character-set:special:\!@#\$%^&*\(\)
  --set password-format:alpha:3,numeric:3,special:2
```

### 24.7.5 To Associate a Password Generator With a Password Policy

A password generator is only taken into account when it is associated with a specific password policy.

To associate a password generator with a password policy by using `dsconfig`, set the `password-generator` property of the password policy.

For example, to specify that the default password policy should use a new password generator, named `Special Generator`, set the `password-generator` property of the default password policy as follows:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  set-password-policy-prop --policy-name "Default Password Policy" \
  --set password-generator:"Special Generator"
```

# 25

# Integrating With Oracle's Enterprise User Security

Oracle's Enterprise User Security (EUS) enables you to store user identities in LDAP-compliant directory service for Oracle Database authentication.

Enterprise User Security enables you to centrally manage database users across the enterprise. Enterprise users are created in LDAP-compliant directory service, and can be assigned roles and privileges across various enterprise databases registered with the directory.

Users connect to Oracle Database by providing credentials that are stored in Oracle Unified Directory. The database executes LDAP search operations to query user specific authentication and authorization information.

Integrating Oracle Unified Directory and Enterprise User Security enhances and simplifies your authentication and authorization capabilities by allowing you to leverage user identities stored in LDAP-compliant directory service without any additional synchronization.

This chapter covers the following topics:



## 25.1 Integration Scenarios

To integrate Oracle Unified Directory and Enterprise User Security, you can select one of the following scenarios:

- User identities stored in the Oracle Unified Directory, as described in Section 25.3, "Integrating Oracle Unified Directory with Enterprise User Security".
- User identities stored in an external LDAP-compliant directory service with Oracle Unified Directory used as the proxy server, as described in Section 25.4, "Integrating with Enterprise User Security and an External LDAP Directory".

## 25.2 What's New in this Release

In this release, Oracle Unified Directory support for EUS includes:

- Certificate authentication and integration with Kerberos authentication.

> **Note:** Certificate authentication only supports DN entry matching the DN in the certificate.

- Password Policies: Password policies are a set of rules that apply to all user passwords in an identity management realm. Password policies include settings for password complexity, minimum password length, and the like. They also include account lockout and password expiration settings.

  The password policy entry defined in the LDAP-compliant directory storing storing the user entries can be used by Oracle Database for Enterprise User Security.

  The database communicates with Oracle Unified Directory and requests the Oracle Unified Directory to report any password policy violations. If the database gets a policy violation response from Oracle Unified Directory, then it flashes the appropriate warning or error message to the user.

  The database reports the following events:

  - It flashes a warning when the user password is about to expire and displays the number of days left for the user to change his or her password.
  - It flashes a warning when the password has expired and informs the user about the number of grace logins that remain.
  - It displays an error when the user password has expired and the user does not have any grace logins left.
  - It displays an error when the user account has been locked due to repeated failed attempts at login.
  - It displays an error if the user account has been disabled by the administrator.
  - It displays an error if the user account is inactive.

  Enterprise user login attempts to the database update the user account status in Oracle Unified Directory or any supported external LDAP-compliant directory. For example, consecutive failed login attempts to the database results in the account getting locked in the directory, as per the directory's password policy.
- The following external LDAP-compliant directories are supported:
  - Microsoft Active Directory
  - Novell eDirectory
  - Oracle Directory Server Enterprise Edition
  - Oracle Unified Directory

  > **Note:** You can configure an Oracle Unified Directory instance as an external directory server with another Oracle Unified Directory instance as the proxy server.

For information about configuring Enterprise User Security, see the *Oracle Database Enterprise User Administrator's Guide*.

## 25.3 Integrating Oracle Unified Directory with Enterprise User Security

You can integrate Oracle Unified Directory with Enterprise User Security, where user identities stored in an Oracle Unified Directory without any additional synchronization. To do so, complete the following:

- Configuring Enterprise User Security for an Oracle Unified Directory Server
- Modifying the Oracle Unified Directory Configuration for Enterprise User Security
- Configuring Oracle Database for Oracle Unified Directory

### 25.3.1 Configuring Enterprise User Security for an Oracle Unified Directory Server

You can configure the EUS for an Oracle Unified Directory server using one of the following options:

- Enabling Enterprise User Security During Installation
- Enabling Enterprise User Security With ODSM for an Existing Instance

---

**Notes:**

- If you want to enable EUS during installation then complete the steps described in Enabling Enterprise User Security During Installation.
- If you want to configure EUS for an existing Oracle Unified Directory instance then complete the steps described in Enabling Enterprise User Security With ODSM for an Existing Instance.

---

#### 25.3.1.1 Enabling Enterprise User Security During Installation

You can use this option when you are installing Oracle Unified Directory. Enable the Oracle Unified Directory directory server instance for integration with EUS while you are setting up the server instance, as described in "Setting Up the Directory Server" in the *Oracle Fusion Middleware Installation Guide for Oracle Unified Directory*.

---

**Note:** Ensure that you select **Enable for EUS** in the Oracle Components Integration screen while running the `oud-setup` graphical interface option or if you are running `oud-setup` with the `--cli` option then specify the following option while launching the installer:

```
oud-setup --eus
```

---

#### 25.3.1.2 Enabling Enterprise User Security With ODSM for an Existing Instance

On an existing directory server instance, you can create a new suffix for EUS by using ODSM. There is no command-line equivalent for this functionality.

To create a suffix for EUS by using ODSM, perform the following steps:

1. Ensure that the server instance has an LDAP connection handler that is enabled for SSL

   If SSL is not enabled, add an LDAPS connection handler, as described in Section 14.2, "Managing the Server Configuration With Oracle Directory Services Manager".

2. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager".
3. Select the **Home** tab.
4. Under the Configuration menu, select **Create Local Naming Context**.

   The **New Local Naming Context** window is displayed.
5. Enter the following details:
   a. In the **Base DN** field, type a name for the suffix that you want to create.

      **Note:** You cannot enable EUS on an existing suffix that has already been populated with user data.

   b. From the Directory Data Options group, select one of the following options for populating the suffix with data:

      **Only Create Base Entry** creates the database along with the base entry of the suffix. Any additional entries must be added after suffix creation.

      **Leave Database Empty** creates an empty database. The base entries and any additional entries must be added after suffix creation.

      **Note:** The suffix must contain at least one entry hence do not select the **Leave Database Empty** option.

      **Import Generated Sample Data** populates the suffix with sample entries.

      Specify the number of entries that should be generated in the **Number of User Entries** field. You can import a maximum of 30,000 sample entries through ODSM. If you want to add more than 30,000 entries, you must use the `import-ldif` command.

   c. In the Oracle Components Integration region, select Enable **for Enterprise User Security (EUS)** to enable the new suffix.

      When you select EUS, in addition to creating this suffix, two suffixes are created automatically: `"cn=oracleschemaversion"` and `"cn=oraclecontext."` An EUS workflow element is also added in front of the local backend workflow element. Further, a DN renaming workflow element for `"cn=schema"` is added, so that it can be accessed using the `"cn=subschemasubentry"` DN.

   d. In the Network Group region, attach the suffix to at least one network group by performing the following steps:

      – To attach the suffix to an existing network group, select **Use Existing** and select the required network group from the list.

      – To attach the suffix to a new network group, select **Create New** and then in the Name field, type a name for the network group you want to create.

      You can attach several network groups to the same suffix.

   e. In the Workflow Element region, attach the suffix to the workflow element by performing either of the following steps:

      – To attach the suffix to an existing workflow element, select **Use Existing** and then select the required workflow element from the list.

- To attach the suffix to a new workflow element, select **Create New** and then in the **Name** field, type a name for the workflow element you want to create.

**f.** Click **Create**.

The following confirmation message is displayed:

```
Configuration created successfully.
```

## 25.3.2 Modifying the Oracle Unified Directory Configuration for Enterprise User Security

After OUD has been enabled for EUS, you must update the realm information in the OUD configuration by performing the following steps:

1. Locate the LDIF template file at `install_dir/config/EUS/modifyRealm.ldif`.
2. Edit the `modifyRealm.ldif` file as follows:
   - Replace `dc=example,dc=com` with the correct naming context for your server instance.
   - Replace `ou=people` and `ou=groups` with the correct location of the user and group entries in your DIT.
3. Use the `ldapmodify` command to update the configuration with the edited LDIF template file, for example:

```
$ ldapmodify -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -v -f
modifyRealm.ldif
```

## 25.3.3 Configuring Oracle Database for Oracle Unified Directory

You must configure the Oracle Database for Enterprise User Security by completing the following steps:

- Configuring the Database
- Registering Your Database

### 25.3.3.1 Configuring the Database

Run Net Configuration Assistant (NetCA) tool to configure the Oracle Unified Directory host name and port numbers for the database.

To configure your database for directory usage:

1. Start Oracle Net Configuration Assistant:

   **Unix**

   Run `netca` (Located at `$ORACLE_HOME/bin`) on the command line.

   **Windows**

   Choose **Start**, **Programs**, **Oracle-HOME_NAME**, **Configuration and Migration Tools**, and select **Net Configuration Assistant**.

   The **Oracle Net Configuration Assistant: Welcome** screen is displayed.

2. Select **Directory Service Usage Configuration** and click **Next**.

   The **Oracle Net Configuration Assistant: Directory Usage Configuration, Directory Type** screen is displayed.

3. Select **Oracle Unified Directory** as the directory type and click **Next**.

   The **Oracle Net Configuration Assistant: Directory Usage Configuration, Directory Location** screen is displayed.

4. Enter the following details:

   – **Hostname**: Enter the name of the host on which the Oracle Unified Directory server is running.

   – **Port**: Enter the Oracle Unified Directory port number.

   – **SSL Port**: Enter the Oracle Unified Directory SSL port number.

   Click **Next**.

   The **Oracle Net Configuration Assistant: Directory Usage Configuration, Select OracleContext** screen is displayed.

5. Select the default Oracle Context to use. You need to select this if there are multiple Oracle Unified Directory realms on the directory server. Click **Next**.

   The **Directory Usage Configuration, Done** screen is displayed.

6. Confirm that the directory usage configuration is successfully completed. Click **Next**.

7. Click **Finish**.

NetCA creates an ldap.ora file in the `$ORACLE_HOME/network/admin` directory (Unix) or `ORACLE_HOME\network\admin` directory (Windows). The file stores the connection information details about the directory.

#### 25.3.3.2 Registering Your Database

Register the database with the directory service. The Database Configuration Assistant (DBCA) tool enables you to register the database with Oracle Unified Directory.

To register the database with the directory:

1. Start DBCA using the `dbca` command.

   – On Unix systems, you can start DBCA using the following command:

   ```
   $ORACLE_HOME/bin/dbca
   ```

   – On Windows, you can also start DBCA from the Start menu:

   Click **Start**, **All Programs**, **Oracle - OracleHomeName**, **Configuration and Migration Tools**, and then select **Database Configuration Assistant**.

   The Welcome screen is displayed.

2. Click **Next**.

   The Operations screen is displayed.

3. Select **Configure Database Options**.

   Click **Next**.

   The Database screen is displayed.

4. Select the database name that you wish to configure. You might also be asked to enter `SYS` user credentials if you are not using operating system authentication.

   Click **Next**.

   The Management Options screen is displayed.

5. Select **Keep the database configured with Database Control** if you want to continue using Database Control to manage the database. You also have the option of using Grid Control to manage the database.

   Click **Next**.

   The Security Settings screen is displayed.

6. Select **Keep the enhanced 11g default security settings** to keep the 11*g* security settings.

   Click **Next**.

   The Network Configuration screen is displayed.

7. Select **Yes**, register the database to register the database with the directory. Enter the distinguished name (DN) of a user who is authorized to register databases in Oracle Unified Directory. Also, enter the password for the directory user. Enter a wallet password. Reenter the password in the **Confirm Password** field.

   Click **Next**.

   ---

   **Note:** The database uses a randomly generated password to log in to the directory. This database password is stored in an Oracle wallet. The wallet can also be used to store certificates needed for SSL connections.

   The wallet password that you specify is different from the database password. The wallet password is used to protect the wallet.

   ---

   The Database Components screen is displayed.

8. Click **Next**.

   The Connection Mode page is displayed.

9. Select **Dedicated Server Mode** or **Shared Server Mode**.

   Click **Finish**.

   The Confirmation dialog box is displayed.

10. Click **OK**.

    ---

    **Note:** After you register the database with the directory, make sure that auto login is enabled for the database wallet. The default wallet is created in the `$ORACLE_BASE/admin/database_sid/wallet` directory (Unix) or `ORACLE_BASE\admin\database_sid\wallet` directory (Windows).

    You can verify that auto login for the wallet is enabled by checking for the presence of the `cwallet.sso` file in the wallet directory. If the file is not present, you can enable auto login by opening the wallet using Oracle Wallet Manager, and using the option to enable auto login for the wallet.

    ---

## 25.4 Integrating with Enterprise User Security and an External LDAP Directory

Integrating Oracle Unified Directory and Enterprise User Security (EUS) enhances and simplifies your authentication and authorization capabilities by allowing you to centralize user identities stored in an external LDAP repository without any additional synchronization.

You can integrate EUS with an external LDAP directory, if the Oracle Unified Directory is configured as a proxy front ending an external LDAP repository. The EUS configuration details are stored locally in Oracle Unified Directory and the remote external LDAP directory contains only the Enterprise Users and the Enterprise Groups details.

This section describes how to integrate Oracle Unified Directory with Oracle Enterprise User Security and contains the following sections:

- Configuring External Directories for the Integration
- Configuring Oracle Unified Directory for the Integration

> **Note:** Create a back-up copy of the `ORACLE_HOME/config/eus/` directory (Unix) or `ORACLE_HOME\config\eus\` directory (Windows). All the configuration files required for the Enterprise User Security integration are in the `eus` directory. Making a back-up copy of the `eus` directory enables you to edit the template-like files in the original `eus` directory based on your environment, and still keep copies of the original files.

### 25.4.1 Configuring External Directories for the Integration

This section contains instructions for integrating Oracle Unified Directory with Enterprise User Security for use with specific external directories.

These instructions are organized by external directory type into the following sections:

- User Identities in Microsoft Active Directory
- User Identities in Oracle Directory Server Enterprise Edition
- User Identities in Novell eDirectory
- User Identities in Oracle Unified Directory

> **Note:** Back-end LDAP schema extensions are no longer required for any of these external directories, *except* Microsoft Active Directory. These changes are now done in the Oracle Unified Directory local store.
>
> Only a single, minimal schema change to add the `orclCommonAttribute` attribute definition is necessary for Active Directory.

#### 25.4.1.1 User Identities in Microsoft Active Directory

Perform the following procedures to integrate Oracle Unified Directory with Enterprise User Security for user identities stored in Active Directory:

1. Make a back-up copy of your Active Directory image. The schema extensions inside of Active Directory are permanent and cannot be canceled. The back-up image enables you to restore all your changes if required.

2. Execute the following command to load the Enterprise User Security required schema, `ExtendAD`, into Active Directory using the Java classes included in Oracle Unified Directory.

   The `ExtendAD` file is located in the *$ORACLE_HOME*`/config/EUS/ActiveDirectory/` directory (Unix) or *ORACLE_HOME*`\config\EUS\ActiveDirectory\` directory (Windows). You can use the **`java`** executable in the *ORACLE_HOME*`/jdk/bin` directory.

   ```
   java ExtendAD -h Active_Directory_Host_Name -p Active_Directory_Port
   -D Active_Directory_Admin_DN -w Active_Directory_Admin_Password
   -AD Active_Directory_Domain_DN -commonattr
   ```

   Example:

   ```
   java ExtendAD -h myhost -p 389 -D cn=administrator,cn=users,dc=example,dc=com
   -w <pwd> -AD dc=example,dc=com -commonattr
   ```

3. Install the Oracle Unified Directory Password Change Notification plug-in, `oidpwdcn.dll`, by performing the following steps:

   a. Complete the following depending on your Windows:

      **Windows 32-bit**

      Copy *OUD_HOME*`\config\EUS\ActiveDirectory\win\oidpwdcn.dll` file to the Active Directory `WINDOWS\system32` directory.

      **Windows 64-bit**

      Copy *OUD_HOME*`\config\EUS\ActiveDirectory\win64\oidpwdcn.dll` file to the Active Directory `WINDOWS\system64` directory.

   b. Use `regedt32` or `regedt64` to edit the registry and enable the `oidpwdcn.dll`. Start `regedt32` by entering **`regedt32`** at the command prompt.

   c. Add **`oidpwdcn`** to the end of the Notification Packages entry in the `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\` registry, for example:

      ```
      RASSFM
      KDCSVC
      WDIGEST
      scecli
      oidpwdcn
      ```

      This enables the password DLL and populates `orclCommonAttribute` attribute with the password verifier required by EUS.

   d. Restart the Active Directory system after making these changes.

4. Run `ldapmodify`: to allow anonymous logins on Active Directory:

   ```
   ldapmodify -h <ADhost> -p <AD port> -D <AD dirmgr> -w <pwd>
   dn: cn=directory service,cn=windows
   nt,cn=services,cn=configuration,dc=example,dc=com
   changetype: modify
   replace: dsHeuristics
   dsHeuristics: 0000002
   ```

**Note:** Ensure that you replace `dc=example,dc=com` with the base DN of your Active Directory server.

5. Reset the password for all the Active Directory users, allowing the plug-in to acquire the password changes and generate and store password verifiers.
6. Verify the Active Directory setup by performing the following steps:
   a. Change the password of an Active Directory user.
   b. Search Active Directory for the user you changed the password for. Verify the `orclCommonAttribute` attribute contains the generated hash password value.

      This value adds the `orclCommonAttribute` attribute definition in Active Directory.
7. Complete the integration by performing the task described in Configuring Oracle Unified Directory for the Integration.

#### 25.4.1.2 User Identities in Oracle Directory Server Enterprise Edition

Perform the following procedures to integrate Oracle Unified Directory with Enterprise User Security for user identities stored in Oracle Directory Server Enterprise Edition:

1. Run `ldapmodify` command from Oracle Directory Server Enterprise Edition to enable extended operation for the account lock, as follows:

```
ldapmodify -h <ODSEE Server> -p <ODSEE port> -D <ODSEE Admin ID> -w <ODSEE
Admin password>
dn: oid=1.3.6.1.4.1.42.2.27.9.6.25,cn=features,cn=config
changetype: add
objectclass: directoryServerFeature
oid: 1.3.6.1.4.1.42.2.27.9.6.25
cn: Password Policy Account Management
```

2. Complete the integration by performing the task described in Configuring Oracle Unified Directory for the Integration.

#### 25.4.1.3 User Identities in Novell eDirectory

Perform the following procedures to integrate Oracle Unified Directory with Enterprise User Security for user identities stored in Novell eDirectory:

1. To configure Novell eDirectory for the integration, enable Universal Password in eDirectory and allow the administrator to retrieve the user password. Refer to Novell's eDirectory documentation on Password Management for more information.
2. Complete the integration by performing the task described in Configuring Oracle Unified Directory for the Integration.

#### 25.4.1.4 User Identities in Oracle Unified Directory

You can configure an Oracle Unified Directory instance as an external directory server with another Oracle Unified Directory instance as the proxy server. In this scenario, the EUS configuration details are stored locally in Oracle Unified Directory proxy server and the external Oracle Unified Directory contains only the Enterprise Users and the Enterprise Groups details.

To do so, you must modify the default password policy to use Salted SHA-1 as password storage scheme by running `dsconfig` command as follows:

```
dsconfig -h <OUD host> -p <OUD admin port> -D <OUD dirmgr> -j <pwdfile> -X -n
set-password-policy-prop --policy-name "Default Password Policy" --set
default-password-storage-scheme:"Salted SHA-1"
```

**Note:** Ensure that you modify the default password policy of the Oracle Unified Directory containing the Enterprise Users and the Enterprise Groups details. Do not modify the default password policy of the Oracle Unified Directory instance acting as the proxy server.

## 25.4.2 Configuring Oracle Unified Directory for the Integration

Configure Oracle Unified Directory with external LDAP Directories by performing the following steps:

- Configuring Enterprise User Security for an Oracle Unified Directory Proxy Server
- Performing Post Configuration Steps
- Modifying the Oracle Unified Directory Proxy Server Configuration for Enterprise User Security

### 25.4.2.1 Configuring Enterprise User Security for an Oracle Unified Directory Proxy Server

You can configure the EUS for an Oracle Unified Directory proxy servers using one of the following options:

- Enabling Enterprise User Security for a Proxy Server During Installation
- Enabling Enterprise User Security for an Existing Proxy Server Instance

**Notes:**

- If you want to enable EUS during installation then complete the steps described in Enabling Enterprise User Security for a Proxy Server During Installation.
- If you want to configure EUS for an existing Oracle Unified Directory instance then complete the steps described in Enabling Enterprise User Security for an Existing Proxy Server Instance.

#### 25.4.2.1.1 Enabling Enterprise User Security for a Proxy Server During Installation

You can enable an Oracle Unified Directory directory server instance for integration with EUS while you are setting up the server instance, as described in "Setting Up the Proxy Server" in the *Oracle Fusion Middleware Installation Guide for Oracle Unified Directory*.

**Notes:**

- Ensure that you select **Configure EUS** in the Deployment Options screen while running the `oud-proxy-setup` graphical interface or if you are running `oud-proxy-setup` with the --cli option then specify the following option while launching the installer:

  ```
  oud-proxy-setup --eusContext {namingContext}
  ```

- If you are running `oud-proxy-setup` with the `--cli` option then you must manually configure LDAP server extension, proxy workflow element and EUS workflow element using `dsconfig` command. In a graphical interface these configurations are automatically configured.
- For Novell eDirectory, enter the LDAPS port of the Oracle Unified Directory proxy server.

#### 25.4.2.1.2 Enabling Enterprise User Security for an Existing Proxy Server Instance

To configure Enterprise User Security for an existing Oracle Unified Directory Proxy Server instance, complete the following steps:

1. Ensure that the server instance has an LDAP connection handler that is enabled for SSL

   If SSL is not enabled, add an LDAPS connection handler, as described in Section 14.2, "Managing the Server Configuration With Oracle Directory Services Manager".

2. Connect to the proxy server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager".
3. Select the **Home** tab.
4. Under the Configuration menu, select **Create Remote EUS Naming Context**.

   The **Create Remote EUS Naming Context** window is displayed.

5. Enter the following details:
   - **Base DN**: Enter the name for the suffix.
   - **Network Group**: Select the network group attached to the suffix.
   - **Server Type**: Select the server containing the EUS user entries.
   - **Host Name**: Enter the host name of the remote server.
   - **Ports Available**: Enter the LDAP port, LDAPS port, or LDAP and LDAPS ports of the remote server.

     **Note:** For Novell eDirectory, enter the LDAPS port of the Oracle Unified Directory proxy server.

   - **Trust All**: Select this check box to trust all the certificates presented by the remote server.
   - **Trust Manager**: Select the trust manager that the server will use when connecting to the LDAPS ports of the remote server to forward requests.
6. Click **Create**.

The following confirmation message is displayed:

```
Configuration created successfully.
```

#### 25.4.2.2 Performing Post Configuration Steps

After completing the required configuration as described in Section 25.4.2.1, "Configuring Enterprise User Security for an Oracle Unified Directory Proxy Server", you must perform the following:

1. Configure the proxy workflow elements, remote root DN and remote root user accounts for the external LDAP-compliant directories by running the dsconfig command as follows:

```
dsconfig set-workflow-element-prop \
          --element-name proxy-we1 \
          --set remote-root-dn:cn=administrator,cn=users,dc=example,dc=com \
          --set remote-root-password:******** \
          --hostname localhost \
          --port 4444 \
          --trustAll \
          --bindDN cn=directory\ manager \
          --bindPasswordFile pwd.txt \
          --no-prompt
```

2. You can configure the proxy workflow elements for the external LDAP-compliant directories with use-client-identity by defining the exclude-list, remote ldap server bind dn and remote ldap server bind password. When the EUS is enabled, the database connects with its own credentials and performs searches on the external LDAP server. As the DB entry is stored locally on OUD proxy, it uses an alternate ID to bind to the external LDAP server as the database entry does not exist on the external LDAP server.

```
dsconfig set-workflow-element-prop \
          --element-name proxy-we1 \
          --add exclude-list:cn=directory\ manager \
          --add exclude-list:cn=oraclecontext,dc=example,dc=com \
          --set
remote-ldap-server-bind-dn:cn=administrator,cn=users,dc=example,dc=com \
          --set remote-ldap-server-bind-password:******** \
          --hostname localhost \
          --port 4444 \
          --trustAll \
          --bindDN cn=directory\ manager \
          --bindPasswordFile pwd.txt \
          --no-prompt
```

#### 25.4.2.3 Modifying the Oracle Unified Directory Proxy Server Configuration for Enterprise User Security

After OUD has been enabled for EUS, you must update the realm information in the OUD configuration by performing the following steps:

1. Locate the LDIF template file at `install_dir/config/EUS/modifyRealm.ldif`.
2. Edit the `modifyRealm.ldif` file as follows:
   - Replace `dc=example,dc=com` with the correct naming context for your server instance.

- Replace `ou=people` and `ou=groups` with the correct location of the user and group entries in your DIT.

3. Use the `ldapmodify` command to update the configuration with the edited LDIF template file, for example:

```
$ ldapmodify -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -v -f
modifyRealm.ldif
```

#### 25.4.2.4 Configuring Oracle Database for Oracle Unified Directory Proxy Server

You must configure the Oracle Database, as described in Configuring Oracle Database for Oracle Unified Directory.

# Part V

# Advanced Administration: Data Replication, Schema Management, and Moving Across Environments

This part describes how to configure, monitor and troubleshoot data replication, to manage the schema, and to move server instances from a test environment to a production environment.

The part includes the following chapters:

# 26

# Replicating Directory Data

Replication enables copies of identical data to be available across multiple servers. Oracle Unified Directory uses a multi-master replication model, which means that all the directory servers within a replication topology can accept read and write operations.

The multi-master replication model is *loosely consistent* by default. This means that changes made on one server are replayed asynchronously to the other servers in the topology. The same entries can be modified simultaneously on different servers. When updates are sent between the two servers, any conflicting changes must be resolved. Various attributes of a WAN, such as latency, can increase the chance of replication conflicts. Conflict resolution generally occurs automatically. A number of conflict rules determine which change takes precedence. In some cases conflicts must be resolved manually.

> **Note:** In certain deployment scenarios, the default loose consistency model might not be adequate. In these situations, you can configure replication to function in *assured* mode. For more information, see Section 26.3.9, "Configuring Assured Replication".

Replication always occurs over a secure connection. Both parties of a replication session must authenticate to the other using SSL certificates. No access control or privileges are enforced. The following sections describe how to configure replication in the directory server.

For information about the mechanics of the replication process see Chapter 6, "Understanding the Oracle Unified Directory Replication Model".

This chapter covers the following topics:

- Section 26.1, "Configuring Data Replication With `dsreplication`"
- Section 26.2, "Configuring Large Replication Topologies"
- Section 26.3, "Modifying the Replication Configuration With `dsconfig`"
- Section 26.4, "Initializing a Replicated Server With Data"
- Section 26.5, "Using the External Change Log"
- Section 26.6, "Configuring Schema Replication"
- Section 26.7, "Replicating to a Read-Only Server"
- Section 26.8, "Detecting and Resolving Replication Inconsistencies"
- Section 26.9, "Purging Historical Replication Data"

- Section 26.10, "Using Isolated Replicas"
- Section 26.11, "Replicating Between Oracle Directory Server Enterprise Edition and Oracle Unified Directory"

# 26.1 Configuring Data Replication With dsreplication

You can set up replication automatically using the graphical setup utility when you first install Oracle Unified Directory, if you configure all of the directory servers in the same manner. You cannot use the setup command to configure replication in command-line mode. If you set up your directory servers by using the setup command, you must use the dsreplication command to configure replication between the servers.

dsreplication accesses the server configuration over SSL through the administration connector. For more information, see Section 14.3, "Managing Administration Traffic to the Server".

In any topology, you should have two replication servers for availability, in case one replication server fails. Replication servers are responsible for keeping track of all changes in the environment. Each replication server contains a list of all other replication servers in the topology.

The examples in this section assume that you have already installed two directory servers and populated one with data. The directory servers can be installed on the same host machine, but if they are, they must have different port numbers.

## 26.1.1 To Enable Replication Between Two Servers

You cannot run more than one instance of the dsreplication enable command to set up replication between multiple servers in parallel. Rather, run the dsreplication enable command successively for each pair of replicated servers in the topology.

To enable replication, use the dsreplication enable command.

The following command enables replication of the data under "dc=example,dc=com" between two directory servers, host1 and host2. Both servers use the default administration port (4444). The command creates a replication server instance on host1, port 8989, and a second replication server instance on host2, port 8989.

```
$ dsreplication enable
  --host1 host1 --port1 4444 --bindDN1 "cn=Directory Manager" \
  --bindPasswordFile1 pwd.txt --replicationPort1 8989 \
  --host2 host2 --port2 4444 --bindDN2 "cn=Directory Manager" \
  --bindPasswordFile2 pwd.txt --replicationPort2 8989 \
  --adminUID admin --adminPasswordFile pwd.txt --baseDN "dc=example,dc=com" -X -n
```

The --adminUID and --adminPasswordFile options refer to the Global Administrator for the replication domain. For more information, see Section 23.6, "Managing Global Administrators". The -X option specifies that all server certificates should be trusted and the -n (--no-prompt) option specifies that the command should be run in non-interactive mode. For information about all the global options for the dsreplication command, type dsreplication -help at the command-line.

#### 26.1.1.1 Controlling Where Replication Servers are Created

Using `dsreplication enable` between two servers automatically configures a replication server on each host. You might want to configure replication between two directory servers without creating a replication server on each host. Use the `--noReplicationServer1` or `--noReplicationServer2` options to add a directory server to a topology without creating an additional replication server. Remember that a replicated topology must contain at least two replication servers to avoid a single point of failure.

You can also enable replication between two servers and specify that one of the servers should only contain a replication server (not a directory server). Use the `--onlyReplicationServer1` or `--onlyReplicationServer2` options to achieve this. Specifying this option will configure a change log and replication port on the server the server will not contain replicated data.

### 26.1.2 To Initialize a Replicated Server

To initialize a replicated server with the data from another replicated server, use the `dsreplication initialize` command.

The following command initializes the base DN `"dc=example,dc=com"` on host2 with the data contained on host1:

```
$ dsreplication initialize --baseDN "dc=example,dc=com" \
  --adminUID admin --adminPasswordFile pwd.txt \
  --hostSource host1 --portSource 4444 \
  --hostDestination host2 --portDestination 4444 -X -n
```

### 26.1.3 To Initialize an Entire Topology

If there are more than two directory servers in the topology, use the `dsreplication intialize-all` command to initialize all replicas simultaneously.

This command takes the details of the source host as arguments, and initializes all other servers for which replication is enabled.

The following command initializes all servers on which replication is enabled, from the contents of the base DN `"dc=example,dc=com"` on host1:

```
$ dsreplication initialize-all --hostname host1 --port 4444 \
  --baseDN "dc=example,dc=com" --adminUID admin --adminPasswordFile pwd.txt
```

### 26.1.4 To Test Replication

The easiest way to test that replication is working is to apply changes on one directory server and to check that those changes have been replicated on another directory server. To test the replication topology set up in the previous procedures, do the following:

1. Use `ldapmodify` to change an entry on host1.
2. Use `ldapsearch` to verify that the change was propagated to host2.

### 26.1.5 To Obtain the Status of a Replicated Topology

You can use the connection details of any directory server in the topology to obtain the status of the entire topology.

Use the `dsreplication status` command to display a list of the directory servers in the topology, along with any missing changes between those servers.

The following command displays the status of the topology set up in the previous procedures:

```
$ dsreplication status --adminUID admin --adminPasswordFile pwd.txt -X \
  --hostname host1 --port 4444
```

## 26.1.6 To Merge Two Existing Replicated Topologies

You can merge two replicated topologies by enabling replication between one server of each topology.

Note the following limitations:

- All of the servers in both topologies must be up and running when you perform the merge.

  If a server its offline, dsreplication cannot update its configuration. If a server is offline when a merge is done, that server will not include the references to the replication servers in the other topology when it comes back online.

- The merge cannot be performed if there are conflicting domain IDs or replication server IDs between the two topologies.

  That is, a server in topology A cannot have the same replication server ID or domain ID as a server in topology B.

  If there are conflicting IDs, the ID of the first server (--host1) is used to resolve the conflict. You must then re-initialize any servers that are out of date, using a server from the same topology as --host1 as the source.

- Both replication topologies must have the same global administrators defined.

1. To merge two replicated topologies, use the dsreplication enable command.

   For example, if you have a replicated topology (topology A) that includes host1, host2 and host3 and a replicated topology (topology B) that includes host4, host5, and host6, the following command effectively merges the two topologies:

   ```
   $ dsreplication enable \
     --host1 host1 --port1 4444 --bindDN1 "cn=Directory Manager" \
     --bindPasswordFile1 pwd.txt --replicationPort1 8989 \
     --host2 host4 --port2 4444 --bindDN2 "cn=Directory Manager" \
     --bindPasswordFile2 pwd.txt --replicationPort2 8989 \
     --adminUID admin --adminPasswordFile pwd.txt --baseDN "dc=example,dc=com" \
     -X -n
   ```

   This example assumes that both the hosts (host1 and host4) include a directory server and a replication server. If they do not, a directory server or replication server is automatically configured.

2. To ensure high availability, you must perform the following steps on all servers that were offline or unavailable during a merge:

   a. Initialize the contents of the suffix cn=admin data by using dsreplication enable

      You can initialize the servers individually, using one of the servers that was available during the merge, or you can use dsreplication initialize-all.

   b. Use the dsconfig command to update the list of replication servers.

### 26.1.7 To Disable Replication For a Specific Replication Domain

1. To disable replication on a specific domain, use the `dsreplication disable` command.

   The following command disables replication of the data under `"dc=example,dc=com"`.

   ```
   $ dsreplication disable --hostname host1 --port 4444 --adminUID admin \
     --adminPasswordFile pwd.txt --baseDN "dc=example,dc=com" -X -n
   ```

   This command removes the replication configuration from the directory server for that domain. If the domain that is disabled is the only replicated domain on this directory server instance, the command also disables the replication server on that instance. If the replication server is disabled, other directory servers that were connected to that replication server are disconnected and automatically reconnect to another replication server in the topology.

2. To disable the replication server itself (including the change log and the replication port) use the following command:

   ```
   $ dsreplication disable --hostname host1 --port 4444 -X -n \
     --adminUID admin --adminPasswordFile pwd.txt --baseDN "dc=example,dc=com" \
     --disableReplicationServer
   ```

   When the replication server is disabled, other directory servers that were connected to that replication server are disconnected and automatically reconnect to another replication server in the topology.

#### 26.1.7.1 Notes About Disabling the Replication Server

Disabling a replication server deletes the replication configuration but does *not* delete the replication server databases. You can therefore retrieve replication changes in the event that the replication server was disabled in error. If you have no requirement for re-enabling replication on this suffix, remove the replication server databases manually, for example: `$rm changelogDB/*`.

If replication is disabled, and then re-enabled, any changes made on that server in the interim are not replicated. You must therefore either forbid changes on the server on which replication is disabled (for the period that replication is disabled) or resynchronize the rest of the topology from that server in the event that changes have occurred.

## 26.2 Configuring Large Replication Topologies

In particularly large topologies, it is often simpler to configure dedicated replication servers (servers that do not include a directory server) and dedicated directory servers (servers that do not included a replication server).

A dedicated directory server contains replicated data but does not contain a change log with the modifications made to that replicated data. A dedicated directory server also has no configured replication port. A dedicated replication server has a configured replication port. The server does not contain replicated data but does contain a change log with the modifications made to the replicated data on other servers in the topology.

---

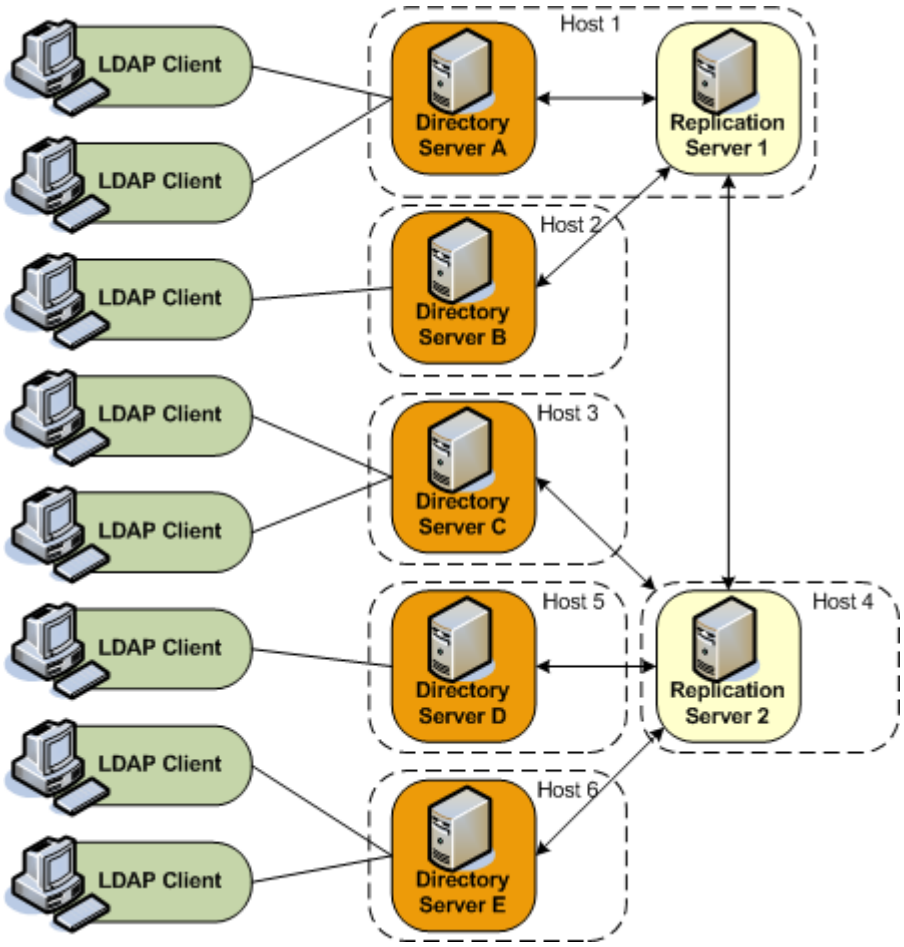
**Note:** Each topology must have at least two replication servers to avoid a single point of failure.

---

For more information and sample topologies, see Chapter 2, "Example Deployments Using the Directory Server".

The following diagram illustrates a large replication topology with one dedicated replication server (Replication Server 2), four dedicated directory servers, and one server that contains both a replication server and a directory server (Host 1).

***Figure 26–1 Large Replicated Topology***




## 26.2.1 To Configure a Dedicated Replication Server

To configure a dedicated replication server, use the `--onlyReplicationServer1` or `--onlyReplicationServer2` option when you enable replication between two servers.

The following example configures replication between Directory Server C and Replication Server 2 in the previous illustration.

```
$ dsreplication enable \
  --host1 host3 --port1 4444 --bindDN1 "cn=Directory Manager" \
  --bindPasswordFile1 pwd.txt --noReplicationServer1 \
  --host2 host4 --port2 4444 --bindDN2 "cn=Directory Manager" \
  --bindPasswordFile2 pwd.txt --onlyReplicationServer2 \
  --replicationPort2 8989 --adminUID admin --adminPasswordFile pwd.txt \
  --baseDN "dc=example,dc=com" -X -n
```

## 26.3 Modifying the Replication Configuration With dsconfig

This section describes how to change certain advanced properties of a replication configuration by using the dsconfig command. Advanced properties are usually optional, or have a default value that is acceptable in most cases. For general information about using dsconfig, see Section 14.1, "Managing the Server Configuration With dsconfig".

You cannot use dsconfig to set up replication between directory servers. Replication can be set up automatically using the GUI install utility, or manually, using the dsreplication command. For more information, see Section 26.1, "Configuring Data Replication With dsreplication".

This section covers the following topics:

- Section 26.3.1, "Retrieving the Replication Domain Name"
- Section 26.3.2, "Changing the Replication Purge Delay"
- Section 26.3.3, "Changing the Window Size"
- Section 26.3.4, "Changing the Initialization Window Size"
- Section 26.3.5, "Changing the Heartbeat Interval"
- Section 26.3.6, "Changing the Isolation Policy"
- Section 26.3.7, "Configuring Encrypted Replication"
- Section 26.3.8, "Configuring Replication Groups"
- Section 26.3.9, "Configuring Assured Replication"
- Section 26.3.10, "Configuring Fractional Replication"
- Section 26.3.11, "Configuring Replication Status"
- Section 26.3.12, "Configuring the Replication Server Weight"

### 26.3.1 Retrieving the Replication Domain Name

The *replication domain name* is generated by the directory server and includes the base DN and a numeric unique identifier.

To obtain a list of the configured replication domains, use the list-replication-domains subcommand. For example:

```
$ dsconfig -h host1 -p 4444 -D "cn=directory manager" -j pwd-file -n list-replication-domains \
   --provider-name "Multimaster Synchronization"

Replication Domain : Type    : server-id : replication-server     : base-dn
-------------------:---------:-----------:-----------------------:--------------------
cn=admin data      : generic : 13981     : host1:8989, host2:8989 : cn=admin data
cn=schema          : generic : 20284     : host1:8989, host2:8989 : cn=schema
dc=example,dc=com  : generic : 26560     : host1:8989, host2:8989 : "dc=example,dc=com"
```

### 26.3.2 Changing the Replication Purge Delay

The replication changes database maintains a record of updates, which might or might not have been replicated. The replication purge delay is a property of the replication server, and specifies the period of time after which internal purge operations are performed on the replication changes database.

#### 26.3.2.1 How Replication Changes Are Purged

Any change that is older than the purge delay is removed from the replication changes database, irrespective of whether that change has been applied. The default purge delay is one day. If the replication changes database is backed up less frequently than the purge delay, changes will be cleared before the changes database has been backed up. Changes can therefore be lost if you use the backup to restore data.

#### 26.3.2.2 To Change the Replication Purge Delay

1. Display the current value of the replication purge delay.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -n \
  get-replication-server-prop \
  --provider-name "Multimaster Synchronization" --advanced \
  --property replication-purge-delay

Property               : Value(s)
-----------------------:---------
replication-purge-delay : 1 d
```

2. Change the purge delay.

   The following command changes the purge delay to one week:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -n \
  set-replication-server-prop \
  --provider-name "Multimaster Synchronization" \
  --set replication-purge-delay:1w
```

### 26.3.3 Changing the Window Size

The window size is a property of the replication server and specifies the number of change requests that are sent to directory servers, without the replication server having to wait for an acknowledgment from the directory server before continuing.

The window size represents the maximum number of update messages that can be sent without immediate acknowledgment from the directory server. It is more efficient to send many messages in quick succession instead of waiting for an acknowledgment after each one. Using the appropriate window size, you can eliminate the time replication servers spend waiting for acknowledgments to arrive. The default window size is 100. If you notice that some directory servers are lagging behind in terms of replicated changes, increase the window size to a higher value and check replication performance again before making further adjustments.

#### 26.3.3.1 To Change the Window Size

1. Display the current value of the window size:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  get-replication-server-prop --provider-name "Multimaster Synchronization" \
   --advanced --property window-size

Property    : Value(s)
------------:---------
window-size : 100
```

2. Change the window size.

   The following command changes the window size to 200.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
```

```
set-replication-server-prop \
--provider-name "Multimaster Synchronization" --set window-size:200
```

## 26.3.4 Changing the Initialization Window Size

During a data import in a replicated topology, it can occur that the importing server is too slow to keep up with the data that is sent by the exporting server. The importing server can therefore block not only the import, but can also stop any other replication changes from being propagated by the exporting server.

An initialization window size enables an exporting server to detect acknowledgements from the slowest importing server and to send data on the replication network *only* when the slow importer is available to receive them.

The initialization window size is set to `100` by default. If there are no slow servers in your topology, you can increase the initialization window size so that exporting servers send more updates before waiting for an acknowledgement. If your topology includes a particularly slow server, you can decrease the initialization window size to ensure that replication is not blocked by this server.

### 26.3.4.1 To Change the Initialization Window Size

1. Display the current value of the initialization window size:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  get-replication-domain-prop --provider-name "Multimaster Synchronization" \
  --domain-name dc=example,dc=com --advanced --property
initialization-window-size
Property                  : Value(s)
--------------------------:---------
initialization-window-size : 100
```

2. Change the initialization window size.

   The following command changes the initialization window size to 50.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -n \
  set-replication-domain-prop --provider-name "Multimaster Synchronization" \
  --domain-name dc=example,dc=com --set initialization-window-size:50
```

## 26.3.5 Changing the Heartbeat Interval

The heartbeat interval is a property of the replication domain and specifies the frequency with which the replication domain communicates with the replication server. The replication domain expects a regular heartbeat at this interval from the replication server. If the heartbeat is not received, the domain closes its connection and connects to another replication server in the topology.

The default heartbeat interval is ten seconds. If replication is running over a WAN or a network with slow response times, you might want to increase the heartbeat interval. In addition, if you observe an error similar to the following in the logs, it is probably necessary to increase the heartbeat interval.

```
[26/May/2011:16:32:50 +0200] category=SYNC severity=NOTICE msgID=15138913
 msg=Replication Heartbeat Monitor on RS rserver/192.157.197.62:8989 30382 for
 dc=example,dc=com in DS 10879 is closing the session because it could not
 detect a heartbeat
```

The heartbeat interval is sensitive to the settings of your JVM. If you require a lower heartbeat interval than the default, you must configure your JVM to have a low pause time during garbage collection by setting the `-XX:+UseConcMarkSweepGC` option.

For more information, see "Configuring the JVM, Java Options, and Database Cache" in *Oracle Fusion Middleware Installation Guide for Oracle Unified Directory*.

#### 26.3.5.1 To Change the Heartbeat Interval

1. Display the current value of the heartbeat interval.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -n \
  get-replication-domain-prop \
  --provider-name "Multimaster Synchronization" \
  --domain-name "dc=example,dc=com (domain 15853)" --advanced \
  --property heartbeat-interval
Property           : Value(s)
-------------------:---------
heartbeat-interval : 10 s
```

2. Change the heartbeat interval.

   The following command changes the heartbeat interval to 5 seconds.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -n \
  set-replication-domain-prop \
  --provider-name "Multimaster Synchronization" \
  --domain-name "dc=example,dc=com (domain 15853)" --set heartbeat-interval:5s
```

### 26.3.6 Changing the Isolation Policy

The isolation policy is a property of the replication domain and specifies the behavior of the directory server if replication is configured but none of the replication servers are up and running when an update is received. The default behavior of the directory server in this situation is to reject all updates.

#### 26.3.6.1 To Change the Isolation Policy

1. Display the current isolation policy.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file \
 get-replication-domain-prop \
  --provider-name "Multimaster Synchronization" \
  --domain-name "dc=example,dc=com (domain 15853)" \
  --advanced --property isolation-policy -n

Property         : Value(s)
-----------------:------------------
isolation-policy : reject-all-updates
```

2. Change the isolation policy.

   The following command specifies that the directory server should accept all updates in this situation.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file \
  set-replication-domain-prop \
  --provider-name "Multimaster Synchronization" \
  --domain-name "dc=example,dc=com (domain 15853)" \
  --set isolation-policy:accept-all-updates -n
```

### 26.3.7 Configuring Encrypted Replication

By default, replication traffic is not encrypted. To enable encryption, use the dsconfig command to set the properties of the crypto manager.

The following command specifies that replication traffic should be encrypted.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  set-crypto-manager-prop --set ssl-encryption:true
```

## 26.3.8 Configuring Replication Groups

*Replication groups* are designed to support multi-data center deployments and disaster recovery scenarios. For information about the design and implementation of replication groups in the directory server, see Section 6.6, "Replication Groups".

> **Note:** Changing the replication group configuration has an impact on assured replication. For more information, see Section 6.7, "Assured Replication".

### 26.3.8.1 To Configure a Replication Group

A replication group is configured on each directory server and replication server that should be part of the same group. On directory servers, a replication group is configured *per replicated domain*. On replication servers, the group is configured for the entire replication server.

Replication groups are configured by giving each replicated domain and replication server the same group ID. This example configures a replication group (1) for the replicated domain `dc=example,dc=com`.

1. On each directory server that will be part of this group, set the group ID for the domain `dc=example,dc=com`.

   ```
   $ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -n \
     set-replication-domain-prop \
     --provider-name "Multimaster Synchronization" \
     --domain-name "dc=example,dc=com (domain 10233)" --advanced \
     --set group-id:1
   ```

2. On each replication server that will be part of this group, set the group ID.

   ```
   $ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -n \
     set-replication-server-prop \
     --provider-name "Multimaster Synchronization" --advanced \
     --set group-id:1
   ```

## 26.3.9 Configuring Assured Replication

In most deployment scenarios, the loosely consistent multi-master replication model is sufficient. However, certain scenarios might require tighter consistency between replicas. In such cases, you can configure *assured replication*, which provides the following benefits:

- **High availability of data.** If a server crashes immediately after a modification is received on that server, there is a risk that the modification will be lost before it is replayed to other servers in the topology. With assured replication, any modification is replayed to another server in the topology *before* an acknowledgement is sent to the client application. The risk of losing data in the event of a server crash is therefore minimized.
- **Immediacy of data availability.** Some applications might require modifications to be available on additional servers in the topology immediately after a modification is made.

Assured replication is an extension of the replication protocol and is configured *per replicated domain*. For more information, see Section 26.3.1, "Retrieving the Replication Domain Name".

Assured replication is not the same as *synchronous replication*. That is, changes do not occur simultaneously on all servers in the topology. However, assured replication can mimic the functionality of synchronous replication to an extent, as far as LDAP clients are concerned. This is achieved by delaying acknowledgements to the client application until a modification has been propagated to additional servers in the topology.

---

**Note:** Assured replication relies on *replication groups*. All replication servers and directory servers that function together in an assured replication configuration must be part of the same replication group.

---

Assured replication can function in two modes:

- **Safe data mode.** Any update must be propagated to a defined number of replication servers before the client receives an acknowledgement that the update has been successful.

  The number of replication servers that must be reached defines the *safe data level*. The higher the safe data level, the higher the overall data availability.

- **Safe read mode.** Any update must be propagated to all the directory servers in the topology before the client receives an acknowledgement that the update has been successful.

In both safe data mode and safe read mode, you can configure a time-out interval to prevent LDAP client calls from hanging if certain servers in the topology are not available.

- On each *directory server*, you can configure a global time-out that comes into effect when the directory server sends an update to its replication server, either safe data mode or safe read mode. If this time-out is reached, the LDAP client call returns immediately and a message is written to the replication log to track the event.
- On each *replication server*, you can configure a global time-out that comes into effect when the replication server sends an update to a peer replication server or to another directory server, either in safe data mode or in safe read mode. If this time-out is reached, the acknowledgement message that is returned to the initiating server (either a directory server or a replication server) includes a message that indicates the time-out. The initial directory server then logs a message that the time-out occurred for that update.

**Note:** The default time-out of two seconds for a directory server and one second for a replication server should be satisfactory for most deployments. *Only* change the time-out if you are viewing time-outs in the logs and if you have a complete understanding of the impact of such a change. The value of the time-out should reflect the anticipated time that an update requires to go through its full path to reach its destination.

The time-out value on a directory server should always be higher than the value on the replication server. For example: `DS1(timeout 2s) -> RS1(timeout 1s) -> RS2(timeout 1s) -> DS2.`

For a detailed explanation of the assured replication mechanism and the various configurable options, see Section 6.7, "Assured Replication".

#### 26.3.9.1 To Configure Assured Replication in Safe Data Mode

This procedure configures assured replication in safe data mode for a topology. The procedure assumes that replication has already been configured.

1. On each directory server in the topology:

   a. Set the assured replication mode.

   ```
   $ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -n \
     set-replication-domain-prop \
     --provider-name "Multimaster Synchronization" \
     --domain-name "dc=example,dc=com (domain 10233)" --advanced \
     --set assured-type:safe-data
   ```

   b. Set the safe data level.

   ```
   $ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -n \
     set-replication-domain-prop \
     --provider-name "Multimaster Synchronization" \
     --domain-name "dc=example,dc=com (domain 10233)" --advanced \
     --set assured-sd-level:2
   ```

   If you have configured replication by using `setup` or `dsreplication`, your replication servers and directory servers will be on the same virtual machine. In this case, you must set the safe data level to 2 or higher.

   c. Set the assured replication time-out.

   ```
   $ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -n \
     set-replication-domain-prop \
     --provider-name "Multimaster Synchronization" \
     --domain-name "dc=example,dc=com (domain 10233)" --advanced\
     --set assured-timeout:5s
   ```

   *Only* change the time-out if you are viewing time-outs in the logs and if you have a complete understanding of the impact of such a change.

   d. Verify the directory server group ID.

   This should be the same for all replication servers and directory servers that form part of this replication group. For instructions on configuring the group ID, see Section 26.3.8, "Configuring Replication Groups".

   e. Display the current assured replication configuration.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -n \
  get-replication-domain-prop \
  --provider-name "Multimaster Synchronization" \
  --domain-name "dc=example,dc=com (domain 10233)" --advanced \
  --property assured-type --property assured-sd-level --property
assured-timeout

Property         : Value(s)
-----------------:------------
assured-sd-level : 2
assured-timeout  : 5 s
assured-type     : safe-data
```

2. On each replication server in the topology:

   a. Display the current assured replication configuration.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -n \
  get-replication-server-prop \
  --provider-name "Multimaster Synchronization" --advanced \
  --property assured-timeout --property group-id

Property                  : Value(s)
--------------------------:---------
assured-timeout           : 1 s
group-id                  : 1
```

   b. Set the assured replication time-out.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -n \
  set-replication-server-prop \
  --provider-name "Multimaster Synchronization" --advanced \
  --set assured-timeout:5s
```

   *Only* change the time-out if you are viewing time-outs in the logs and if you have a complete understanding of the impact of such a change.

   c. Verify the replication server group ID.

   This should be the same for all replication servers and directory servers that form part of this replication group. For instructions on configuring the group ID, see Section 26.3.8, "Configuring Replication Groups".

#### 26.3.9.2 To Configure Assured Replication in Safe Read Mode

Assured replication is configured *per replicated domain*. This procedure configures assured replication in safe read mode for a topology. The procedure assumes that replication has already been configured.

1. On each directory server in the topology:

   a. Set the assured replication mode.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -n \
  set-replication-domain-prop \
  --provider-name "Multimaster Synchronization" \
  --domain-name "dc=example,dc=com (domain 10233)" --advanced \
  --set assured-type:safe-read
```

   b. Set the assured replication time-out.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -n \
  set-replication-domain-prop \
```

```
  --provider-name "Multimaster Synchronization" \
  --domain-name "dc=example,dc=com (domain 10233)" --advanced \
  --set assured-timeout:5s
```

*Only* change the time-out if you are viewing time-outs in the logs and if you have a complete understanding of the impact of such a change.

**c.** Verify the directory server group ID.

This should be the same for all replication servers and directory servers that form part of this replication group. For instructions on configuring the group ID, see Section 26.3.8, "Configuring Replication Groups". For more information about groups and assured replication, see Section 6.7, "Assured Replication".

**d.** Display the current assured replication configuration.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -n \
  get-replication-domain-prop \
  --provider-name "Multimaster Synchronization" \
  --domain-name "dc=example,dc=com (domain 10233)" --advanced \
  --property assured-type --property assured-timeout --property group-id

Property         : Value(s)
-----------------:------------
assured-timeout  : 5 s
assured-type     : safe-read
group-id         : 1
```

**2.** On each replication server in the topology:

**a.** Display the current assured replication configuration.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -n \
  get-replication-server-prop \
  --provider-name "Multimaster Synchronization" --advanced \
  --property assured-timeout --property degraded-status-threshold \
  --property group-id

Property                  : Value(s)
--------------------------:---------
assured-timeout           : 1 s
degraded-status-threshold : 5000
group-id                  : 1
```

**b.** Set the assured replication time-out.

*Only* change the time-out if you are viewing time-outs in the logs and if you have a complete understanding of the impact of such a change.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -n \
  set-replication-server-prop \
  --provider-name "Multimaster Synchronization" --advanced \
  --set assured-timeout:5s
```

**c.** Set the degraded status threshold.

The degraded status threshold defines the stage at which the server is regarded as "too slow", based on the number of updates queued in the replication server for that directory server. For more information, see Section 6.5.2, "Degraded Status".

Do *not* adjust this value unless you observe time-outs in the logs.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -n \
```

```
set-replication-server-prop \
--provider-name "Multimaster Synchronization" --advanced \
--set degraded-status-threshold:2000
```

**d.** Verify the replication server group ID.

This should be the same for all replication servers and directory servers that form part of this replication group. For instructions on configuring the group ID, see Section 26.3.8, "Configuring Replication Groups". For more information about groups and assured replication, see Section 6.7, "Assured Replication".

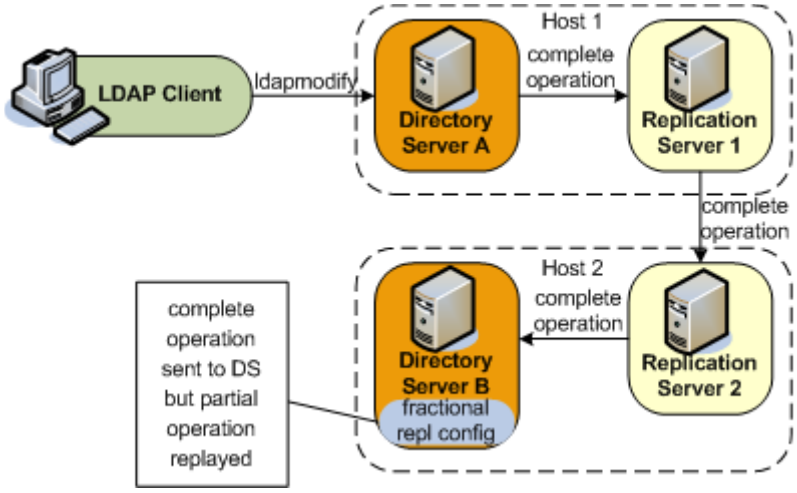
## 26.3.10 Configuring Fractional Replication

Fractional replication enables you to replicate specific parts of directory data to other replicas in the topology. This feature is particularly useful in the following scenarios:

- **Limited disk space.** Restricting the data that is replicated can significantly cut down on the amount of disk space that is required on certain replicas, particularly if you restrict the replication of attributes such as jpeg photos, which represent large data volumes.
- **Security concerns.** Certain data, such as user passwords, might be sensitive and not required on certain replicas, especially if there is a risk of inappropriate access on these replicas.

This section describes how to configure fractional replication on one or more servers in a topology. For information about the architecture of the fractional replication mechanism, see Section 6.8, "Fractional Replication".

Fractional replication is configured on the directory server that receives the partial data, and is attribute-based. Consider the following illustration:




Fractional replication is configured on Directory Server B. An `ldapmodify` operation is sent to Directory Server A. The entire operation is forwarded to Replication Server 1, then to Replication Server 2, then to Directory Server B. When the operation is replayed on Directory Server B, certain attributes from the operation are filtered out, based on that server's fractional configuration.

Fractional replicas remain writable directly from client applications. However, if an add or modify operation that includes certain "forbidden attributes" is attempted on a fractional replica, the operation is denied and the server returns an "Unwilling to perform" error.

Fractional replication can be configured in one of two modes:

- **Exclusive mode.** In this mode, the multi-valued `fractional-exclude` attribute is used to filter out the specified attributes from an incoming LDAP add or modify operation.

  Excluded attributes must be *optional* attributes of an object class.

- **Inclusive mode.** In this mode, the multi-valued `fractional-include` attribute is used to filter in only the specified attributes from an incoming LDAP add or modify operation.

  All other attributes (except for those that are mandatory in the object class) are removed from the change that is replayed on the server.

The two modes are mutually exclusive, that is, you can include only one of these attributes in a domain configuration.

Fractional replication is configured *per replicated domain* (see Section 26.3.1, "Retrieving the Replication Domain Name"). A *fractional domain* implies that certain attributes are entirely absent from the domain. These attributes are filtered out at operation replay time but are also absent from the existing data in the domain.

To ensure coherency of the data across a replicated topology, it is necessary to identify whether a particular data set is fractional. The configuration of a new fractional domain therefore implies specific steps to ensure that the domain is free of forbidden attributes, and recognizable as a fractional domain. For more information, see Section 26.3.10.3, "To Configure and Initialize a Fractional Domain".

Use the `dsconfig` command to configure fractional replication in a domain, as follows.

### 26.3.10.1 To Configure Exclusive Fractional Replication

The following example configures a replica to exclude the `photo` and `jpegPhoto` attributes from any creation or modification of an entry whose object class is `inetOrgPerson`.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X \
  set-replication-domain-prop --provider-name "Multimaster Synchronization" \
  --domain-name "dc=example,dc=com (domain 10233)" \
  --set fractional-exclude:inetOrgPerson:photo,jpegPhoto
```

Object classes and attributes can be specified by their names, or by their OIDs, so the following example has the same effect as the previous example:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X \
  set-replication-domain-prop --provider-name "Multimaster Synchronization" \
  --domain-name "dc=example,dc=com (domain 10233)" \
  --set fractional-exclude:2.16.840.1.113730.3.2.2:0.9.2342.19200300.100.1.7, \
  0.9.2342.19200300.100.1.60
```

If you use object class or attribute names *and* OIDs, both values are added. For example, the following command adds both the attribute name and its OID to the list of excluded attributes:

```
$ dsconfig set-replication-domain-prop ...
  --set fractional-exclude:*:jpegPhoto,*:0.9.2342.19200300.100.1.60
```

If you wanted to remove this attribute from the list, you would need to remove both the attribute name and the OID.

To specify that the photo and jpegPhoto attributes should be removed from any creation or modification of any entry (regardless of its object class), use an asterisk in place of the object class. For example:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X \
  set-replication-domain-prop --provider-name "Multimaster Synchronization" \
  --domain-name "dc=example,dc=com (domain 10233)" \
  --set fractional-exclude:*:photo,jpegPhoto
```

### 26.3.10.2 To Configure Inclusive Fractional Replication

The following example configures a replica to include only the uid and employeeNumber attributes from any creation or modification of an entry whose object class is inetOrgPerson. All other attributes are ignored in the modification, except those that are mandatory for the object class.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X \
  set-replication-domain-prop --provider-name "Multimaster Synchronization" \
  --domain-name "dc=example,dc=com (domain 10233)" \
  --set fractional-include:inetOrgPerson:uid,employeeNumber
```

Object classes and attributes can be specified by their names, or by their OIDs, so the following example has the same effect as the previous example:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X \
  set-replication-domain-prop --provider-name "Multimaster Synchronization" \
  --domain-name "dc=example,dc=com (domain 10233)" \
  --set fractional-include:2.16.840.1.113730.3.2.2:0.9.2342.19200300.100.1.1, \
  2.16.840.1.113730.3.1.3
```

If you use object class or attribute names *and* OIDs, both values are added. For example, the following command adds both the attribute name and its OID to the list of included attributes:

```
$ dsconfig set-replication-domain-prop ...
  --set fractional-include:*:jpegPhoto,*:0.9.2342.19200300.100.1.60
```

If you wanted to remove this attribute from the list, you would need to remove both the attribute name and the OID.

To specify that a particular attribute should be included in the creation or modification of any entry (regardless of its object class), use an asterisk in place of the object class. The following example includes only the description attribute in a creation or modification operation on any entry.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X \
  set-replication-domain-prop --provider-name "Multimaster Synchronization" \
  --domain-name "dc=example,dc=com (domain 10233)" \
  --set fractional-include:*:description
```

### 26.3.10.3 To Configure and Initialize a Fractional Domain

The following steps are required when you initialize a new fractional domain:

1. Configure exclusive or inclusive fractional replication, as described in the previous two sections.

   At this point, the domain obtains a *bad generation ID* status. For more information, see Section 6.5, "Replication Status".

   This means that all modifications on the domain are blocked until the data is synchronized with the rest of the topology.

2. Import a new data set from one of the other servers in the topology.

   The new data set can be imported online, by using `dsreplication initialize` or by using `import-ldif` in online or offline mode. The server from which you import the data must either be an entire replica (that is, not a fractional replica) or must have the same fractional configuration as the server to which you are importing the data. During the import, all entries will be filtered with the fractional configuration set up in the previous step.

   For information about how to import a data set, see Section 26.4.1, "Initializing a Single Replicated Server" and Section 17.1, "Importing and Exporting Data".

3. After the data import, the domain returns to *normal* status.

   For more information, see Section 6.5, "Replication Status".

   The domain is now able to accept new entries from local LDAP operations, or synchronization operations with other servers in the topology. The data in the domain is free of any "forbidden" attributes.

## 26.3.11 Configuring Replication Status

Each replicated domain in a replicated topology has a certain *replication status*, depending on its connections within the topology, and on how up to date it is with regard to the changes that have occurred throughout the topology. For more information, see Section 6.5, "Replication Status".

Replication status is generated automatically, based on how up to date a server is within the replicated topology. The only parameter that can be configured is the degraded status threshold. This parameter defines the maximum number of changes that can be in the replication server's queue for all domains of the directory servers that are connected to this replication server. When this number is reached, for a specific directory server, that server is assigned a degraded status. The degraded status remains until the number of changes drops beyond this value.

**Note:** The default value of the degraded status threshold should be adequate for most deployments. Only modify this value if you observe several time-out messages in the logs when assured replication is configured.

### 26.3.11.1 To Configure the Degraded Status Threshold

The default number of changes defined by this threshold is 5000. This example sets the threshold to 6000, to take into account a network with more latency.

On the replication server, use `dsconfig` to set the degraded status threshold.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  set-replication-server-prop --provider-name "Multimaster Synchronization" \
  --set degraded-status-threshold:6000
```

## 26.3.12 Configuring the Replication Server Weight

In large topologies with several directory servers and several replication servers, it is more efficient to spread the directory servers out across the replication servers in a predefined manner. You can specify how many directory servers should connect to each replication server in a topology according to the relative capacity of the machine on which the replication server is running. For more information, see Section 6.2.3.2, "Replication Server Load Balancing".

To configure the replication server weight, run the `dsconfig` command as follows:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  set-replication-server-prop \
  --provider-name "Multimaster Synchronization" --set weight:2
```

By default, the weight of each replication server in the topology is 1.

## 26.4 Initializing a Replicated Server With Data

This section describes how to initialize a replicated server with data by using the dsreplication command. `dsreplication` accesses the server configuration over SSL via the administration connector. For more information, see Section 14.3, "Managing Administration Traffic to the Server".

This section references some of the information covered in Section 17.1.1, "Populating a Stand-Alone Directory Server With Data". It is recommended that you read that section before this one.

### 26.4.1 Initializing a Single Replicated Server

The easiest way to initialize a single directory server in a replicated topology is to use the `dsreplication` command to copy the data over from another directory server in the topology. This command requires replication to have been enabled between the source server and the destination server. The command replaces all data under the specified base DN on the destination server with the data from the source server.

For example, the following command initializes the base DN `"dc=example,dc=com"` on `host2` with the data on `host1`.

```
$ dsreplication initialize --baseDN "dc=example,dc=com" \
  --adminUID admin --adminPasswordFile pwd.txt \
  --hostSource host1 --portSource 4444 \
  --hostDestination host2 --portDestination 4444 --trustAll
```

### 26.4.2 Initializing a New Replicated Topology

To initialize all directory servers in a new replicated topology, use one of the following options:

- Initialize all directory servers individually with the same data, using one of the methods described in Section 17.1.1, "Populating a Stand-Alone Directory Server With Data". When you have initialized all directory servers with data, enable replication between the servers.
- Initialize a single directory server using one of the methods described in Section 17.1.1, "Populating a Stand-Alone Directory Server With Data". Enable replication for all directory servers, then use the `dsreplication intialize-all` command to initialize all the remaining servers simultaneously. This command takes the details of the source server as arguments, and initializes all other servers for which replication is enabled.

  For example, the following command initializes all directory servers from the contents on `host1`.

  ```
  $ dsreplication initialize-all --hostname localhost --port 4444 --trustAll \
    --baseDN "dc=example,dc=com" --adminUID admin --adminPasswordFile pwd.txt
  ```

### 26.4.3 Adding a Directory Server to an Existing Replicated Topology

When you add a directory server to an existing replicated topology, the new server must be populated with the same *generation* of data as the existing directory servers in the topology. The data generation is an ID stored within the root entry of the replication domain. When the data generation does not exist, it is computed by the replication mechanism and stored. To ensure that the new directory server has the same data generation as the other servers in the topology, use one of the following methods to populate the directory server with data:

- Use the same original LDIF file, backup file, or binary copy that was used to populate the other directory servers.
- Use the result of an export, backup, or binary copy from another directory server in the topology.

If you install the new directory server using the GUI install and specify that it will be part of the replicated topology, the server is initialized with the correct data generation automatically.

If you do not install the directory server using the GUI install, and you use the `dsreplication` command to enable replication, you must initialize the server manually using one of the methods described in the previous section.

If a directory server in the topology does not contain the same data generation as the rest of the topology, data cannot be replicated to or from the server. However, the directory server remains connected to the topology, enabling it to be initialized using the replication protocol. Replication on this directory server is said to be *downgraded*.

When a directory server with the correct data generation is added to an existing topology, the replication mechanism automatically replays any changes that occurred since the first directory server in the topology was initialized with data. This action ensures that the new directory server is synchronized with the rest of the topology.

### 26.4.4 Changing the Data Set in an Existing Replicated Topology

Changing the data set implies importing an entirely new set of data to every directory server in the topology. When the data set is changed, two tasks are performed:

- The new data is applied to each directory server in the topology.
- The replication servers are cleared of any changes they might contain. This task includes resetting the data generation on the directory servers so that the new data generation is used.

If you change the data set using the `dsreplication initialize` command, both of these tasks are performed automatically. However, if you use the `import-ldif` command or the binary copy method to change the data set, you must perform these tasks manually, as described in the following section.

#### 26.4.4.1 To Change the Data Set With `import-ldif` or Binary Copy

1. Clear the generation ID from the directory servers by running the `dsreplication pre-external-initialization` command.

   It is sufficient to run this command on only one directory server in the topology. All directory servers in the topology will be updated, unless you specify that only one server should be updated. For example, the following command prepares all servers in the topology for initialization by using `import-ldif` or binary copy:

   ```
   $ dsreplication pre-external-initialization -h host1 -p 4444 -X \
     -b dc=example,dc=com -I admin -j pwd-file
   ```

```
Are you going to initialize only the contents of server host1:4444 (type
'no' if you will initialize contents of all replicated servers for the given
Base DNs)? (yes / no) [no]:
Preparing base DN dc=example,dc=com to be initialized externally ..... Done.
Now you can proceed to the initialization of the contents of the base DNs on
all the replicated servers. You can use the command import-ldif or the binary
copy to do so. When the initialization is completed you must use the subcommand
{post-external-initialization} for replication to work with the new base DNs
contents.
```

2. Use `import-ldif` or binary copy to initialize all directory servers in the topology with data.
3. Reset the generation ID by running the `dsreplication post-external-initialization` command.

   It is sufficient to run this command on only one directory server in the topology. All other directory servers are updated. For example, the following command resets the generation ID for all directory servers in the topology after initialization using `import-ldif` or binary copy:

```
$ dsreplication post-external-initialization -h localhost \
  -p 4444 -b dc=example,dc=com -I admin -j pwd-file -X
Updating replication information on base DN dc=example,dc=com ..... Done.
Post initialization procedure completed successfully.
```

### 26.4.5 Appending Data in an Existing Replicated Topology

The easiest way to import a large number of entries to an existing replicated topology that already contains a large number of entries is to use the `import-ldif` command with the `-a` or `--append` option.

When you import data by using the `import-ldif` command, the imported data is not replicated automatically. You must therefore run `import-ldif --append` on every directory server in the topology. This strategy enables you to import the data with no downtime in the directory service.

You can also use the `dsreplication initialize-all` command after you have imported the data to a single directory server in the topology. However, this strategy will result in the directory service being unavailable for a certain period of time.

## 26.5 Using the External Change Log

The External Change Log (ECL) publicizes all changes that have occurred in a directory server database and is particularly useful for synchronizing the LDAP directory with other subsystems.

The ECL is built online from the replication change log and does not use an additional database for its storage. It is not a regular JEB backend, therefore no index needs to be configured.

This section describes how to enable the ECL in your directory service and how to configure client applications so that they can access the ECL. The section covers the following topics:

- Section 26.5.1, "Enabling the External Change Log"
- Section 26.5.2, "External Change Log APIs"

### 26.5.1 Enabling the External Change Log

The ECL is available by default on any server instance that includes *both* a directory server *and* a replication server. The ECL is not available by default on a server instance that is configured as either a *dedicated directory server* or a *dedicated replication server* (as described in Section 26.2, "Configuring Large Replication Topologies").

The ECL is enabled when replication is configured in one of the following ways:

- By configuring a directory server as part of a replicated topology during installation. For more information, see "Setting Up Replication During Installation" in *Oracle Fusion Middleware Installation Guide for Oracle Unified Directory*.
- By configuring replication after installation, by using the `dsreplication` command. For more information, see Section 26.1, "Configuring Data Replication With `dsreplication`".

> **Note:** The ECL is *not* available if you configured replication with the `--onlyReplicationServer` or `--noReplicationServer` options.

Although the ECL functionality is based on the replication mechanism, some client applications might require access to the ECL content on a local server, outside of a replicated topology. You can enable the ECL on a local server, for a specific base DN, by running the following command:

```
$ dsreplication enable-changelog -h localhost -p 4444 -D "cn=directory manager" \
  -j pwd-file -r 8989 -b dc=example,dc=com -X -n
```

The replication port (`-r`) is required to configure the ECL, even on a standalone server, because the ECL relies on the replication mechanism. You need only specify the replication port if the change log (or replication) was not previously configured on the server. The default value of the replication port is 8989.

To verify that the ECL is configured on a directory server instance, run the following search command:

```
$ ldapsearch -h localhost -p 1389 -D "cn=directory manager" -j pwd-file \
  -s base -b "" "objectclass=*" namingContexts
```

```
dn:
namingContexts: cn=changelog
namingContexts: dc=Europe,dc=com
namingContexts: dc=us,dc=com
```

## 26.5.2 External Change Log APIs

The ECL supports two APIs, which enable two distinct *modes* of operation:

- **Cookie mode.** This is the recommended API that you should use to access the ECL.

  In cookie mode, the client application provides an ECL exchange control in its request to the server. In this mode, the DIT and schema provided in the entries that are returned by the server are not compatible with the LDAP change log draft (`http://tools.ietf.org/html/draft-good-ldap-changelog-04`).

- **Draft-compatible mode.** This mode should be used only by existing applications that rely on the LDAP change log draft.

  In this mode, the DIT and schema provided in the entries that are returned by the server are compatible with the LDAP change log draft.

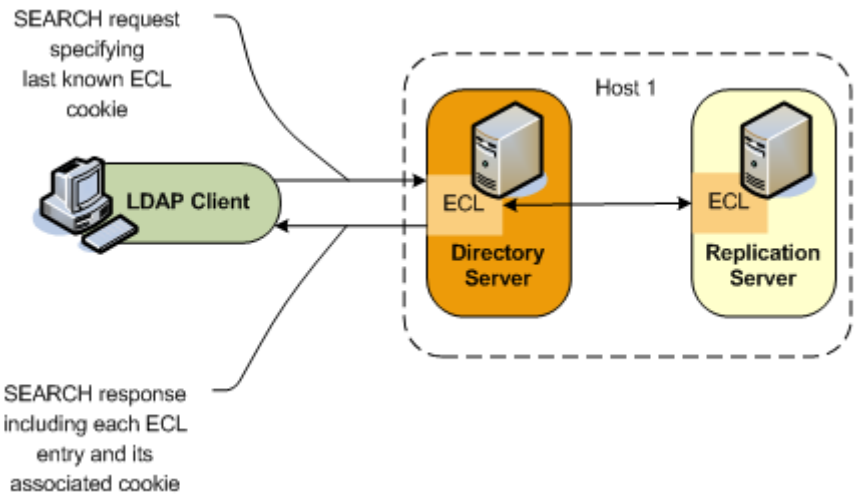
  For improved performance and for simplicity, you should port client applications to use the cookie mode. For more information, see Section 26.5.12, "Porting Applications That Rely on Other Change Logs".

## 26.5.3 How a Client Application Uses the External Change Log in Cookie Mode

Each entry in the ECL has an associated cookie. When a client application sends a SEARCH request, the application provides either the cookie of the last message that was read from the ECL (in a previous SEARCH), or an empty value. The server returns the ECL entries associated with that cookie.

Each entry is returned with its associated cookie. When the application disconnects, it stores the last cookie that it received, and provides this cookie to the server with its next SEARCH request.

This transmission of ECL cookies is illustrated in the following diagram.




The content of the cookie is *not* a public interface for the client application. The client application sends the cookie as a request control and the server sends the cookie as a response control.

The cookie exchange control has an OID of 1.3.6.1.4.1.26027.1.5.4. If the server identifies that the cookie provided by the application is corrupted, the request is rejected. The request is also rejected if the server identifies that the configuration of the ECL has changed since the server sent this cookie to the application, or that the ECL has been purged and the oldest change stored is newer than the cookie value. In this case, additional information is returned, indicating that a full re synchronization of the external application is recommended.

> **Note:** If a server is disconnected from the replication topology and processes changes from clients that are connected to it, convergence cannot be guaranteed.

The following request and response examples indicate how the client application searches using the external change log and how the ECL responds.

#### Request One

To start reading the ECL, the client sends the first SEARCH request on cn=changelog, specifying an empty value in the cookie exchange control.

```
$ ldapsearch -h localhost -p 1389 -D "cn=directory manager" -j pwd-file \
  --control "1.3.6.1.4.1.26027.1.5.4:false:;" -b "cn=changelog" \
  "(objectclass=*)" "*" +
```

#### Response One

The server sends each change to the client in a SearchResultEntry. The cookie attribute specifies the new cookie value. This value is also sent in a cookie exchange control, along with the entry.

```
# Public changelog exchange control(1.3.6.1.4.1.26027.1.5.4):
  dc=europe,dc=com:0000012187eae081456200000001;o=example:;
dn: replicationcsn=0000012187eae081456200000001,dc=europe,dc=com,cn=changelog
objectClass: top
objectClass: changeLogEntry
replicationCSN: 0000012187eae081456200000001
replicaIdentifier: 17762
targetDN: cn=chek-piao chea,ou=unit1,o=people,dc=europe,dc=com
changeTime: 20090528155105Z
changes:: cmVwbGFjZTogc2VlQWxzbwpzZWVBbHNvOiBjbj1tY29uZmlnCi0KcmVwbGFjZTogbW9kaW
  ZpZXJzTmFtZQptb2RpZmllcnNOYW1lOiBjbj1EaXJlY3RvcnkgTWFuYWdlcixjbj1Sb290IEROcyxjb
  j1jb25maWcKLQpyZXBsYWNlOiBtb2RpZnlUaW1lc3RhbXAKbW9kaWZ5VGltZXN0YW1wOiAyMDA5MDUy
  ODE1NTEwNVoKLQo=
changeType: modify
changeLogCookie: dc=europe,dc=com:0000012187eae081456200000001;
targetEntryUUID: 08d1830c-02f1-34a6-9cf4-8d1270ec1db0
changeNumber: 0
```

#### Request Two

To read the ECL from the last returned entry, the client sends the SEARCH request on cn=changelog, specifying the last cookie value that it received in the cookie exchange control.

```
$ ldapsearch -h localhost -p 1389 -D "cn=directory manager" -j pwd-file
  --control
"1.3.6.1.4.1.26027.1.5.4:false:dc=europe,dc=com:0000012187eae081456200000001;"
  -b "cn=changelog" "(objectclass=*)"
```

**Note:** The contents of the external change log are base 64 encoded. For information about decoding the content, see Section A.3.2, "base64".

## 26.5.4 Format of External Change Log Entries

The DN for entries that are returned in the ECL is of the form:

```
replicationcsn=replicationCSN,replication-domain-DN,cn=changelog
```

For example:

```
dn: replicationcsn=0000012187eae081456200000001,dc=europe,dc=com,cn=changelog
```

The following attributes are returned for ECL entries:

```
targetDN / MUST
changeType / MUST
changeTime / MUST
changeNumber / MUST // used only for compatibility mode

changes / MAY, MUST for add, mod
newRDN / MAY, MUST for modrdn
deleteOldRDN / MAY, MUST for modrdn
newSuperior / MAY, MUST for modrdn

replicaIdentifier / MAY, OPERATIONAL / specific OUD value
replicationCSN / MAY, OPERATIONAL / specific OUD value
targetEntryuuid / MAY, OPERATIONAL / specific OUD value
changelogcookie / MAY, OPERATIONAL
```

## 26.5.5 Specifying the Attributes to be Included in the External Change Log

By default, attributes are included in the ECL only if they are affected by a change operation. So, for example, if the `sn` attribute of an entry is modified, only that attribute will appear in the ECL. You can, however, specify a list of attributes that will be included in the ECL regardless of whether they are affected by a change operation. In addition, you can also determine if this list of attributes is included for all types of operations or for delete operations only.

You can configure the attributes using the following properties:

- `ecl-include`
- `ecl-include-del-only`

### Using the `ecl-include` Property

You can use the `ecl-include` property to configure attributes to be included in the ECL if an entry is modified.

Use the `dsconfig` command to set the value of the `ecl-include` property. For example, to specify that the `cn`, and `sn` attributes always be included in the ECL if an entry is modified, run the following command:

```
dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -Q -n -X \
set-external-changelog-domain-prop --provider-name "Multimaster Synchronization" \
--domain-name dc=example,dc=com \
--add ecl-include:cn --add ecl-include:sn
```

In the ECL entry that is returned by the server, the attribute name is prefixed with `target`. For example, in the previous example, the ECL entries for changes on `dc=example,dc=com` will always contain the attributes `targetcn` and `targetsn`. The values of these attributes will be the values of the `cn` and `sn` attributes of the entry before it was modified or moved.

#### Using the `ecl-include-del-only` Property

In combination with the `ecl-include` property, you can use the `ecl-include-del-only` property to retrieve extra attributes for delete operations only.

Use the `dsconfig` command to set the value of the `ecl-include-del-only` property.

```
dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -Q -n -X \
set-external-changelog-domain-prop --provider-name "Multimaster Synchronization" \
--domain-name dc=example,dc=com \
--add ecl-include:cn --add ecl-include:sn --set ecl-include-del-only:true
```

### 26.5.6 Specifying the Attributes to be Excluded in the External Change Log

Client applications that use ECL are not always interested in all the LDAP operations executed on the server. Therefore, to avoid processing of irrelevant information you can filter a list of attributes.

You can use the `ecl-blacklist` property to configure attributes to be excluded from the ECL. It only skips MODIFY operations sent to the client application when all the modifications apply to blacklisted attributes.

> **Note:** The blacklist mechanism requires the use of the cookie mode.

Use the `dsconfig` command to set the value of the `ecl-blacklist` property. For example, to specify that the modify operations concerning attributes `email` and `telephonenumber` should be excluded from ECL, run the following command:

```
dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -Q -n -X \
set-external-changelog-domain-prop --provider-name "Multimaster Synchronization" \
--domain-name dc=example,dc=com --add ecl-blacklist:email \
--add ecl-blacklist:telephonenumber
```

### 26.5.7 Initializing Client Applications to Use the External Change Log

No specific server configuration is required for clients to use the ECL. However, any client application that needs to use the ECL must be initialized, as described in the following sections.

- Section 26.5.7.1, "To Initialize a Client Application to Use the External Change Log"
- Section 26.5.7.2, "Reinitializing a Client Application When a Domain is Added"
- Section 26.5.7.3, "Reinitializing a Client Application When a Domain is Removed or Disabled"

#### 26.5.7.1 To Initialize a Client Application to Use the External Change Log

The following example describes a scenario in which host 2 is initialised from host 1. Host 1 is not frozen during the initialization operation, so continues to receive changes. This procedure guarantees that host 2 does not lose any of the changes that were received on host 1.

1. Save the current state of host 1 by reading the last ECL cookie value on host 1.

   This is the value of the `lastExternalChangelogCookie` attribute of the root DSE. For example:

   ```
   $ ldapsearch -h localhost -p 1389 -D "cn=directory manager" -j pwd-file \
     -s base -b "" "objectclass=*" lastExternalChangelogCookie
     dn:
     objectClass: top
     objectClass: ds-root-dse
     lastExternalChangelogCookie: dc=europe:00000121cea5221c04b100000005 \
       00000121cea5319e04b400000009;
   ```

   Note that host 1 is not frozen and continues to receive changes.

2. To initialize host 2, export the Oracle Unified Directory database from host 1 and import it to host2.

3. Initialize the application from the exported database.

   Restart replication on host 2, using the current state saved in Step1. The application can now start reading the ECL by providing the last cookie value as the value of the search control. For example:

   ```
   $ ldapsearch -h localhost -p 1389 -D "cn=directory manager" -j pwd-file
     --control
   "1.3.6.1.4.1.26027.1.5.4:false:dc=europe:00000121cea5221c04b100000005 \
       00000121cea5319e04b400000009" -b "cn=changelog" "(objectclass=*)"
   ```

#### 26.5.7.2 Reinitializing a Client Application When a Domain is Added

When a new replication domain is added to a topology, the ECL is enabled on that domain by default. Client applications that use the ECL must be reinitialized for the new domain.

The server enforces this requirement by rejecting SEARCH operations if the cookie that is provided does not refer to the new domain. The operation result code is `UNWILLING TO PERFORM`. The server provides a detailed message that includes a list of the domains that are missing and a cookie value for a possible partial initialization.

The client application must be reinitialized using one of the following methods:

- **Full reinitialization.** The application is reinitialized for all domains.

  1. Read the value of the `lastExternalChangelogCookie` attribute. This value refers to all domains in the topology, including the new domain.
  2. Export the database for all domains, including the new domain.
  3. Initialize the application for all domains from the export output. For more information, see Section 26.5.7.1, "To Initialize a Client Application to Use the External Change Log".
  4. The application can now search the ECL using the `last_cookie_from_dse_root`.

- **Partial reinitialization.** The application is reinitialized only for the new domain.

1. Export the database for the new domain only.
2. Initialize the application from the export output, which contains only the entries in the new domain. For more information, see Section 26.5.7.1, "To Initialize a Client Application to Use the External Change Log".
3. The application can now search the ECL, using the cookie value for a possible partial initialization that was returned by the server in its `UNWILLING TO PERFORM` error. Note that this might result in some updates that have already been processed being replayed, because the cookie value represents the initial state of the database.

---

**Note:** In draft compatibility mode, the draft API does not allow the server to enforce the application to be properly initialized. Therefore, in draft compatibility mode, any changes on the new domain are published in the ECL as soon as the new domain is added.

To prevent the server from publishing changes for the new domain, follow the instructions in Section 26.5.11, "Disabling the External Change Log for a Specific Domain". To ensure that an application is notified of changes to a particular domain only, specify this domain either in the base DN (in cookie mode only) or as a search filter on the `targetDN` attribute.

---

### 26.5.7.3 Reinitializing a Client Application When a Domain is Removed or Disabled

When a replication domain is removed from a topology (or when the ECL is disabled for a specific domain), client applications must be alerted to the fact that no more changes will occur on that domain.

The server enforces this requirement by rejecting SEARCH operations if the cookie that is provided refers to the removed domain. The operation result code is `UNWILLING TO PERFORM`. The server provides a detailed message, that includes a list of the domains that are present in the cookie but have been removed (or for which the ECL has been disabled), and a cookie value for a possible continuation.

The client application can use one of the following methods to handle the removed domain:

- **Smooth continuation.** In this case, the application applies its own policy of what to do when a domain is removed. To assist with the formulation of this policy, the application can search the ECL by providing the cookie value for a possible continuation that is returned by the server in the error message.
- **Full reinitialization.** The application is reinitialized for all domains.
  1. Read the value of the `lastExternalChangelogCookie` attribute. This value refers to all domains in the topology, excluding the removed domain.
  2. Export the database for all domains.
  3. Initialize the application for all domains from the export output. For more information, see Section 26.5.7.1, "To Initialize a Client Application to Use the External Change Log".
  4. The application can now search the ECL using the `lastExternalChangelogCookie`.

### 26.5.8 Controlling Access to the External Change Log

Access to the ECL is ruled by global ACIs that can be configured on the server. By default, only the root user can access the ECL.

For information about configuring global ACIs, see Section 22.1, "Managing Global ACIs With `dsconfig`".

### 26.5.9 Purging the External Change Log

The ECL is purged simultaneously with the replication change log. For information about changing the interval at which the replication change log is purged, see Section 26.3.2, "Changing the Replication Purge Delay".

Sometimes, an application might submit a search request on the ECL, providing a cookie value that is older than the oldest change stored on the server (because a purge has occurred since the last request from that application). In this case, the server rejects the requests and indicates that the cookie is too old and that a full resync is required.

### 26.5.10 Disabling the External Change Log on a Server

To disable the ECL on a server, for a specific base DN, use the `dsreplication disable-changelog` command, as follows:

```
$ dsreplication disable-changelog -h localhost -p 4444 -D "cn=directory manager" \
  -j pwd-file -b dc=example,dc=com -X -n
```

### 26.5.11 Disabling the External Change Log for a Specific Domain

In certain situations, you might want to exclude changes on a specific domain from the external change log. You can disable the ECL for a specific replication domain, which prevents changes to this domain from being published in the ECL.

1. Obtain the domain name, as described in Section 26.3.1, "Retrieving the Replication Domain Name".
2. Set the external changelog domain properties for that domain.

   For example, to prevent changes to the schema from being published in the ECL, run the following `dsconfig` command:

   ```
   $ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -n \
     set-external-changelog-domain-prop \
     --provider-name "Multimaster Synchronization" --domain-name cn=schema \
     --set enabled:false
   ```

### 26.5.12 Porting Applications That Rely on Other Change Logs

The ECL is based on the LDAP change log draft (`http://tools.ietf.org/html/draft-good-ldap-changelog-04`) but does not strictly support this change log. The LDAP change log draft uses an integer as the key to browse the change log whereas the ECL uses a cookie.

On the client side, the cookie mechanism has the following advantages:

- Ability to fail-over from one ECL instance to another
- Ability to load balance request over several ECL instances

On the server side, the cookie mechanism has the following advantages:

- Easier implementation in a multi-master environment

- Cheaper in terms of resources required on the server
- Smaller performance impact for other applications that generate changes

> **Note:** The Oracle Directory Server Enterprise Edition (ODSEE) Retro Change Log (RCL) supports the LDAP change log draft, with some specific additions.

### 26.5.12.1 Differences Between the ECL and the LDAP Change Log Draft

The following sections describe the differences between the two change logs, which will assist you in porting client applications.

**26.5.12.1.1 Index Differences** The LDAP change log draft specifies the change log index as an integer (`changenumber` attribute). This works well when the change log is served by a single server (which was the case at the time that the LDAP change log draft specification was written.) When the change log service supports more than one server and when failover is supported from one server to another, the integer format is not appropriate.

Note that you should index the `replicationCSN` attribute on `cn=changelog` for compatibility with Oracle Directory Server Enterprise Edition. If you index the `replicationCSN` attribute on parameters other than `cn=changelog`, the index might have a performance impact.

**26.5.12.1.2 DIT and Schema Differences** The LDAP change log draft specifies the DN for entries in the change log as `changenumber=changenumer,cn=changelog`. The ECL uses the following DN for entries in the change log:

`replicationcsn=`*`replicationCSN`*`,`*`replication-domain-DN`*`,cn=changelog`

The ECL schema is based on the LDAP change log draft schema, however, Oracle Unified Directory manages an index in the ECL through a cookie that is opaque to the application, rather than through the `changenumber` attribute. The schema differ as follows:

| Origin | MUST | MAY |
|---|---|---|
| LDAP Change Log Draft | `changenumber` | `changes` |
| | `targetDn` | `newRDN` |
| | `changetype` | `deleteOldRDN` |
| | | `newSuperior` |

| Origin | MUST | MAY |
|---|---|---|
| ODSEE RCL | changenumber | changes |
| | targetDn | newRDN |
| | changetype | deleteOldRDN |
| | changetime | newSuperior |
| | | changeHasReplFixupOp |
| | | changeIsReplFixupOp |
| | | deletedEntryAttrs |
| | | replicaIdentifier (operational) |
| | | replicationCSN (operational) |
| | | targetUniqueId (operational) |
| Oracle Unified Directory ECL | changenumber:0 | changes |
| | targetDn | newRDN |
| | changetype | deleteOldRDN |
| | changetime | newSuperior |
| | | replicaIdentifier (operational) |
| | | replicationCSN (operational) |
| | | targetentryuuid (operational) |
| | | changelogcookie (operational) |

### 26.5.12.2 Additional Differences Between the ECL and the Oracle Directory Server Enterprise Edition Retro Change Log

#### Schema and implementation-based values

The Oracle Directory Server Enterprise Edition RCL specifies that the target entry unique ID is stored in the `targetuniqueid` attribute. The format of this attribute value is specific to Oracle Directory Server Enterprise Edition. The `replicationcsn` attribute also has a value that is specific to Oracle Directory Server Enterprise Edition.

#### First and last ECL index

The Oracle Directory Server Enterprise Edition RCL supports the following attributes in the root DSE entry:

- The `firstchangenumber` attribute, which contains the first (oldest) change log index as an integer change number.

  This value is updated when the change log is purged. Before connecting to the change log server, an application reads the first change log index and compares it with the change log index that it stored. If the first change log index is more recent than the last change log index stored by the application, the application knows that the changes from the application index to the first change log index will never be returned by the server. They can only be obtained by reading the entries (full resync).

With the Oracle Unified Directory ECL, this procedure is not required of the application. Instead the Oracle Unified Directory server does the check and rejects the request when the cookie is too old. For more information, see Section 26.5, "Using the External Change Log".

- The `lastchangenumber` attribute, which contains the latest (newest) change log index as an integer change number.

  The Oracle Unified Directory ECL supports the equivalent feature with the `lastExternalChangelogCookie` attribute. For more information, see Section 26.5, "Using the External Change Log".

**Purge delay**

In the Oracle Directory Server Enterprise Edition RCL, the external change log and the regular replication change log are different databases. In Oracle Unified Directory, the two change logs are in the same database. This design decision has several advantages. An additional consequence of this design decision is that Oracle Directory Server Enterprise Edition can have two different trim policies (purge delays), while in Oracle Unified Directory the trim policy is the same.

#### 26.5.12.3 API for Compatibility With the LDAP Change Log Draft and the Oracle Directory Server Enterprise Edition Retro Change Log

Oracle Unified Directory provides an additional API that is compatible with the LDAP draft change log and supports most of the additional features of the Oracle Directory Server Enterprise Edition Retro Change Log. The use of this API has a performance impact in terms of CPU and database (disk) space on the server side, and some computation for the application that fails over from one ECL server to another one.

The use of this compatible API (*compatible mode*) is configured when the server receives a request on the ECL with no change log cookie. The server returns entries with a `changenumber` attribute, the value of which is an incremental integer.

The client can search the ECL by providing a filter on the `changenumber`. The target entry unique ID is stored in an attribute called `targetuniqueid` with a format compatible with the Oracle Directory Server Enterprise Edition Retro Change Log. The first and last `changenumber` are present as attributes of the root DSE entry.

**26.5.12.3.1 Limitations of the Compatibility API** Because Oracle Unified Directory does not store the ECL in a dedicated database, it does not support all the features supported by a JEB back end, such as specific indexes.

In addition, in order to support the `changenumber`-based ordering that is specified by the LDAP change log draft, Oracle Unified Directory must store a mapping from the `changenumber` to the replication state. When the server processes a request, it must try to retrieve the replication state from the `changenumber` that is provided in the request filter. If this cannot be achieved, the request is rejected.

## 26.6 Configuring Schema Replication

Schema replication is enabled by default. When you configure replication as part of the server setup, the schema of the new server is automatically initialized with the schema of the existing server in the topology.

### 26.6.1 Specifying the Schema Source

When you configure replication with the `dsreplication enable` command, you can specify that the schema of the second directory server be used to initialize the

schema of the first server. If you do not specify an option, the schema of the first directory server is used by default.

In the following example, the data of `host1` is used to initialize `host2` but the schema of `host2` is used to initialize the schema on `host1`:

```
$ dsreplication enable --host1 host1 --port1 4444 \
  --bindDN1 "cn=Directory Manager" --bindPasswordFile1 pwd.txt \
  --replicationPort1 8989 --host2 host2 --port2 4444 \
  --bindDN2 "cn=Directory Manager" --bindPasswordFile2 pwd.txt \
  --replicationPort2 8989 --adminUID admin --adminPasswordFile pwd.txt \
  --baseDN "dc=example,dc=com" --useSecondServerAsSchemaSource -X
```

## 26.6.2 Disabling Schema Replication

In certain circumstances, you might not want the schema to be replicated. The schema is replicated under a separate base DN, `"cn=schema"`.

### 26.6.2.1 To Specify That Schema Should Not Be Replicated

When you configure replication with the `dsreplication enable` command, you can specify that the schema should not be replicated, using the `--noSchemaReplication` option.

> **Note:** If you use QuickSetup to enable replication, you cannot specify that the schema should not be replicated.

### 26.6.2.2 To Disable Schema Replication

In an existing topology in which the schema are being replicated, you can disable this functionality by disabling replication of the schema base DN. The following example disables schema replication from the directory server running on the local host on port `1389`:

```
$ dsreplication disable -h localhost -p 1389 -D "cn=directory manager" \
  -j pwd-file -b "cn=schema" -X
```

> **Note:** The previous example does not disable schema replication for the entire topology. To disable schema replication for the entire topology, you must run the equivalent command for each directory server in the topology.

# 26.7 Replicating to a Read-Only Server

The Oracle Unified Directory replication model is a multi-master model, that is, all the replication servers in the topology can process both read and write operations. However, you can configure a directory server to be read-only, in which case add, modify, and delete operations from LDAP clients are rejected on this server.

> **Note:** A read-only directory server functions like a *consumer replica* does in the Oracle Directory Server Enterprise Edition replication model.

### 26.7.1 Configuring a Replica as Read-Only

This example assumes a replication configuration with replication servers on two hosts, host1 and host2. The example makes the directory server on host2 a read-only replica. The example uses the `dsconfig` command, which accesses the server configuration via the administration connector. For more information, see Section 14.3, "Managing Administration Traffic to the Server".

Use the `dsconfig` command to set the `writability-mode` of host2.

```
$ dsconfig -h host2 -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  set-global-configuration-prop --set writability-mode:internal-only
```

A writability mode of *internal-only* means that replication operations are processed on the server, but the server is not writeable directly by LDAP client applications.

## 26.8 Detecting and Resolving Replication Inconsistencies

Directory server replication has been designed to ensure that replicated databases remain consistent, even in the case of hardware faults, directory server restarts, or network failures. Despite these efforts, however, it is possible that hardware failures (disk errors, memory errors) or software errors (causing memory corruption) might lead to inconsistent databases.

These topics explain how to detect replication inconsistencies, and how to resolve them when they are identified.

### 26.8.1 Types of Replication Inconsistencies

When inconsistencies occur, they might remain hidden for some time or they might trigger replication or application errors. Examples of inconsistencies include the following:

- An entry is present on all but one directory server in the replication topology.
- An entry has a DN on one directory server that is different to its DN on all other directory servers.
- An entry has different attributes on one directory server than on other directory servers in the replication topology.

### 26.8.2 Detecting Inconsistencies

Use the following methods to check for replication inconsistencies:

- **Check for information in the replication log file**. The replication log file is configured by default and lists inconsistencies that are detected by the replication mechanism. Imagine, for example, that a modify operation is performed on an entry that is missing from one directory server in the topology. When replication attempts to replay this operation to that server, it will detect the problem and produce an error in the `logs/replication` error log. This kind of error will not stop replication, but the operation will not be replayed and the administrator will need to repair the inconsistency.
- **Pay attention to errors reported by client applications or users**. Client applications or users might experience errors when accessing the directory server that might be due to replication inconsistencies.

- **Make regular checks for database consistency**. With the current directory server release, these checks must be performed manually, using searches or database exports.

### 26.8.3 Resolving Inconsistencies

If a replication inconsistency is found on a single directory server in the topology, it is not possible to fix this inconsistency using regular LDAP operations. This is because the LDAP operation itself would be replicated to the other directory servers in the topology and might cause damage on those servers. In addition, the fix might involve modifying attributes that are generated by the directory server, such as the `entryuid` or `modifyTimestamp` attributes. Such attributes cannot be modified by regular LDAP operations.

Replication repair operations must therefore be done using LDAP operations that specify the Replication Repair Control (OID: `1.3.6.1.4.1.26027.1.5.2`).

> **Caution:** Because the replication repair control allows you to skip several controls usually done by the directory server, it should be used with great care and only when consistency problems have been detected and asserted.

The repair control alters the regular processing of an operation as follows:

- The operation can modify attributes that might not normally be modified or added (`NO-USER-MODIFICATION`), such as `entryuuid` and `ds-sync-hist`.
- No replication change number is associated with the operation.
- The operation is not published to the replication server and is therefore a local-only operation.
- Replication does not try to resolve conflicts or to generate historical information for this operation.
- Most of the schema checks are not performed for this operation.

For example, the following `ldapmodify` operation repairs an entry on `host1` only, with the changes contained in the file `changes.ldif`:

```
$ ldapmodify -J 1.3.6.1.4.1.26027.1.5.2 -h localhost -p 1389 \
  -D "cn=Directory Manager" -j pwd-file -f changes.ldif
```

When you repair an entry, you must repair all of its regular attributes as well as the attributes generated by the directory server, such as `modifyTimestamp`, `modifiersName`, `createTimestamp`, `creatorsName`, and `ds-sync-hist`. The values of these attributes should be read from a directory server that contains the correct values, and recreated on the server with faulty values.

The `ds-sync-hist` attribute contains historical information that replication uses to solve modify conflicts. This attribute can only be viewed by an administrator.

### 26.8.4 Solving Naming Conflicts

Entries with identical DNs can be created on separate directory servers if they are created before the servers replicate the changes to each other. When the remote operation is replicated to the local server, a naming conflict occurs. The naming conflict results in the creation of a *conflict entry* on the local server.

Conflict entries have a specific DN, of the form `entryuuid=entryUid+oldRDN`. Every conflict entry includes a `ds-sync-conflict` attribute, whose value is the DN of the conflicting regular entry.

For example, imagine that the entry `cn=bjensen,ou=People,dc=example,dc=com` is created simultaneously on two directory servers. The entry on server 1 is given a unique ID of uid1 and the entry on server 2 is given a unique ID of uid2. Both directory servers will have the following two entries after replication:

```
cn=bjensen,dc=example,dc=com
...
entryuuid=uid2+cn=bjensen,dc=example,dc=com
  ds-sync-conflict:cn=bjensen,dc=example,dc=com
```

When you have identified the conflicting entry, you can rename it so that it has a unique DN.

If the naming attribute in a conflicting entry is multi-valued, you can rename the conflicting entry as follows:

1. Rename the entry while keeping the old RDN value, for example:

   ```
   $ ldapmodify -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file
   dn: entryuuid=uid2+cn=bjensen,dc=example,dc=com
   changetype: modrdn
   newrdn: cn=bljensen
   deleteoldrdn: 0
   ^D
   ```

   You cannot delete the old RDN value in this step because it also contains the `entryuuid` operational attribute, which cannot be deleted.

2. Remove the old RDN value of the naming attribute and the conflict marker attribute, for example:

   ```
   $ ldapmodify -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file
   dn: cn=bljensen,dc=example,dc=com
   changetype: modify
   delete: cn
   cn: bjensen
   delete: ds-sync-conflict
   ^D
   ```

If the naming attribute in a conflicting entry is single-valued, for example `dc` (domain component), you cannot simply rename the entry to another value of the same attribute. Instead, you must give the entry a temporary name, as follows:

1. Rename the entry by using a different naming attribute, and keep the old RDN, for example::

   ```
   $ ldapmodify -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file
   dn: entryuuid=uid2+dc=HR,dc=example,dc=com
   changetype: modrdn
   newrdn: o=TempHR
   deleteoldrdn: 0
   ^D
   ```

   You cannot delete the old RDN value in this step because it also contains the `entryuuid` operational attribute, which cannot be deleted.

2. Change the desired naming attribute to a unique value and remove the conflict marker attribute, for example:

```
$ ldapmodify -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file
dn: o=TempHR,dc=example,dc=com
changetype: modify
replace: dc
dc: NewHR
delete: ds-sync-conflict
^D
```

3. Rename the entry back to the intended naming attribute and delete the temporary RDN, for example:

```
$ ldapmodify -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file
dn: dc=NewHR,dc=example,dc=com
changetype: modrdn
newrdn: dc=NewHR
deleteoldrdn: 1
^D
```

## 26.9 Purging Historical Replication Data

Oracle Unified Directory maintains a history of all changes that have been made on the server as a result of replication operations. This historical replication data is stored in an attribute of each user entry, and can eventually take up a large amount of space on your disk. Historical information is therefore purged when an entry is modified, or when you specifically run a command to purge the data.

By default, information that is older than one day is purged. You can specify the age of data that should be purged by setting the value of the `conflicts-historical-purge-delay` property of the replication domain. The following example specifies that data older than five days should be purged. Note that the value of the property is expressed in minutes.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  set-replication-domain-prop --provider-name "Multimaster Synchronization" \
  --domain-name dc=example,dc=com --set conflicts-historical-purge-delay:7200m
```

You can also purge historical data immediately, or schedule a task to purge the data at a specific time. Imagine, for example, that you initialize a server with a large number of entries, then perform a significant number of changes on these entries. The resulting replication historical data will increase the size of the database quite substantially. If your server is then used mainly for read operations, the large database size remains, because no modifications are made to trigger a purge of the historical data. In this case, you can launch a once off purge task to remove the historical data that was generated by the initial modifications, and return the database to a more accurate size.

Because the purge process can take some time, you are required to specify the maximum duration of the purge (in seconds). To purge historical data immediately, run the following command:

```
$ dsreplication -h localhost -p 4444 --adminUID admin --adminPasswordFile pwd.txt
\
  purge-historical --maximumDuration 3600 --baseDN dc=example,dc=com -X -n
```

For information about scheduling commands as tasks, see Section 14.4, "Configuring Commands As Tasks".

## 26.10 Using Isolated Replicas

An *isolated replica* is a directory server that can accept changes from other replicas for replay but cannot send changes to the replication server to which it is connected. An isolated replica cannot be the source of data updates to the topology. You can use isolated replicas to separate a directory server from the rest of the replication topology.

Every directory server in the topology has a `trusted` configuration property that is set to `true` by default. Isolated replicas are identified as such by configuring them as *untrusted* servers in the topology, that is, by setting the `trusted` configuration property to `false`. Data that comes from an untrusted directory server is discarded by a replication server. This ensures that an isolated replica cannot be the source of data updates in the replication topology.

Only *directory servers* are configured as trusted or untrusted. Replication servers do not have the trusted configuration flag.

To configure a directory server as untrusted, use the `dsreplication set-trust` command, as follows:

```
$ dsreplication --adminUID admin --adminPasswordFile pwd.txt -X \
  set-trust --trustedHost host1 --trustedPort 4444 \
  --modifiedHost host2 --modifiedPort 5444 --trustValue untrusted
```

The `dsreplication set-trust` command is supported in both interactive and non-interactive modes.

The configuration of trusted and untrusted servers is subject to the following restrictions:

- You can only configure the trust flag of a directory server from another trusted server in the topology. You cannot configure the trust flag from that server itself. The `-trustedHost` and `--modifiedHost` options can therefore not refer to the same directory server.
- When you modify a directory server from `untrusted` to `trusted`, the host that is being modified must be running, otherwise the command will fail.
- When you modify a directory server from `untrusted` to `trusted`, the host that is being modified must not contain any *untrusted changes*. An untrusted change is a change that has been made on an untrusted directory server and has therefore not been propagated to the rest of the topology. If the host that is being modified contains untrusted changes, the affected suffixes should be re-initialized with an appropriate data set from one of the trusted servers in the topology before the host is modified to trusted.
- If you modify the schema on an untrusted server, that server cannot be reconfigured as a trusted server. In this case, the server instance must be deleted and recreated.

Use the `dsreplication status` command to determine whether a directory server is trusted or untrusted. For example:

```
$ dsreplication status --adminUID admin --adminPasswordFile pwd.txt -X \
  --hostname host1 --port 4444
```

### 26.10.1 Deployment Scenarios for Isolated Replicas

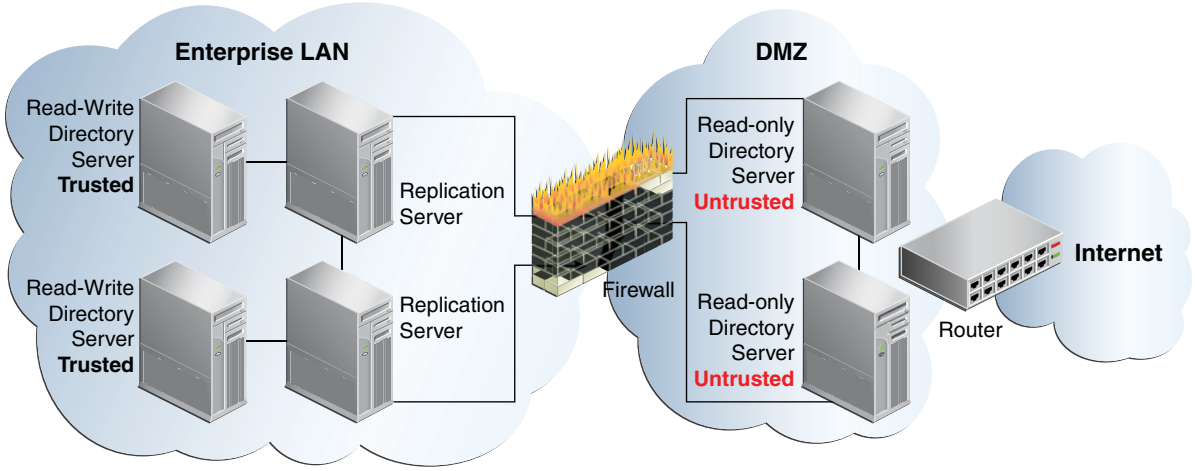
There are two main scenarios for using isolated replicas in a replication topology:

- Providing additional security in a demilitarized zone (DMZ)
- Testing client applications in a staging area

### 26.10.1.1 Using Isolated Replicas in a DMZ

A demilitarized zone (DMZ) is the area in an enterprise network that is exposed to an untrusted network, such as the Internet. A DMZ provides a layer of protection because it stands between a trusted and untrusted network. Direct access from the outside is limited to the equipment located inside the DMZ. The following figure shows how isolated replicas can be used in a DMZ.

***Figure 26–2 Isolated Replicas in a Demilitarized Zone***




By placing read-only directory servers in the DMZ, you can prevent compromised data from being transmitted to the replication servers in the private area of your network. When you deploy a replica in a DMZ, the replica is not protected by the enterprise firewall and might therefore at risk of being compromised. In such case, an unauthorized user might obtain access to the configuration of the replica and change it into a writable replica. Such a replica is therefore tagged as *untrusted* by the replication servers that are protected by the firewall.

Configuring the servers in the DMZ as untrusted safeguards against malicious data being accepted from them. The servers inside the private area are configured to have read and write access. This configuration ensures that data changes are propagated throughout the replication topology, only by the directory servers in the private area. The read-only directory servers in the DMZ obtain data changes from the replication servers located inside the private network. If an outside attacker attempts to compromise data, the direct access point is a read-only server inside the DMZ. Malicious data cannot be transmitted because directory servers in the DMZ are untrusted. The integrity of the server data inside the private enterprise LAN is therefore protected.

This scenario has the following configuration requirements:

- Each directory server in the DMZ is configured as untrusted *and* as read-only.
- Each replication server in the topology is located inside the private enterprise LAN.
- Each directory server in the private enterprise LAN is configured as a trusted server with read-write access.

Each trusted directory server in this topology has the following access rights:

- Can send changes to the replication server to which it is connected. Those changes will be propagated to all other directory servers in the topology.

- Can replay changes sent by the replication server to which it is connected.
- Can be the source of an online full update operation to initialize other servers with its data.

Each untrusted directory server in this topology has the following access limitations:

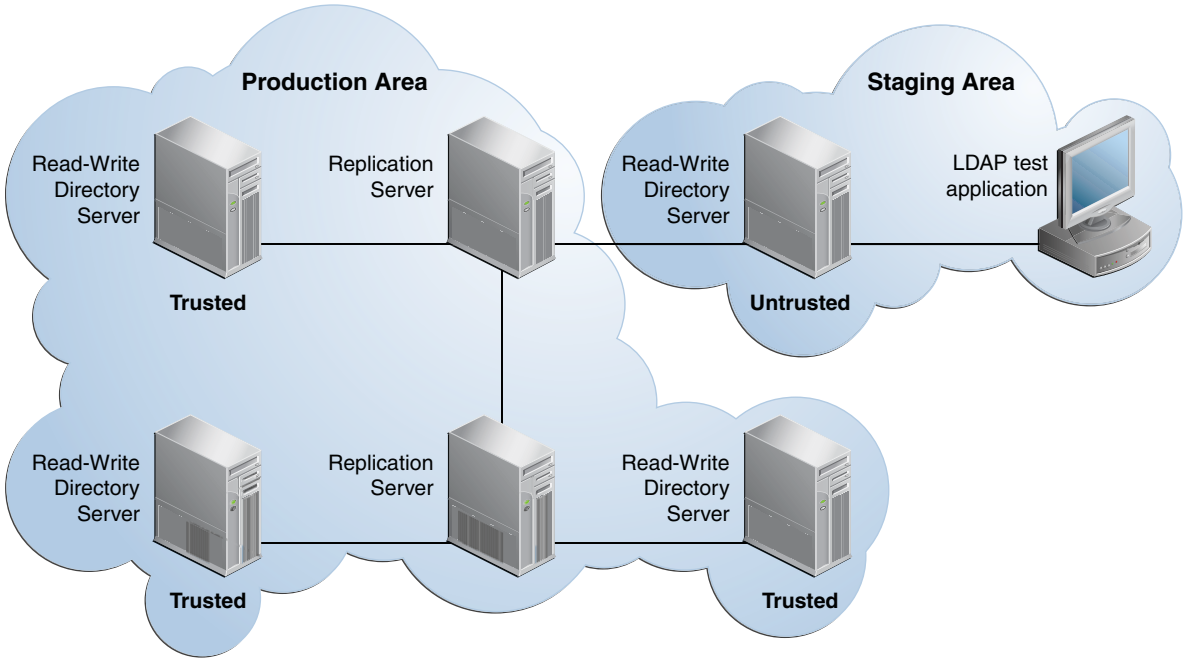
- Is not authorized to send changes to the replication server to which it is connected. If an untrusted directory server sends changes, the changes are evaluated as compromised data, and the replication server discards the changes.
- Can replay changes sent by the replication server to which it is connected.
- Cannot be the source of an online full update operation to initialize other servers with its data.

### 26.10.1.2 Using Isolated Replicas for Testing

Isolated replicas can be useful to test an application against live data in a staging area. This can be accomplished by configuring the isolated replicas to be untrusted, but with read and write access. The application's access point is the isolated replica and data is written only to the isolated replicas in the staging area.

The following figure shows how isolated replicas can be used in a staging area.

*Figure 26–3 Isolated Replicas in a Staging Area*




## 26.11 Replicating Between Oracle Directory Server Enterprise Edition and Oracle Unified Directory

Oracle Unified Directory provides a mechanism to replicate data between Oracle Directory Server Enterprise Edition and Oracle Unified Directory. The main purpose of this replication gateway is to enable migration from Oracle Directory Server Enterprise Edition to Oracle Unified Directory.

Setting up replication between these two disparate topologies involves three steps:

- Migrating the Oracle Directory Server Enterprise Edition schema and configuration to the Oracle Unified Directory server.
- Configuring replication between the Oracle Directory Server Enterprise Edition server and the Oracle Unified Directory server.
- Initializing the Oracle Unified Directory server with the data from the Oracle Directory Server Enterprise Edition server.

The following procedures describe each step. These procedures assume that you have the following:

- An installed and running Oracle Directory Server Enterprise Edition server.

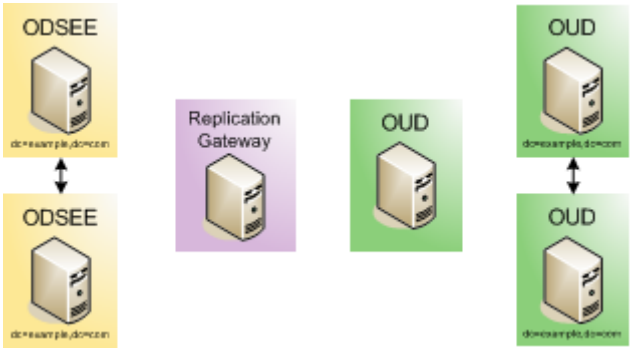
  The Oracle Unified Directory replication gateway supports the DS6-mode password policy only. If your Oracle Directory Server Enterprise Edition instance is using a DS5-mode password policy, you must upgrade it.

- An installed and running Oracle Unified Directory directory server.

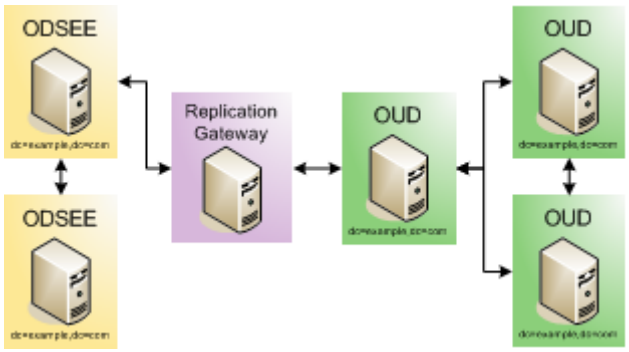
  The Oracle Unified Directory server must be configured *without* any suffixes, because that server is initialized with the data from the Oracle Directory Server Enterprise Edition server.

  If you have an existing, replicated Oracle Unified Directory topology, create an additional Oracle Unified Directory server instance, with no suffixes, and attach that server to the replication gateway. All `ds2oud` commands should be run on that empty Oracle Unified Directory server. When replication is working between the Oracle Directory Server Enterprise Edition server and the Oracle Unified Directory server, you can add the Oracle Unified Directory server to the existing replicated Oracle Unified Directory topology.

  For example, assuming an existing Oracle Unified Directory topology, your server layout prior to migration would be as follows:




  After migration, your server layout would be as follows:




## 26.11.1 To Migrate the Oracle Directory Server Enterprise Edition Schema and Configuration

Oracle Unified Directory allows migration of the configuration and the schema of Sun ONE Directory Server 5.2, Sun Java System Directory Server Enterprise Edition 6.3.1, Sun Directory Server Enterprise Edition 7.0, and Oracle Directory Server Enterprise Edition 11g Release 1 (11.1.1) including all patchsets. The migration of this types of instances can be done using the `ds2oud` command tool. The support of these versions of directory is only available for the tool `ds2oud`, but it does not apply to the use of the replication gateway which still requires at least a Oracle Directory Server Enterprise Edition 11g Release 1 (11.1.1).

In other words, depending on the instance version you migrate, the resulting Oracle Unified Directory instance requires supplementary manual steps to be fully functional, including modifying the data with respect to objectclasses and password policies, and converting metadata. However, if you run at least Oracle Directory Server Enterprise Edition 11g Release 1 (11.1.1), then it automatically takes care of data conversions while exporting the user data as described in Step 2 a in this section.

The procedure in this section describes various options of the `ds2oud` command. You can run the `ds2oud` command completely interactively by typing `ds2oud` on the command line. In interactive mode, the command prompts you for the required responses. For more information, see Section A.2.3, "ds2oud".

1. On the Oracle Unified Directory directory server, run the `ds2oud --diagnose` command, providing the connection details of the Oracle Directory Server Enterprise Edition server. The `ds2oud` command is located in `instance_dir/OUD/bin` for Linux and `instance_dir\OUD\bat` for Windows.

   This command assesses the Oracle Directory Server Enterprise Edition server instance and informs you whether any of the server configuration must be migrated to the Oracle Unified Directory server.

   ```
   $ ds2oud --diagnose -h host1.example.com -p 1389 \
     -D "cn=directory manager" -j pwdfile
   ```

   The `--diagnose` subcommand identifies the following elements of an Oracle Directory Server Enterprise Edition configuration:

   - any enabled user plug-ins
   - enabled subtree entry counter plug-ins (subtree entry counter plug-ins are not supported in Oracle Unified Directory)
   - extensions to the default schema

- any CoS or role definitions
- macro ACIs
- ACI syntax validity
- the type of password policy (only DS6-mode is supported)
- conflicting entries in the data
- encrypted attributes (attribute encryption is not supported in Oracle Unified Directory)

2. To verify data compliance with regard to the Oracle Unified Directory schema:

   a. Export the Oracle Directory Server Enterprise Edition data to LDIF.

      On the Oracle Directory Server Enterprise Edition server, run the `dsconf export` command as shown in the following example:

      ```
      $ dsconf export -f opends-export -h host1.example.com -p 1389 \
        dc=example,dc=com odsee-data.ldif
      ```

      **Note:** The option `-f opends-export` in the preceding command is only applicable for Oracle Directory Server Enterprise Edition 11*g* Release 1 (11.1.1).

      When you use the `opends-export` option during migration, DSEE-specific attributes might exist in some entries, preventing these entries from being imported. For instance, `nds5replconflict` might exist in the Oracle Directory Server Enterprise Edition data.Therefore, it is imperative to filter this attribute during import to Oracle Unified Directory using the following import option:

      ```
      --excludeAttribute "nsds5replconflict"
      ```

   b. When you have exported the data to LDIF, run the `ds2oud` command on the Oracle Unified Directory. For example:

      ```
      $ ds2oud --ldifDBFile odsee-data.ldif --userSchemaFile 99user.ldif
      ```

      where `odsee-data.ldif` is the Oracle Directory Server Enterprise Edition data exported to LDIF and `99user.ldif` is the customized Oracle Directory Server Enterprise Edition schema file, if you have customised the Oracle Directory Server Enterprise Edition schema.

      This command highlights any schema inconsistencies between the Oracle Directory Server Enterprise Edition data and the Oracle Unified Directory schema. Any schema extensions required by the Oracle Directory Server Enterprise Edition data must be added to the Oracle Unified Directory schema before you migrate the data.

3. Run the `ds2oud` command with one or more of the migration options to migrate the schema, the server configuration, or both.

   You must migrate the schema *before* you migrate the configuration, so that Oracle Unified Directory can validate the data.

   a. Running `ds2oud --migrateUserSchema` adds the Oracle Directory Server Enterprise Edition user schema (usually located in the file `99user.ldif`) to the Oracle Unified Directory schema.

If you plan to replicate collective attributes or password policy features in Oracle Unified Directory, you must prepare the Oracle Directory Server Enterprise Edition schema for these features. Oracle Unified Directory provides a customer schema file (`99OudSchemaExtract.ldif`) located in: `install-dir/OracleUnifiedDirectory/config/ds2oud` that enables you to add Oracle Unified Directory-specific schema elements to an Oracle Directory Server Enterprise Edition instance.

For information about adding this schema file to the Oracle Directory Server Enterprise Edition schema, see "*Extending Schema With a Custom Schema File*" in Oracle Directory Server Enterprise Edition Administration Guide.

**b.** Running `ds2oud --migrateConfiguration` does the following:

- Creates the naming contexts based on the existing Oracle Directory Server Enterprise Edition suffixes. You can specify whether the naming contexts are created in a single shared workflow element (`userRoot`) or in a workflow element per suffix. If the configuration includes sub-suffixes, one workflow element per suffix is imposed.
- Migrates certain global configuration parameters that apply to Oracle Unified Directory, including `size-limit`, `lookthrough-limit`, `idle-time-limit`, `max-psearches`, and `bind-with-dn-requires-password`.
- Migrates the global and backend `allidsthreshold` parameters to the Oracle Unified Directory `index-entry-limit` backend property.
- Adds any configured indexes, and migrates specific `allidsthreshold` parameters on the index or index type to the new indexes.
- Translates the DSE ACI into `ds-cfg-global-aci`, and checks the validity of ACIs by using Oracle Unified Directory syntax validation.
- Migrates the plug-in configuration if possible for the following plug-ins: 7-bit check, UID uniqueness, Referential Integrity, Strong password policy check.
- Sets up a password policy and configures the default password policy to be equivalent to the default Oracle Directory Server Enterprise Edition password policy. Note that migration is possible only for Oracle Directory Server Enterprise Edition servers that are using a DS6-mode password policy.

**c.** To migrate the schema and the configuration parameters, run the following command:

```
$ ds2oud --migrateAll -D "cn=directory manager" -j pwdfile \
  -h host1.example.com -p 1389 \
  --oudBindDN "cn=directory manager" --oudBindPasswordFile pwdfile \
  --oudHostname localhost --oudAdminPort 4444 --oudPort 1389
```

where `-D`, `-j` `-h` and `-p` specify the connection parameters of the Oracle Directory Server Enterprise Edition instance.

Most ACIs are stored in the entries themselves, and are therefore migrated when you export the data from the Oracle Directory Server Enterprise Edition instance and import it to the Oracle Unified Directory instance. The `--migrateAll` subcommand migrates only global ACIs that are stored in the configuration.

You are prompted for additional information relating to the Oracle Unified Directory configuration. This command creates a compatible configuration on the Oracle Unified Directory directory server.

## 26.11.2 To Configure Replication Between Oracle Directory Server Enterprise Edition and Oracle Unified Directory

Install and configure the replication gateway, as described in "Setting Up the Replication Gateway" section in *Oracle Fusion Middleware Installation Guide for Oracle Unified Directory*.

At this point you must configure a global administrator on the Oracle Unified Directory server, for replication. If you intend to connect this server to an existing replicated Oracle Unified Directory topology at a later stage, use the same global administrator credentials that you have defined on the other Oracle Unified Directory servers.

## 26.11.3 To Initialize the Oracle Unified Directory with Oracle Directory Server Enterprise Edition Data

1. Prepare the Oracle Unified Directory server to be initialized. For example:

```
$ dsreplication pre-external-initialization -h localhost -p 4444 \
  --adminUID admin --adminPasswordFile pwd.txt --baseDN dc=example,dc=com \
  -X -n --noPropertiesFile
```

2. On the Oracle Directory Server Enterprise Edition server, run the following command to export the data set:

```
$ dsadm export -f opends-export dsee-instance-path baseDN exportedLDIFPath
```

   where *exportedLDIFPath* is the path of the resulting LDIF file that contains the replicated data.

   If the Oracle Directory Server Enterprise Edition data includes encrypted attributes, decrypt them with the `--decrypt-attr` option.

   **Note:** `dsadm export` creates a file in LDIF format.

   `dsadm backup` creates a binary copy of the database files of the Oracle Directory Server Enterprise Edition server. Because the database implementations of Oracle Directory Server Enterprise Edition and Oracle Unified Directory are very different, you cannot use the binary copy to export data from one server type to another.

3. Copy the LDIF file that was generated in step 1 to a directory that is accessible by the Oracle Unified Directory server. Ensure that the file permissions on the LDIF file allow read access by the server.

4. On the Oracle Unified Directory server, import the LDIF data, as follows:

```
$ import-ldif -h localhost -p 4444 -D "cn=directory manager" -j pwd-file \
  --includeBranch dc=example,dc=com --ldifFile path/to/exportedLDIFFile \
  --clearBackend --trustAll --noPropertiesFile
```

   Note that if you use a relative path to the LDIF file, the root for the relative path is the instance root, rather than the current working directory. So, for example, a

path of `imports/odsee-data.ldif` here refers to `instance-root/imports/odsee-data.ldif`.

5. Run the post-initialization script on the Oracle Unified Directory server, for example:

```
$ dsreplication post-external-initialization -h localhost -p 4444 \
  --adminUID admin --adminPasswordFile pwd.txt --baseDN dc=example,dc=com \
  -X -n --noPropertiesFile
```

6. To test that replication is working correctly, modify at least one entry on each Oracle Directory Server Enterprise Edition server and check the modification on the Oracle Unified Directory server.

# 27

# Managing Directory Schema

The *schema* defines and governs the types of information objects that can be stored in a directory. A schema defines the types of entries in the directory information tree, maintains element uniqueness, and prevents unchecked schema growth that can arise when new elements are added to the directory.

This chapter provides instructions on viewing and extending the schema provided with the directory server, and covers the following topics:

- Section 27.1, "Oracle Unified Directory Schema Overview"
- Section 27.2, "Configuring Schema Checking"
- Section 27.3, "Working With Object Identifiers (OIDs)"
- Section 27.4, "Extending the Schema"
- Section 27.5, "Replicating the Schema"
- Section 27.6, "Managing the Schema With Oracle Directory Services Manager"

For detailed information about specific schema elements, see Chapter 9, "Understanding the Oracle Unified Directory Schema Model."

## 27.1 Oracle Unified Directory Schema Overview

A directory server instance reads the schema once at startup and then uses the schema information to match a search filter request or assertion to an entry's attributes to determine if any add or modify operations are permitted by the client.

In most cases, the default schema should be sufficient for most applications. However, you can take advantage of the flexibility of the directory server to extend the schema to suit your applications. The general procedure is not to relinquish the standard schema to a new custom schema, but to use the standard attributes or object classes wherever possible. If you require custom attributes or object classes that are not handled with the standard schema, you can create or extend the standard schema with auxiliary attributes and object classes required for your application.

The schema is stored in the directory under the suffix (`cn=schema`). The directory server also has a subschema subentry that defines the schema elements plus the set of operational attributes in the directory.

You can extend the schema in one of two ways:

- Extend the schema over LDAP.
- Create a custom schema definition file.

### 27.1.1 Designing and Extending the Schema

Before you consider extending the default schema, or designing your own schema, ensure that you have a solid understanding of schema syntax and design.

The basic steps to design or extend a schema are as follows:

1. Map the data to the default schema. Where possible, use the existing schema elements that are defined in the directory server. Standard schema elements help to ensure compatibility with directory-enabled applications. Because the schema is based on the LDAP standard, it has been reviewed and agreed upon by a large number of directory users.
2. Identify unmatched data. The default schema was designed to accommodate a large variety of information objects. However, if the schema does not handle your specific data type, then make note of it and any other data types needed for your directory.
3. Extend the default schema to define new elements. For optimal performance, reuse existing schema elements wherever possible. Also, minimize the number of mandatory attributes that you define for each object class. Keep the schema as simple as possible. Do not define more than one object class or attribute for the same purpose.
4. Use schema checking. Schema checking ensures that attributes and object classes conform to the schema rules.
5. Select and apply a consistent data format. The LDAP schema allows you to place any data on any attribute value. However, you should store data consistently by selecting a format appropriate for your LDAP client application and directory users.

### 27.1.2 Default Schema Files

The default schema provided with the directory server is a collection of LDIF files stored in `OUD_ORACLE_HOME/config/schema`. These schema files are applied to every server instance that is associated with that `OUD_ORACLE_HOME`.

A directory server instance loads the schema files in alphanumeric order (numerals first) at server startup.

**Caution:** Never modify the standard schema definitions and internal operational attributes in these files.

The following table describes the default schema files and their contents.

***Table 27–1 Default Schema Files***

| Schema File | Description |
|---|---|
| `00-core.ldif` | Contains the schema definitions for the LDAPv3 standard user and organization. |
| `01-pwpolicy.ldif` | Contains the schema definitions for password policies based on the `draftldappolicy` draft. |
| `02-config.ldif` | Contains the schema definitions for the attribute and object class definitions in the directory configuration file. |
| `03-changelog.ldif` | Contains the schema definitions for storing changes to directory data based on the `draftldap-changelog`. |

*Table 27–1 (Cont.) Default Schema Files*

| Schema File | Description |
|---|---|
| 03-rfc2713.ldif | Contains the schema definitions for representing Java objects in an LDAP directory based on RFC 2713. |
| 03-rfc2714.ldif | Contains the schema definitions for representing CORBA object references in an LDAP directory based on RFC 2714. The Common Object Request Broker Architecture (CORBA) integrates machines in a multivendor, multiplatform environments using CORBA objects. A directory server can be a repository for CORBA object references, which allow for a centrally administered service for CORBA-compliant applications. |
| 03-rfc2739.ldif | Contains the schema definitions for representing calendar attributes for a vCard directory based on RFC 2739. Calendar applications require a calendar user agent to locate a URI, located in a directory, for an individual's calendar. Note that the definition in RFC 2739 contains a number of errors. This schema file has been altered from the standard definition in order to fix a number of those problems. |
| 03-rfc2926.ldif | Contains the schema definitions for mapping Service Location Protocol (SLP) advertisements based on RFC 2926. This specification allows directory servers to serve SLP directory agent back ends that create mappings between SLP templates and the LDAP directory schema. |
| 03-rfc3112.ldif | Contains the schema definitions for the authentication password syntax based on RFC 3112. |
| 03-rfc3712.ldif | Contains the schema definitions for storing printer information in the directory based on RFC 3712. |
| 03-uddiv3.ldif | Contains the schema definitions for storing UDDI v3 information in the directory based on RFC 4403. Universal Description, Discovery and Integration (UDDI) is a platform-independent, XML-based registry for companies on the Internet. UDDI enables companies to publish service listings and defines which software applications interact together over the Internet. |
| 04-rfc2307bis.ldif | Contains the schema definitions for storing naming service information in the directory based on draftrfc2307bis. |
| 05-rfc4876.ldif | Contains schema definitions from RFC 4876, which defines a schema for storing Directory User Agent (DUA) profiles and preferences. |
| 05-solaris.ldif | Contains schema definitions required for Solaris and OpenSolaris LDAP naming services. |
| 06-compat.ldif | Contains the attribute type and objectclass definitions for use with the directory server configuration. |
| 10-ad-paging.ldif | Contains schema definitions required for the Active Directory paging function. |
| 10-distribution.ldif | Contains the schema definitions required for the distribution functionality of a proxy server instance. |
| 10-global-index-catalog.ldif | Contains the schema definitions required for the global indexing functionality of a proxy server instance. |
| 10-loadbalancing.ldif | Contains the schema definitions required for the load balancing functionality of a proxy server instance. |
| 10-proxy.ldif | Contains the schema definitions specific to a proxy server instance. |
| 10-replication-gateway.ldif | Contains the schema definitions specific to a replication gateway server instance. |
| 10-virtualization.ldif | Contains the schema definitions required for the virtualization functionality of a proxy server instance. |

## 27.2 Configuring Schema Checking

Oracle Unified Directory provides a schema-checking mechanism that verifies whether newly-written or added entries conform to the directory server's schema. This mechanism ensures that data imported using `import-ldif`, or added using `ldapmodify`, meets the syntax rules of the schema.

The schema checking configuration is part of the advanced global configuration, and can be displayed with the following command:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  --advanced get-global-configuration-prop
Property                              : Value(s)
--------------------------------------:--------------------------------------
...
check-schema                          : true
...
invalid-attribute-syntax-behavior     : reject
...
single-structural-objectclass-behavior : reject
...
```

The following configuration properties control schema-checking:

- `check-schema`. Possible values: `true` (default), `false`. This property controls whether the directory server should do schema-checking on newly imported or added entries. By default, the property is set to true. If you need to tune the server for maximum performance and you are certain that your clients will never make a change that causes a schema violation, you can set the property to `false`. The small performance benefits are minimal compared to the potential risks to your directory.
- `invalid-attribute-syntax-behavior`. Possible values are: `reject` (default), `accept`, and `warn`. This property controls how the server should behave if an attempt is made to use an attribute value that violates the associated syntax. By default, the server rejects any requests to use attributes that violate the schema. If this property is set to `accept`, the server silently accepts attribute violations. If this attribute is set to `warn`, the server accepts violations, but writes a message to the error log. If the `check-schema` property is set to `false`, invalid attribute syntax checking is not enforced.
- `single-structural-objectclass-behavior`. Possible values are: `reject` (default), `accept`, and `warn`. This property controls how the server should behave if an attempt is made to create or alter an entry that does not have exactly one structural object class. This means that object classes with no structural object classes or more than one are rejected by default. If this property is set to `accept`, entries with no structural object classes are allowed. If this property is set to `warn`, entries with no structural object classes (or more than one) are allowed, but a message is written to the error log. If the `check-schema` property is set to `false`, single structural object class checking is not enforced.

> **Caution:** Changing the value of these properties from the default puts the integrity of the schema at risk, so you should generally *not* alter these values.

## 27.3 Working With Object Identifiers (OIDs)

An object identifier (OID) is a numeric string used to uniquely identify an object in a directory. OIDs are used in directory schema, controls, and extended operations that require unique identification of elements.

LDAP object classes and attributes require a base object identifier (OID) that must be unique within your organization to avoid naming conflicts in the directory. If you plan to use your directory internally within your organization, use the OIDs provided in the directory server. If you plan to export your schema or publicly expose your schema in any way, consider entering a request for a unique OID for your organization. For more information, see Section 27.3.1, "Obtaining a Base OID".

After you have obtained a base OID, you can add branches to it for your organization's object classes and attributes. For example, the directory server uses an assigned base OID of `1.3.6.1.4.1.26027`. For each component type, the directory server provides unique branch numbers to the base OID for each schema component.

Oracle Unified Directory provides a comprehensive set of OIDs that should be sufficient for most applications.

The following table shows the base OIDs used for each schema component:

***Table 27–2 Base OIDs Used for Each Schema Component***

| OID Value | Type |
|---|---|
| `1.3.6.1.4.1.26027.1.1` | Attribute |
| `1.3.6.1.4.1.26027.1.2` | Object classes |
| `1.3.6.1.4.1.26027.1.3` | Attribute syntaxes |
| `1.3.6.1.4.1.26027.1.4` | Matching rules |
| `1.3.6.1.4.1.26027.1.5` | Controls |
| `1.3.6.1.4.1.26027.1.6` | Extended operations |
| `1.3.6.1.4.1.26027.1.9` | General use |
| `1.3.6.1.4.1.26027.1.999` | Experimental use |

For each schema type, a unique branch number is added to the base OID. For example, attribute types use a branch number of `1` to form the OID of `1.3.5.1.4.1.26027.1.*1*`. For each specific attribute type, the directory server assigns another set of branch numbers, one for each attribute type.

The following table displays a (partial) list of assigned OID values for attribute types.

***Table 27–3 Assigned OID Values for Attribute Types***

| OID Value | Attribute Type |
|---|---|
| `1.3.6.1.4.1.26027.1.1.1` | `ds-cfg-java-class` |
| `1.3.6.1.4.1.26027.1.1.2` | `ds-cfg-enabled` |
| `1.3.6.1.4.1.26027.1.1.3` | `ds-cfg-allow-attribute-name-exceptions` |
| `1.3.6.1.4.1.26027.1.1.4` | `ds-cfg-allowed-client` |
| `1.3.6.1.4.1.26027.1.1.5` | `ds-cfg-allow-ldap-v2` |

Oracle Unified Directory allows the use of non-numeric OIDs as long as a corresponding numeric OID is defined within the schema. For example, you can use a

non-numeric OID, `mytestattribute-oid` for the named attribute, `myTestAttribute`. The non-numeric OID must be all lowercase with the `-oid` appended to the named attribute. The use of non-numeric OIDs is an LDAP-specification violation but is permissible for ease of use.

### 27.3.1 Obtaining a Base OID

If you plan to make your directory server publicly available, or if you plan to redistribute your schema definitions for custom applications, you can obtain a base OID for your organization. You can use your own OIDs in a custom schema file if you plan to create custom extensions to the directory server. Alternatively, you can modify the schema configuration files by adding your base OID with its respective branch number.

---

**Note:** Do not modify the default OIDs unless you are sure of what you are doing. Modifying the OIDs can potentially damage your directory server.

---

To obtain and create base OIDs for your organization, perform the following steps:

1. Point your browser to the Internet Assigned Numbers Authority (IANA) web site at (`http://www.iana.org`) or a national organization in your country that handles such tasks. In some countries, corporations already have OIDs assigned to them. If your organization does not already have an OID, you can fill out a request at the IANA web site.
2. Determine the unique object classes, attributes, names, and other schema elements. Ensure that the names are descriptive to make it easier to manage the schema. One trick is to add a custom prefix to your custom object classes and attributes. For example, if your organization is Example.com, you can add the prefix `Example` before each custom schema element, such as adding `Example` to a `Person` object class as in `ExamplePerson`.
3. Create an OID registry to keep track of OID assignments. The registry is nothing more than a list that you maintain to ensure that OIDs and their descriptions are unique within your directory. The registry should be sufficiently protected so that only a privileged administrator can modify the registry.
4. Create branches in the OID tree to accommodate the schema elements.
5. Shut down the directory servers in your topology.
6. Manually edit the schema configuration files on each directory server in your topology. Replace each OID with your company's OID. This avoids problems with schema replication seeing differences in the schema and attempting to synchronize the information.
7. Manually edit any custom schema extensions. Ideally, you should define any custom extensions in a separate file.

## 27.4 Extending the Schema

Oracle Unified Directory supports multiple methods to extend the schema. The standard schema files are a set of LDIF files located in `OUD_ORACLE_HOME/config/schema`. Do not modify these files directly, because doing so can result in unpredictable server behavior.

The standard schema definitions apply to every server instance associated with that OUD_ORACLE_HOME. Custom schema definitions located in *instance-dir*/OUD/config/schema/99user.ldif apply only to the server instance in which they are created.

You can extend the schema as follows:

- **Extend the schema over LDAP.** Define your schema extensions, write the definitions to an LDIF file, and add the custom schema extensions by using the `ldapmodify` command.

  When you use this method, the directory server automatically writes the new schema definitions to the file:

  *instance-dir*/OUD/config/schema/99user.ldif

  To specify a different schema file, include the `X-SCHEMA-FILE` element with the name of your schema file. For example, as part of your attribute type definition, include the element `X-SCHEMA-FILE '98myschema.ldif'`.

  When you extend the schema over LDAP, you do not need to restart the server to take the schema modifications into account.

- **Create a custom schema file.** Create a custom schema file with your definitions and move the file to the directory:

  *instance-dir*/OUD/config/schema/

  The directory server loads schema files in alphanumeric order with numbers loaded first. As such, you should name custom schema files as follows: `[00-99]filename.ldif`. The number should be higher than any standard schema file that has already been defined. If you name custom schema files with a number that is lower than the standard schema files, the server might encounter errors when loading the schema.

  When you extend the schema with a custom schema file, the server must be restarted before the schema modifications are taken into account.

- **Modify an existing schema file.** You can add a custom schema extension to an existing custom schema file, such as *instance-dir*/OUD/config/schema/99user.ldif.

  When you extend the schema by modifying an existing schema file, the server must be restarted before the schema modifications are taken into account.

When you add new schema elements, all attributes must be defined before they can be used in an object class. If you are creating several object classes that inherit from other object classes, you must create the parent object class first.

Each custom attribute or object class that you create should be defined in only one schema file.

When you define new schema definitions manually, the best practice is to add these definitions to the `99user.ldif` file or to your designated schema file.

### 27.4.1 Managing Attribute Types

You can add new attribute types to the schema by using the `ldapmodify` command. The attribute types syntax requires that you provide at least a valid OID to define a new element. In typical applications, you can optionally include the following identifiers for the attribute type. To see the full set of attribute type elements, see Section 9.3, "Understanding Attribute Types".

`OID`
Required. Specifies the OID that uniquely identifies the attribute type in the directory server. The LDAP v3 specification requires the OID to be a UTF-8 encoded dotted decimal. However, Oracle Unified Directory supports the use of non-numeric OIDs for easy identification as long as the schema is used internally within the organization. The format is `attributename-oid`, for example, `telephoneNumber-oid`. Each non-numeric OID must have its corresponding dotted decimal OID defined in the schema.

`NAME`
Optional. Specifies the set of human-readable names that are used to refer to the attribute type. If there is a single name, enclose it in single quotes, for example, `'blogURL'`. If there are multiple names, enclose each name in single quotes separated by spaces, and then enclose the entire set of names within parentheses, for example, `('blog' 'blogURL')`. Ensure that there is a space between the left parenthesis and the name, and a space before the closing parenthesis.

`SUP`
Optional. Specifies the superior attribute type when you want one attribute type to inherit elements from another attribute type. The matching rule and attribute syntax specifications from the superior attribute type can be inherited by the subordinate type if it does not override the superior attribute type definition. The OID, any of the human-readable names associated with the superior attribute type or both can be used to collectively reference all of the subordinate attribute types.

`DESC`
Optional. Specifies a human-readable description of the attribute type.

`SYNTAX`
Optional. Specifies the attribute syntax for use with the attribute type. If provided, it should be given as a numeric OID. The core syntaxes are defined in section 3.3. of RFC 4517 (`http://www.ietf.org/rfc/rfc4517.txt`) and in Appendix A of the same document.

`SINGLE-VALUE`
Optional. Specifies whether the attributes of that type are allowed to have only a single value in any entry in which they appear. If `SINGLE-VALUE` is not present, the attributes are allowed to have multiple distinct values in the same entry.

`NO-USER-MODIFICATION`
Optional. Indicates that the values of the attributes of the given type cannot be modified by external clients (that is, the values can be modified only by internal processing within the directory server).

`USAGE`
Optional. Indicates how the attribute is to be used. Possible values are as follows: `userApplications`. Used to store user data. `directoryOperation`. Used to store data required for internal processing within the directory server. `distributeOperation`. Used to store operational data that must be synchronized across directory servers in the topology. `dSAOperation`. Used to store operational data that is specific to a particular directory server and should not be synchronized across the topology.

`extensions`
Optional. Specifies the extensions available to the attribute type. Oracle Unified Directory provides the following extensions:

- X-ORIGIN. Provides information on where the attribute type is defined. The element is a non-standard tool that you can use to locate the schema element, for example, the RFC number (RFC4517).
- X-SCHEMA-FILE. Indicates which schema file contains the attribute type definition. Used for internal purposes only and is not exposed to clients. You can use this extension to specify where the directory server should store your custom schema definitions.
- X-APPROX. Indicates which approximate matching rule should be used for the attribute type. If specified, the value should be the name of the OID of a registered approximate matching rule.

For example, you can specify the addition of a new attribute type, blogURL, in an LDIF file that will be added to the schema.

```
$ cat blogURL.ldif
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: ( 1.3.6.1.4.1.32473.1.1.590
  NAME ( 'blog' 'blogURL' )
  DESC 'URL to a personal weblog'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
  SINGLE-VALUE
  X-ORIGIN 'Oracle Unified Directory Server'
  USAGE userApplications )
```

**Note:** Pay special attention to the spaces in an attribute type declaration. The LDAP specification requires that a space exist between the opening parenthesis and the OID, and the value of the USAGE element and the closing parenthesis. Further, the LDIF specification states that LDIF parsers should ignore exactly one space at the beginning of each line. Therefore, it is a good practice to add two (2) spaces at the beginning of the line that starts with an element keyword. For example, add two spaces before NAME, DESC, SYNTAX, SINGLE-VALUE, X-ORIGIN, and USAGE in the previous example.

The OIDs used in this example are for illustration purposes only and should not be implemented in your directory.

#### 27.4.1.1 To View Attribute Types

The cn=schema entry has a multivalued attribute, attributeTypes, that contains definitions of each attribute type in the directory schema. You can view the schema definitions by using the ldapsearch command. Schema elements are represented as LDAP subentries, and searches on cn=schema must therefore include the LDAP Subentry search control.

1. Use the ldapsearch command with the LDAP Subentry search control, as follows:

```
$ ldapsearch -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
  -b "cn=schema" -s base "(objectclass=*)" attributeTypes
dn: cn=schema
attributeTypes: ( 2.5.4.41 NAME 'name' EQUALITY caseIgnoreMatch SUBSTR
 caseIgnoreeSubstringsMatch SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{32768}
 X-ORIGIN 'RFC 4519' )
```

```
attributeTypes: ( 2.5.4.49 NAME 'distinguishedName' EQUALITY
 distinguishedNameMatch SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 X-ORIGIN
 'RFC 4519' )
attributeTypes: ( 2.5.4.0 NAME 'objectClass' EQUALITY objectIdentifierMatch
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.38 X-ORIGIN 'RFC 4512' )
...(more output)...
```

2. To view a specific attribute type, use the `--dontWrap` option and then use the `grep` command (on UNIX systems) to search for the required attribute.

   The following example searches for attribute types that contain the string `telexNumber`.

```
$ ldapsearch -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
  -b cn=schema -s base --dontWrap "(objectclass=*)" \
  attributeTypes | grep "telexNumber"
  attributeTypes: ( 2.5.4.21 NAME 'telexNumber' SYNTAX
1.3.6.1.4.1.1466.115.121.1.52 X-ORIGIN 'RFC 4519' )
  attributeTypes: ( 2.5.4.21.1 NAME 'c-TelexNumber' SUP telexNumber COLLECTIVE
X-ORIGIN 'RFC 3671' )
```

### 27.4.1.2 To Create an Attribute Type

The `cn=schema` entry has a multivalued attribute, `attributeTypes`, that contains definitions of each attribute type in the directory schema. You can add custom schema definitions by using the `ldapmodify` command. This example adds an attribute named `blog`.

1. Using a text editor, create an LDIF file with your schema extensions.

```
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: ( 1.3.6.1.4.1.32473.1.1.590
  NAME ( 'blog' 'blogURL' )
  DESC 'URL to a personal weblog'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
  SINGLE-VALUE
  X-ORIGIN 'Oracle Unified Directory Server'
  USAGE userApplications )
```

2. Use `ldapmodify` to add the file.

```
$ ldapmodify -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
  -a -f blogURL.ldif
Processing MODIFY request for cn=schema
MODIFY operation successful for DN cn=schema
```

3. Verify the addition by displaying it using `ldapsearch`.

```
$ ldapsearch -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
  -b "cn=schema" -s base --dontWrap "(objectclass=*)" \
  attributeTypes | grep 'blog'
attributeTypes: ( 1.3.6.1.4.1.32473.1.1.590 NAME ( 'blog' 'blogURL' )
DESC 'URL to a personal weblog' SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE X-ORIGIN 'Oracle Unified Directory Server' USAGE userApplications
)
```

---

**Note:** Oracle Unified Directory automatically adds new attribute definitions to the file *instance-dir*`/OUD/config/schema/99user.ldif`.

---

#### 27.4.1.3 To Delete an Attribute Type

The `cn=schema` entry has a multivalued attribute, `attributeTypes`, that contains definitions of each attribute type in the directory schema. You can delete custom schema definitions by using the `ldapmodify` command. Oracle Unified Directory does not allow deletions to standard schema definitions.

> **Caution:** Be careful when deleting attribute types, because doing so can harm your directory. Do not delete an attribute type unless absolutely necessary.

1. Create the delete request in an LDIF file.

```
dn: cn=schema
changetype: modify
delete: attributeTypes
attributeTypes: ( 1.3.6.1.4.1.32473.1.1.590
  NAME ( 'blog' 'blogURL' )
  DESC 'URL to a personal weblog'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
  SINGLE-VALUE
  X-ORIGIN 'Oracle Unified Directory Server'
  USAGE userApplications )
```

2. Use the `ldapmodify` command to process the delete request.

```
$ ldapmodify -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
  --defaultAdd --fileName "remove_blogURL.ldif"
Processing MODIFY request for cn=schema
MODIFY operation successful for DN cn=schema
```

### 27.4.2 Managing Object Classes

Object classes are named sets of attribute definitions that are used to control the types of data stored in entries. You can add new object classes to the schema by using the `ldapmodify` command. The object class syntax requires that you provide at least a valid OID to define your new element. In typical applications, you will also include the following optional identifiers for the object class type. For more information about the object class definition, see Section 27.1, "Oracle Unified Directory Schema Overview".

`OID`

Required. Specifies the OID that uniquely identifies the object class in the directory server. The LDAP v3 specification requires the OID to be a UTF-8 encoded dotted decimal. However, Oracle Unified Directory supports the use of non-numeric OIDs for easy identification because the schema is used internally within the organization. For example, the format is *objectClassName*`-oid`, such as `person-oid`.

`NAME`

Optional. Specifies the set of human-readable names that are used to refer to the object class. If there is a single name, enclose it in single quotes, for example, `'blogURL'`. If there are multiple names, enclose each name in single quotes separated by spaces, and then enclose the entire set of names within parentheses, for example, `('blog' 'blogURL')`. Ensure that there is a space between the left parenthesis and the name, and a space before the closing parenthesis.

DESC
Optional. Specifies a human-readable description of the object class. If specified, the description should be enclosed in single quotation marks.

SUP
Optional. Specifies the superior object class when you want it to inherit elements from another object class. The directory server allows only one superior object class, although the LDAP v3 specification allows for multiple superior object classes.

OBSOLETE
Optional. Indicates whether the object class is active or not. If an object class is marked as OBSOLETE, then it should not be referenced by any new elements created in the directory server.

SUP oids
Optional. The SUP keyword should be followed by the OID of the superior class.

KIND
Optional. Indicates the type of object class that is being defined. Allowed values are ABSTRACT, AUXILIARY and STRUCTURAL.

MUST oids
Optional. Specifies the set of attribute types that are required to be present (that is, have at least one value) in entries with that object class. If there is only a single required attribute, then the MUST keyword should be followed by the name or the OID of that attribute type. If there are multiple required attribute types, then separate them with dollar signs ($) and enclose the entire set of attribute types in parentheses. For example, MUST (sn $cn).

MAY oids
Optional. Specifies the set of attribute types that are allowed but not required to be present in entries with that object class. If there is only a single required attribute, then the MAY keyword should be followed by the name or the OID of that attribute type. If multiple required attribute types are specified, then separate them by dollar signs ($) and enclose the entire set of attribute types in parentheses. For example, MAY (userPassword $telephoneNumber $seeAlso $description).

extensions
Optional. Specifies the extensions available to the object class. The directory server provides the following extensions: X-ORIGIN. Provides information on where the object class is defined. The element is a non-standard tool that the user can use to conveniently locate the schema element. X-SCHEMA-FILE. Indicates which schema file contains the object class definition. Used for internal purposes only and is not exposed to clients. You can use this extension to specify where the directory server is to store your custom schema definitions.

For example, you can specify the addition of a new object class, blogger, in an LDIF file to be added to the schema.

```
$ cat blogger.ldif
dn: cn=schema
changetype: modify
add: objectClasses
objectClasses: ( 1.3.6.1.4.1.32473.1.1.10
  NAME ( 'blogger' )
  DESC 'Someone who has a blog'
  SUP inetOrgPerson
  STRUCTURAL
```

```
MAY blog
X-ORIGIN 'Oracle Unified Directory Server' )
```

Pay special attention to the spaces in your object class declaration. The LDAP specification requires that a space exist between the opening parenthesis and the OID, and the value of the `X-ORIGIN` element and the closing parenthesis. Further, the LDIF specification states that LDIF parsers should ignore exactly one space at the beginning of each line. Therefore, it is a good practice to add two spaces before the line that begins with an element keyword, such as, `NAME`, `DESC`, `SUP`, `STRUCTURAL`, `MAY`, and `X-ORIGIN` in the previous example.

The OIDs used in this example are for illustration purposes only and should not be implemented in your directory.

#### 27.4.2.1 To View Object Classes

The `cn=schema` entry has a multivalued attribute, `objectClasses`, that contains definitions of each object class in the directory schema. You can view the schema definitions by using the `ldapsearch` command.

1. Use the `ldapsearch` command to view object class definitions.

```
$ ldapsearch -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
  -b cn=schema -s base "(objectclass=*)" objectClasses
dn: cn=schema
objectClasses: ( 2.5.6.0 NAME 'top' ABSTRACT MUST objectClass X-ORIGIN
 'RFC 4512' )
objectClasses: ( 2.5.6.1 NAME 'alias' SUP top STRUCTURAL MUST aliasedObjectName
 X-ORIGIN 'RFC 4512' )
objectClasses: ( 2.5.6.2 NAME 'country' SUP top STRUCTURAL MUST c MAY
 ( searchGuide $ description ) X-ORIGIN 'RFC 4519' )
objectClasses: ( 2.5.6.3 NAME 'locality' SUP top STRUCTURAL MAY ( street $
 seeAlso $ searchGuide $ st $ l $ description ) X-ORIGIN 'RFC 4519' )
objectClasses: ( 2.5.6.4 NAME 'organization' SUP top STRUCTURAL MUST o MAY
 ( userPassword $ searchGuide $ seeAlso $ businessCategory $ x121Address $
 registered Address $ destinationIndicator $ preferredDeliveryMethod $
 telexNumber $ teletexTerminalIdentifier $ telephoneNumber $
 internationaliSDNNumber $ facsimileTelephoneNumber $ street $ postOfficeBox $
 postalCode $ postalAddress $ physicalDeliveryOfficeName $ st $ l $ description
 ) X-ORIGIN 'RFC 4519' )
...(more output)...
```

2. Use the `--dontWrap` option and the `grep` command to search for a specific object class.

```
$ ldapsearch -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
  -b cn=schema -s base --dontWrap "(objectclass=*)" \
  objectClasses | grep "inetOrgPerson"
objectClasses: ( 2.16.840.1.113730.3.2.2 NAME 'inetOrgPerson' SUP
organizationalPerson
STRUCTURAL MAY ( audio $ businessCategory $ carLicense $ departmentNumber $
displayName
$ employeeNumber $ employeeType $ givenName $ homePhone $ homePostalAddress $
initials
$ jpegPhoto $ labeledURI $ mail $ manager $ mobile $ o $ pager $ photo $
roomNumber
$ secretary $ uid $ userCertificate $ x500UniqueIdentifier $ preferredLanguage
$ userSMIMECertificate $ userPKCS12 ) X-ORIGIN 'RFC 2798' )
```

#### 27.4.2.2 To Create an Object Class

The `cn=schema` entry has a multivalued attribute, `objectClasses`, that contains definitions of each object class in the directory schema. You add custom schema by using the `ldapmodify` command. This example adds an object class `blogger` based on the attribute type that was created in the previous example.

1. Using a text editor, create an LDIF file with your schema extensions.

```
dn: cn=schema
changetype: modify
add: objectClasses
objectClasses: ( 1.3.6.1.4.1.32473.1.1.10
  NAME ( 'blogger' )
  DESC 'Someone who has a blog'
  SUP inetOrgPerson
  STRUCTURAL
  MAY blog
  X-ORIGIN 'Oracle Unified Directory Server' )
```

2. Use the `ldapmodify` command to add the file.

```
$ ldapmodify -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
  -a -f blogger.ldif
Processing MODIFY request for cn=schema
MODIFY operation successful for DN cn=schema
```

3. Verify the addition by displaying it with `ldapsearch`.

```
$ ldapsearch -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
  -b cn=schema -s base --dontWrap "(objectclass=*)" \
  objectClasses | grep 'blogger'
```

> **Note:** Oracle Unified Directory automatically adds new object class definitions to the file *instance-dir*`/OUD/config/schema/99user.ldif`.

#### 27.4.2.3 To Delete an Object Class

The `cn=schema` entry has a multivalued attribute, `objectClasses`, that contains definitions for each object class in the directory schema. You can delete custom object class definitions by using the `ldapmodify` command.

> **Caution:** Be careful when deleting object classes, because doing so can harm your directory. Do not delete an object class unless absolutely necessary.

1. Create the delete request in LDIF format.

```
dn: cn=schema
changetype: modify
delete: objectClasses
objectClasses: ( 1.3.6.1.4.1.32473.1.1.10
  NAME ( 'blogger' )
  DESC 'Someone who has a blog'
  SUP inetOrgPerson
  STRUCTURAL
  MAY blog
  X-ORIGIN 'Oracle Unified Directory Server' )
```

2. Remove the object class by using `ldapmodify` to apply the LDIF file.

```
$ ldapmodify -h localhost -p 1389 -D "cn=Directory Manager" -j pwd-file \
  --fileName "remove_objectclass_schema.ldif"
```

## 27.5 Replicating the Schema

In a replicated topology, schema definitions are automatically replicated to ensure that all servers use a single schema. Schema modifications on any server are replicated to all other servers in the topology.

When you configure replication, the schema of the first server is used to initialize the schema of the second server by default. You can, however, specify that the schema of the second server be used to initialize the schema of the first server. You can also specify that schema replication be disabled altogether. For more information, see Section 26.6, "Configuring Schema Replication".

## 27.6 Managing the Schema With Oracle Directory Services Manager

You can manage most elements of the directory schema with ODSM. The following topics indicate the steps to manage the most common aspects of viewing and extending the schema.

### 27.6.1 Add a New Attribute Type

You can add a new attribute type to the schema by using ODSM, as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager".
2. Select the **Schema** tab.
3. The **Attributes** panel is expanded by default. If it is not expanded, click the arrow to expand it.
4. Click the **Add** icon.
5. Complete the following information on the **Create new attribute** window:
   - **Name.** Enter a unique name for the new attribute type.
   - **Object ID.** Specify the OID that uniquely identifies the attribute type in the directory server. Oracle Unified Directory supports the use of non-numeric OIDs for easy identification as long as the schema is used internally within the organization. However, for this release ODSM supports numeric OIDs only.
   - **Description.** Enter a human-readable description of the attribute type.
   - **Syntax.** Enter the attribute syntax for use with the attribute type. If provided, the syntax should be specified as a numeric OID. The core syntaxes are defined in section 3.3. of RFC 4517 and in Appendix A of the same document.
   - **Size.** Enter a maximum size for the value of the attribute, in bytes. In the case of multi-valued attributes, this refers to the maximum size of a single value, not of the combined values.
   - **Usage.** Specify how the attribute will be used. Possible values are as follows:
     - **userApplications.** The attribute will be used to store user data.

  - **directoryOperation.** The attribute will be used to store data that is required for internal processing within the directory server.
  - **distributedOperation.** The attribute will be used to store operational data that must be synchronized across directory servers in the topology.
  - **dSAOperation.** The attribute will be used to store operational data that is specific to a particular directory server and should not be synchronized across the topology.

- **Ordering.** Select the ordering index details for this attribute type. For more information see Section 17.7, "Indexing Directory Data".
- **Equality.** Select the equality index details for this attribute type. For more information see Section 17.7, "Indexing Directory Data".
- **Substring.** Select the substring index details for this attribute type. For more information see Section 17.7, "Indexing Directory Data".
- **Obsolete.** Select this box if the attribute type is no longer in use but is retained for compatibility.
- **Single Value.** Indicate whether attributes of this type may have only a single value in any entry in which they appear. If this checkbox is not selected, the attributes may have multiple distinct values in the same entry.
- **Collective.** Indicate whether the attribute is a collective attribute. For more information, see Section 17.12, "Using Collective Attributes".
- **Super.** If this new attribute extends an existing attribute, enter or select the name of the existing super type.
- **Origin.** Enter the source of this new attribute type, for example, RFC 4512.

  To view the source of all the schema elements in the directory, select **Show All** from the **View** menu.
- **Schema File Extension.** If the attribute type's definition is contained in a file, enter the path to the file.

6. Click **Create** to create the new attribute.

## 27.6.2 Add an Attribute Based on an Existing Attribute

You can add an attribute type that is based on an existing attribute type by using ODSM, as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager".
2. Select the **Schema** tab.
3. The **Attributes** panel is expanded by default. If it is not expanded, click the arrow to expand it.
4. Select the attribute on which you want to base the new attribute type.
5. Click the **Create like** icon.
6. Certain fields are completed by default, based on the attribute that you selected.

   Complete the remaining fields for the new attribute type.

   For information about the fields and their values, see Section 27.6.1, "Add a New Attribute Type".

7. Click **Create** to create the new attribute.

### 27.6.3 Modify an Attribute

You can modify an existing attribute type by using ODSM, as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager".
2. Select the **Schema** tab.
3. The **Attributes** panel is expanded by default. If it is not expanded, click the arrow to expand it.
4. Select the attribute type that you want to modify.
5. Modify the required fields, on the right hand pane.

   For information about the fields, see Section 27.6.1, "Add a New Attribute Type".
6. Click **Apply** to save your changes.

### 27.6.4 Delete an Attribute

You can delete an existing attribute type by using ODSM, as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager".
2. Select the **Schema** tab.
3. The **Attributes** panel is expanded by default. If it is not expanded, click the arrow to expand it.
4. Select the attribute type that you want to delete.
5. Click the **Delete** icon and click **OK** to confirm the deletion.
6. Click **Apply** to save your changes.
7. Click the **Refresh** icon to refresh the list of attributes on the left hand pane and confirm that the attribute has been deleted from the schema.

---

**Note:** The server will return an error if you attempt to delete an attribute type that is already referenced by one or more entries in the server.

---

### 27.6.5 View All Directory Attributes

You can view all existing attribute types by using ODSM, as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager".
2. Select the **Schema** tab.
3. The **Attributes** panel is expanded by default. If it is not expanded, click the arrow to expand it.
4. All the attributes that are defined in the schema are listed in the left hand pane.
5. Select an attribute to display its properties in the right hand pane.

### 27.6.6 Search for Attributes

You can search for a specific attribute types by using ODSM, as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager".
2. Select the **Schema** tab.
3. The **Attributes** panel is expanded by default. If it is not expanded, click the arrow to expand it.
4. All the attributes that are defined in the schema are listed in the left hand pane.
5. Enter part or all of the attribute name in the **Search** field and click the **Go** icon.

   The search field supports pattern matching. For example, enter `*uid` to find all attributes that end with the string `uid`.
6. Select an attribute to display its properties in the right hand pane.

### 27.6.7 View the Indexing Details of an Attribute

Indexes are configured per server and index configuration is not replicated. A local database index is used to find entries that match search criteria. A VLV index is used to process searches efficiently with VLV controls. Unindexed searches are denied by default, unless the user has the unindexed-search privilege.

A local database index can be one of the following types:

- **approximate** - Improves the efficiency of searches using approximate search filters.
- **equality** - Improves the efficiency of searches using equality search filters.
- **ordering** - Improves the efficiency of searches using "greater than or equal to" or "less than or equal to" search filters.
- **presence** - Improves the efficiency of searches using presence search filters.
- **substring** - Improves the efficiency of searches using substring search filters.

You can view the indexes that are defined for an attribute by using ODSM, as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager".
2. Select the **Schema** tab.
3. The **Attributes** panel is expanded by default. If it is not expanded, click the arrow to expand it.
4. Select an attribute to display its properties in the right hand pane.
5. Scroll down to the **Indexed** property to view the indexing details for that attribute.

### 27.6.8 Add a New Object Class

You can add a new attribute type to the schema by using ODSM, as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager,".
2. Select the **Schema** tab.
3. Click the **Object classes** panel to expand it.

All existing object classes are displayed on the left pane.

4. Click the **Add** icon.
5. Complete the following information on the **Create new object class** window:
   - **Name.** Enter a unique name for the new object class.
   - **Object ID.** Specify the OID that uniquely identifies the object class in the directory server. Oracle Unified Directory supports the use of non-numeric OIDs for easy identification as long as the schema is used internally within the organization. However, for this release ODSM supports numeric OIDs only.
   - **Description.** Enter a human-readable description of the object class.
   - **Type.** Specify the type of object class. Possible values are as follows:
     - **Structural.** A structural object class defines the core type for any entry that contains it. An entry must have exactly one structural class (although that structural class can inherit from other structural or abstract classes).
     - **Auxiliary.** An auxiliary object class does not define the core type of an entry, but defines additional characteristics of that entry. An entry can contain zero or more auxiliary object classes. The set of auxiliary classes that are allowed for use in an entry can be controlled by a DIT content rule that is associated with that entry's structural object class.
     - **Abstract.** An abstract object class cannot be used directly in an entry but must be subclassed by either a structural object class or an auxiliary object class. The subclasses will inherit any required and/or optional attribute type defined by the abstract class.
   - **Superclass.** Click the **Add** icon to specify one or more superior object classes. The new object class will inherit elements from its superior object classes.
   - **Mandatory Attributes.** Click the **Add** icon to specify the set of attribute types that are required to be present (that is, have at least one value) in entries with that object class.
   - **Optional Attributes.** Click the **Add** icon to specify the set of attribute types that are allowed but not required to be present in entries with that object class.
   - **Inherited Attributes.** After the object class has been created, this field indicates the attributes that are inherited from the superior object classes of this object class.
   - **Origin.** Enter the source of this new object class, for example, RFC 4512.

     To view the source of all the schema elements in the directory, select **Show All** from the **View** menu.
   - **Schema File Extension.** If the definition of the new object class is contained in a file, enter the path to the file.
6. Click **Create** to create the new object class.

## 27.6.9 Add an Object Class Based on an Existing Object Class

You can add an object class that is based on an existing object class by using ODSM, as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager".
2. Select the **Schema** tab.

3. Expand the **Object classes** panel.
4. Select the object class on which you want to base the new object class.
5. Click the **Create like** icon.
6. Certain fields are completed by default, based on the object class that you selected. The existing object class is used as the superior object class for the new object class

   Complete the remaining fields for the new object class.

   For information about the fields and their values, see Section 27.6.8, "Add a New Object Class".
7. Click **Create** to create the new object class.

## 27.6.10 View the Properties of an Object Class

You can view the properties of an existing object class by using ODSM, as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager".
2. Select the **Schema** tab.
3. Expand the **Object Classes** panel.
4. All the object classes that are defined in the schema are listed in the left hand pane.
5. Select an object class to display its properties in the right hand pane.

## 27.6.11 Modify an Object Class

You can modify an existing object class by using ODSM, as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager".
2. Select the **Schema** tab.
3. Expand the **Object Classes** panel.
4. Select the object class that you want to modify.
5. Modify the required fields, on the right hand pane.

   For information about the fields, see Section 27.6.8, "Add a New Object Class".
6. Click **Apply** to save your changes.

## 27.6.12 Delete an Object Class

You can delete an existing object class by using ODSM, as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager".
2. Select the **Schema** tab.
3. Expand the **Object Classes** panel.
4. Select the object class that you want to delete.
5. Click the **Delete** icon and click **OK** to confirm the deletion.
6. Click **Apply** to save your changes.

7. Click the **Refresh** icon to refresh the list of attributes on the left hand pane and confirm that the object class has been deleted from the schema.

---

**Note:** The server will return an error if you attempt to delete an object class that is already referenced by one or more entries in the server.

---

## 27.6.13 Search for Object Classes

You can search for a specific object class by using ODSM, as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager".
2. Select the **Schema** tab.
3. Expand the **Object Classes** panel.
4. All the object classes that are defined in the schema are listed in the left hand pane.
5. Enter part or all of the object class name in the **Search** field and click the **Go** icon.

   The search field supports pattern matching. For example, enter `*person` to find all object classes that end with the string `person`.
6. Select an object class to display its properties in the right hand pane.

## 27.6.14 Display a List of LDAP Syntaxes

LDAP syntaxes are essentially data type definitions. The syntax for an attribute type indicates the type of data that should be held by the corresponding values. Syntaxes can be used to determine whether a particular value is acceptable for a given attribute, and to provide information about how the directory server should interact with existing values.

Oracle Unified Directory supports the ability to reject values that violate the associated attribute syntax, and this is the default behavior for the purposes of standards compliance. It is possible to disable attribute syntax checking completely if necessary. It is also possible to accept values that violate the associated syntax but log a warning message to the directory server's error log when this occurs. For information about disabling schema checking, see Section 27.2, "Configuring Schema Checking".

You cannot modify the LDAP syntaxes but you can view all existing LDAP syntaxes by using ODSM, as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager".
2. Select the **Schema** tab.
3. Expand the **Syntaxes** panel.
4. All the supported LDAP syntaxes are listed in the left hand pane.
5. Select a syntax to display its properties in the right hand pane.

   The information that is displayed includes all of the attributes and matching rules that currently refer to that syntax.

## 27.6.15 Search for a Syntax

You can search for a specific LDAP syntax by using ODSM, as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager".
2. Select the **Schema** tab.
3. Expand the **Syntaxes** panel.
4. All the supported LDAP syntaxes are listed in the left hand pane.
5. Enter part or all of the syntax name in the **Search** field and click the **Go** icon.

   The search field supports pattern matching. For example, enter `*time` to find all syntaxes that end with the string `time`.
6. Select a syntax to display its properties in the right hand pane.

## 27.6.16 Display a List of LDAP Matching Rules

Matching rules are used by the directory server to compare two values for the same attribute, that is, to perform matching operations on them. There are several different types of matching rules, including the following:

- **Equality matching rules.** These matching rules are used to determine whether two values are logically equal to each other. Different implementations of equality matching rules can use different criteria for making this determination (for example, whether to ignore differences in capitalization or deciding which spaces are significant).
- **Ordering matching rules.** These matching rules are used to determine the relative order for two values, for example, when evaluating greater-or-equal or less-or-equal searches, or when the results need to be sorted.
- **Substring matching rules.** These matching rules are used to determine whether a given substring assertion matches a particular value.
- **Approximate matching rules.** These matching rules are used to determine whether two values are approximately equal to each other. This is frequently based on "sounds like" or some other kind of fuzzy algorithm. Approximate matching rules are not part of the official LDAP specification, but they are included in Oracle Unified Directory for added flexibility.

You cannot modify the matching rules but you can view all existing matching rules by using ODSM, as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager".
2. Select the **Schema** tab.
3. Expand the **Matching Rules** panel.
4. All the configured matching rules are listed in the left hand pane.
5. Select a matching rule to display its properties in the right hand pane.

   The information that is displayed includes all of the attributes and matching rules that currently refer to that matching rule.

## 27.6.17 Search for a Matching Rule

You can search for a specific matching rule by using ODSM, as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager".

2. Select the **Schema** tab.
3. Expand the **Matching Rules** panel.
4. All the configured matching rules are listed in the left hand pane.
5. Enter part or all of the matching rule name in the **Search** field and click the **Go** icon.

   The search field supports pattern matching. For example, enter `*match` to find all matching rules that end with the string `match`.
6. Select a matching rule to display its properties in the right hand pane.

### 27.6.18 Display a List of Content Rules

Content rules provide a mechanism for defining the content that can appear in an entry. At most one content rule may be associated with an entry, based on its structural object class. If such a rule exists for an entry, it will work in conjunction with the object classes contained in that entry to define which attribute types must, may, and must not be present in the entry, as well as which auxiliary classes the entry may include.

You can view all the content rules that are configure in the server by using ODSM, as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager".
2. Select the **Schema** tab.
3. Expand the **Content Rules** panel.
4. All the configured content rules are listed in the left hand pane.
5. Select a content rule to display its properties in the right hand pane.

### 27.6.19 Search for a Content Rule

You can search for a specific content rule by using ODSM, as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager".
2. Select the **Schema** tab.
3. Expand the **Content Rules** panel.
4. All the configured content rules are listed in the left hand pane.
5. Enter part or all of the content rule name in the **Search** field and click the **Go** icon.
6. Select a content rule to display its properties in the right hand pane.

### 27.6.20 Create a New Content Rule

You can add a new content rules to the schema by using ODSM, as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager".
2. Select the **Schema** tab.
3. Expand the **Content Rules** panel.
4. Click the **Add** icon.

5. Complete the following information on the **Create new content rule** window:
   - **Name.** Enter a unique name for the new content rule.
   - **Structural Object Class.** Specify the name of the structural object class with which this content rule is associated.
   - **Description.** Enter a human-readable description of the content rule.
   - **Auxiliary Object Classes.** Click the **Add** icon to specify the list of auxiliary object classes that may be present in entries with the associated structural class. If no values are provided, such entries will not be allowed to have any auxiliary object classes. You can specify the allowed auxiliary object classes by using their names or OIDs.
   - **Mandatory Attributes.** Click the **Add** icon to specify the list of attribute types that are required to be present in entries with the associated structural class. This list is in addition to the attribute types that are required by the object classes included in the entry. These additional attribute types do not need to be allowed by any of those object classes. You can specify the mandatory attributes by using their names or OIDs.
   - **Optional Attributes.** Click the **Add** icon to specify the list of attribute types that are allowed, but not required, to be present in entries with the associated structural class. This list is in addition to the attribute types that are allowed by the object classes included in the entry. You can specify the optional attributes by using their names or OIDs.
   - **Disallowed Attributes.** Click the **Add** icon to specify the list of attribute types that are prohibited from being present in entries with the associated structural class. This list may not include any attribute types that are required by the structural class or any of the allowed auxiliary classes. The list can be used to prevent the inclusion of attribute types which would otherwise be allowed by one of those object classes. You can specify the disallowed attributes by using their names or OIDs.
   - **Origin.** Enter the source of this new content rule, for example, RFC 4517.

     To view the source of all the schema elements in the directory, select **Show All** from the **View** menu.
   - **Schema File Extension.** If the content rule's definition is contained in a file, enter the path to the file.
6. Click **Create** to create the new content rule.

## 27.6.21 Create a Content Rule Based on an Existing Content Rule

You can add a content rule that is based on an existing content rule by using ODSM, as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager".
2. Select the **Schema** tab.
3. Expand the **Content Rules** panel.
4. Select the content rule on which you want to base the new content rule.
5. Click the **Create like** icon.
6. Certain fields are completed by default, based on the content rule that you selected.

Complete the remaining fields for the new content rule.

For information about the fields and their values, see Section 27.6.20, "Create a New Content Rule".

7. Click **Create** to create the new content rule.

## 27.6.22 Modify a Content Rule

You can modify an existing content rule by using ODSM, as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager".
2. Select the **Schema** tab.
3. Expand the **Content Rules** panel.
4. Select the content rule that you want to modify.
5. Modify the required fields, on the right hand pane.

   For information about the fields, see Section 27.6.20, "Create a New Content Rule".
6. Click **Apply** to save your changes.

## 27.6.23 Delete a Content Rule

You can delete an existing content rule by using ODSM, as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager".
2. Select the **Schema** tab.
3. Expand the **Content Rules** panel.
4. Select the content rule that you want to delete.
5. Click the **Delete** icon and click **OK** to confirm the deletion.
6. Click **Apply** to save your changes.
7. Click the **Refresh** icon to refresh the list of content rules on the left hand pane and confirm that the content rule has been deleted from the schema.

# 28

# Moving From a Test to a Production Environment

This chapter describes how to move, or *clone*, an Oracle Unified Directory installation between environments, specifically, between a test environment and a production environment. Moving between environments enables you to develop and test applications in a test environment, and then roll out the test applications and, optionally, test data to your production environment. In the remainder of this chapter, the test environment is referred to as the source environment and the production environment as the target environment.

This chapter includes the following topics:

- Section 28.1, "Introduction to Moving Across Environments"
- Section 28.2, "Limitations in Moving From Test to Production"
- Section 28.3, "Overview of the Test to Production Process"

Note that the Oracle Unified Directory "test to production" plug-in offers a subset of the functionality that is provided by the Oracle Fusion Middleware "test to production" framework. The documentation in this chapter is specific to Oracle Unified Directory. For a comprehensive description of moving other Fusion Middleware components between environments, see "Moving from a Test to a Production Environment" section in the Oracle Fusion Middleware Administrator's Guide.

## 28.1 Introduction to Moving Across Environments

Moving an Oracle Unified Directory installation minimizes the amount of work that would otherwise be required to reapply all the customization and configuration changes made in one environment to another. You can install, configure, customize, and validate Oracle Unified Directory in a test environment. Once the system is stable and performs as required, you can create the production environment by moving a copy of the server and its configuration from the test environment, instead of redoing all the changes that were incorporated into the test environment.

If you have an existing production environment, you can move any modifications of the test environment, such as customization, to the production environment.

Moving an Oracle Unified Directory installation from a test to a production environment assumes that the production environment is on the same operating system as the test environment. In addition, the operating system architecture must be the same in both environments. For example, both environment must be running 32-bit operating systems or 64-bit operating systems.

## 28.2 Limitations in Moving From Test to Production

Moving an Oracle Unified Directory installation between environments is supported with the following restrictions:

- Moving from a test to a production environment is supported for directory server instances only. You cannot move a proxy server instance or a replication gateway server instance between environments.
- You cannot move a replicated topology. To move an entire replicated topology, you must first move each server instance in the topology, then configure replication manually between the server instances. If you move a server that is part of a replicated topology, the replication configuration is removed from the configuration in the destination environment.
- Security data is not moved during the test to production process. This includes the following elements:
  - the SSL configuration (keystore, truststore, and other security configuration located in the `config` directory by default)
  - the SNMP V3 security file (located in the `config/snmp` directory by default)

## 28.3 Overview of the Test to Production Process

The move from a test to a production environment involves three broad steps:

1. Moving the Oracle Unified Directory binaries to the production system.
2. Moving the Oracle Unified Directory configuration to the production system.
3. Moving the data to the production system.

These procedures assume that you are moving an Oracle Unified Directory test system to a new production deployment (and do not have an existing production system).

### 28.3.1 Moving the Binaries

To obtain a copy of the Oracle Unified Directory binaries on the new production system, install the binaries as described in "Installing Oracle Unified Directory" in the *Installation Guide for Oracle Unified Directory*.

### 28.3.2 Moving the Configuration

Moving the configuration between environments, involves three steps:

1. Copying the configuration from the source environment.
2. Editing the configuration, if required.
3. Pasting the configuration in the target environment.

#### 28.3.2.1 Copying the Configuration

To obtain a copy of an existing configuration, run the `oudCopyConfig` command in the source environment.

On UNIX systems, run the command as follows:

```
$ OUD_ORACLE_HOME/bin/oudCopyConfig -javaHome java_home \
  -sourceInstanceHomeLoc instance_dir -archiveLoc archive_location \
  -logDirLoc log_directory
```

For example:

```
$ OUD_ORACLE_HOME/bin/oudCopyConfig -javaHome /usr/jdk \
  -sourceInstanceHomeLoc /local/asinst_1 -archiveLoc /tmp/oud.jar \
  -logDirLoc /tmp/logs
```

On Windows systems, run the command as follows:

```
$ OUD_ORACLE_HOME\bat\oudCopyConfig.bat -javaHome java_home \
  -sourceInstanceHomeLoc instance_dir -archiveLoc archive_location \
  -logDirLoc log_directory
```

For a complete synopsis of the `oudCopyConfig` command, see Section A.2.9, "oudCopyConfig".

The `oudCopyConfig` command performs the following actions:

- creates an archive (*archive_location*) that contains the required configuration data to move the test instance (*instance_dir*) to a production environment. `-archiveLoc` specifies the full path to the archive.
- creates a move plan in the archive.
- logs any messages to *log_directory*. If not specified, the default location of logged messages is the system temporary directory.

#### 28.3.2.2 Editing the Configuration

You can modify certain configuration parameters by editing the *move plan*. A move plan is an XML file that exposes customizable parameters during the move across environments.

The move plan is generated when you run the `oudCopyConfig` command and is used by the `oudPasteConfig` command to duplicate the configuration.

After you have copied the configuration, edit the configuration as follows:

1. Run the `oudExtractMovePlan` command to obtain a copy of the configuration. On UNIX systems, run the command as follows:

   ```
   $ OUD_ORACLE_HOME/bin/oudExtractMovePlan -javaHome java_home \
     -archiveLoc archive_location -planDirLoc moveplan_dir \
     -logDirLoc log_directory
   ```

   For example:

   ```
   $ OUD_ORACLE_HOME/bin/ExtractMovePlan -javaHome /usr/jdk \
     -archiveLoc /tmp/oud.jar -planDirLoc /tmp \
     -logDirLoc /tmp/logs
   ```

   On Windows systems, run the command as follows:

   ```
   $ OUD_ORACLE_HOME\bat\oudExtractMovePlan.bat -javaHome java_home \
     -archiveLoc archive_location -planDirLoc moveplan_dir \
     -logDirLoc log_directory
   ```

   For a complete synopsis of the `oudextractMovePlan` command, see Section A.2.10, "oudExtractMovePlan".

   The `oudExtractMovePlan` command creates an editable version of the configuration in a file named `moveplan.xml`, in the location specified by the `-planDirLoc` argument. This directory must exist, and be writable.

2. In a text editor, edit the `moveplan.xml` file, as required.

The following parameters can be configured in the move plan:

- OUD non SSL port
- OUD SSL port
- OUD admin connector port
- SNMP listen port
- SNMP trap port
- JMX port
- OUD root user password file
- SMTP server and port
- Absolute paths to files or directories, including the following:
  - Backup directory
  - Database directory
  - Profile directory
  - Dictionary file
  - Referential integrity plug-in log file
  - SMTP account status notification handler message template file

3. Save the `moveplan.xml` file.

#### 28.3.2.3 Pasting the Configuration

When you have edited the move plan, paste the configuration into the target environment as follows:

1. Move the archive and move plan to the target host.

   In most scenarios, the test environment and the production environment are on separate machines. You must therefore move or copy the archive and move plan to the target machine.

   If your test and production environments are on the same machine, this step is unnecessary.

2. Paste the configuration in the target environment, by running the `oudPasteConfig` command on the target environment.

   On UNIX systems, run the command as follows:

```
$ OUD_ORACLE_HOME/bin/oudPasteConfig -javaHome java_home \
  -targetInstanceHomeLoc instance_dir -archiveLoc archive_location \
  -targetOracleHomeLoc ORACLE_HOME -movePlanLoc move_plan_location \
  -logDirLoc log_directory -targetInstanceName instance_name
```

   For example:

```
$ OUD_ORACLE_HOME/bin/oudPasteConfig -javaHome /usr/jdk \
  -targetInstanceHomeLoc /local/asinst_2 -archiveLoc /tmp/oud.jar \
  -targetOracleHomeLoc /local/ORACLE_HOME -movePlanLoc /tmp/moveplan.xml \
  -logDirLoc /tmp/logs -targetInstanceName asinst_2
```

   On Windows systems, run the command as follows:

```
$ OUD_ORACLE_HOME\bat\oudPasteConfig.bat -javaHome java_home \
  -targetInstanceHomeLoc instance_dir -archiveLoc archive_location \
```

```
-targetOracleHomeLoc ORACLE_HOME -movePlanLoc move_plan_location \
-logDirLoc log_directory -targetInstanceName instance_name
```

For a complete synopsis of the `oudPasteConfig` command, see Section A.2.11, "oudPasteConfig".

The `oudPasteConfig` command creates a new server instance with the configuration obtained from the archive and the amended move plan, if any.

### 28.3.3 Moving the Data

The simplest way to move data from a test system to a production is to export the data from the test system, and import it to the production system.

For information about how to do this, see Section 17.1, "Importing and Exporting Data".

# Part VI

# Advanced Administration: Monitoring and Tuning Performance

This part describes how to monitor Oracle Unified Directory server instances and how to tune server performance.

This part includes the following chapters:

- Chapter 29, "Monitoring Oracle Unified Directory"
- Chapter 30, "Tuning Performance"

# 29

# Monitoring Oracle Unified Directory

Oracle Unified Directory provides an extensible monitoring framework. This chapter provides an overview of the monitoring functionality, and describes how to configure monitoring. When the monitoring framework has been configured, you can view the statistics on a server instance, or replicated topology.

This chapter covers the following topics:



## 29.1 Monitoring Overview

Monitoring information and performance data can be found in:

- logs

  For information about configuring logs, see Section 29.3, "Configuring Logs".

- alerts

  For information about configuring alerts, see Section 29.4, "Configuring Alerts and Account Status Notification Handlers".

- `cn=monitor`

  For information about `cn=monitor`, see Section 29.5, "Monitoring the Server With LDAP".

- DIRECTORY_SERVER_MIB, defined by RFC 2605

  For information about monitoring the server with SNMP, see Section 29.6, "Monitoring the Server With SNMP".

To access the monitoring information, ensure that you have the required protocol:

- For logs you need a file system.
- For alerts you need JMX:RMI or SMTP.

- For `cn=monitor` you need LDAP or JMX:RMI (for example jconsole).
- For DIRECTORY_SERVER_MIB you need SNMP.

## 29.2 Configuring Monitor Providers

Monitor providers are enabled by default and provide information about the server that can be useful for monitoring or troubleshooting purposes. The `cn=monitor` entry contains the monitoring information that is published by the monitor providers. When the monitor provider is disabled, the provided information is no longer available under `cn=monitor`.

Monitor providers can be configured by using the `dsconfig` command. For more information, see Section 14.1, "Managing the Server Configuration With `dsconfig`".

### 29.2.1 To View Monitor Providers

Run the `dsconfig` command with the `list-monitor-providers` subcommand, as follows:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  list-monitor-providers

Monitor Provider   : Type              : enabled
-------------------:-------------------:--------
Client Connections : client-connection : true
Entry Caches       : entry-cache       : true
JVM Memory Usage   : memory-usage      : true
JVM Stack Trace    : stack-trace       : true
System Info        : system-info       : true
Version            : version           : true
```

### 29.2.2 To Disable a Monitor Provider

Run the `dsconfig` command with `set-monitor-provider-prop` as follows:

For example, to set the JVM Stack Trace monitor provider to false, use the following command:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  set-monitor-provider-prop --provider-name "JVM Stack Trace" \
  --set enabled:false
```

Running the `dsconfig` command with the `list-monitor-providers` subcommand now shows the JVM Stack Trace monitor provider as false:

```
Monitor Provider   : Type              : enabled
-------------------:-------------------:--------
Client Connections : client-connection : true
Entry Caches       : entry-cache       : true
JVM Memory Usage   : memory-usage      : true
JVM Stack Trace    : stack-trace       : false
System Info        : system-info       : true
Version            : version           : true
```

## 29.3 Configuring Logs

Oracle Unified Directory provides several types of logs: access logs, audit logs, error logs, debug logs, and a replication repair log. The replication repair log is read-only and its use is restricted to enabling replication conflict resolution. The following

sections describe how to configure access, audit, error, and debug logs by using the `dsconfig` command-line interface or Oracle Directory Services Manager. In addition, the section describes how to log admin operations.

For a breakdown of the result codes found in the logs, see Section D.17.11, "result code."

This section contains the following topics:

- Section 29.3.1, "Configuring Logs by Using dsconfig"
- Section 29.3.2, "Configuring Logs by Using ODSM"
- Section 29.3.3, "Logging Operations to Access Log Publishers"

## 29.3.1 Configuring Logs by Using dsconfig

The easiest way to configure logging with `dsconfig` is to use the command in interactive mode, which walks you through the configuration. This section provides the required commands in non-interactive mode, so that you can see the specific parameters that are set. For more information about `dsconfig`, see Section 14.1, "Managing the Server Configuration With `dsconfig`".

Log configuration includes the definition of three configuration objects:

- **Log publisher.** A log publisher is defined for each logger. The log publisher type corresponds to the type of log. For more information about log publishers, see Section 29.3.1.1, "Configuring Log Publishers".
- **Log retention policy.** The retention policy determines how long archived log files are stored. For more information about log retention policies, see Section 29.3.1.2, "Configuring Log Retention Policies".
- **Log rotation policy.** The rotation policy determines how often log files are rotated. For more information on log rotation policies, see Section 29.3.1.3, "Configuring Log Rotation Policies".

### 29.3.1.1 Configuring Log Publishers

Oracle Unified Directory provides several log publishers by default.

Any number of log publishers of any type can be defined and active at any time. This means that you can log to different locations or different types of repositories and that you can specify various sets of criteria for what to include in the logs.

For more information about the configuration properties associated with log publishers, see the Oracle Unified Directory Configuration Reference.

This section covers the following topics:

- Section 29.3.1.1.1, "To List Existing Log Publishers"
- Section 29.3.1.1.2, "To Enable a Log Publisher"
- Section 29.3.1.1.3, "Logging in ODL Format"
- Section 29.3.1.1.4, "Logging Internal Operations"
- Section 29.3.1.1.5, "Configuring the Name of Rotated Log Files Using Local Time Stamp"

#### 29.3.1.1.1 To List Existing Log Publishers

1. To view the existing log publishers run the following `dsconfig` command:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  list-log-publishers
```

The default output will be similar to the following:

```
Log Publisher                  : Type              : enabled
-------------------------------:-------------------:--------
File-Based Access Logger       : file-based-access : true
File-Based Admin Access Logger : file-based-access : true
File-Based Audit Logger        : file-based-access : false
File-Based Debug Logger        : file-based-debug  : false
File-Based Error Logger        : file-based-error  : true
Oracle Access Logger           : file-based-access : false
Oracle Error Logger            : file-based-error  : false
Replication Repair Logger      : file-based-error  : true
```

2. To display the properties of a log publisher run the following `dsconfig` command:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  get-log-publisher-prop --publisher-name "File-Based Error Logger"
```

#### 29.3.1.1.2 To Enable a Log Publisher

Not all of the log publishers are enabled by default. If a log publisher is disabled, messages of that type are not logged.

To enable a log publisher, set its enabled property to true. For example, to enable the audit logger, run the following command:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  set-log-publisher-prop --publisher-name "File-Based Audit Logger" \
  --set enabled:true
```

When a log publisher is enabled, the server immediately starts logging messages to the appropriate publisher. You do not need to restart the server for this change to take effect.

#### 29.3.1.1.3 Logging in ODL Format

Oracle Unified Directory also writes diagnostic log files in the Oracle Diagnostic Logging (ODL) format.

ODL is disabled by default. To enable ODL, set the `enabled` property of the ODL Access Log publisher or the ODL Error Log publisher to `true`. The following example enables the access logger:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  set-log-publisher-prop --publisher-name "Oracle Access Logger" \
  --set enabled:true
```

To enable the error logger, use `--publisher-name "Oracle Error Logger"`.

ODL access logs are stored in the following file:

`instance_dir/OUD/logs/access.log`

ODL error logs are stored in the following directory:

`instance_dir/OUD/logs/errors.log`

Note that the standard access and error loggers are not disabled when you enable the ODL loggers. You should therefore disable the standard access and error logs after

you enable the ODL loggers, unless you specifically want to maintain logs in both formats.

For more information about ODL, including an explanation of the log file format, see "Managing Log Files and Diagnostic Data" in the Oracle Fusion Middleware Administrator's Guide.

#### 29.3.1.1.4 Logging Internal Operations

By default, the `suppress-internal-logging` property for log publishers is set to `true`. If you need to log internal operations (such as operations performed by the LDIF connection handler and certain plug-ins), set `suppress-internal-logging` to `false`. The following example sets `suppress-internal-logging` to `false` for the file-based access logger:

```
dsconfig set-log-publisher-prop \
--publisher-name File-Based\ Access\ Logger \
--add operations-to-log:internal \
--hostname localhost \
--port 4444 \
-X \
--bindDN cn=directory\ manager \
--bindPasswordFile /tmp/password \
--no-prompt
```

#### 29.3.1.1.5 Configuring the Name of Rotated Log Files Using Local Time Stamp

By default, Oracle Unified Directory automatically renames (rotates) its local server log file using date stamp in GMT format.

You can change these default settings for log file rotation. You can configure a server instance to include a local time stamp in the file name of rotated log files.

To configure the log file names using local time stamp, you must set the `log-file-use-local-time` property of the appropriate log publisher to `true`. The following example describes how to set up the local time stamp in the the file name of access rotated log files:

```
dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
set-log-publisher-prop --publisher-name "Oracle Access Logger" \
--set log-file-use-local-time:true
```

> **Note:** The rotated log file name using local time stamp follows the format used by Oracle Directory Server Enterprise Edition to ensure compatibility.

### 29.3.1.2 Configuring Log Retention Policies

Log retention policies dictate size and space limits for log files. Oracle Unified Directory provides the following three log retention policies:

- File count retention (`file-count`). By default, this policy sets the maximum number of log files to 10, for a specified type of log file.
- Free disk space retention (`free-disk-space`). By default, this policy sets a minimum remaining free disk space limit to 500 Mb, for a specified type of log file.
- Size limit retention (`size-limit`). By default, this policy sets the disk spaced used to a maximum of 500 Mb, for a specified type of log file.

By default, the log retention policy that is enabled is File count retention.

You can also create your own custom log retention policies.

#### 29.3.1.2.1 To View the Log Retention Policies

To view a list of the existing log retention policies run the following `dsconfig` command:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  list-log-retention-policies
```

The default output will be similar to the following:

```
Log Retention Policy             : Type            : disk-space-used : free-disk-space : number-of-files
---------------------------------:-----------------:-----------------:-----------------:---------------
File Count Retention Policy      : file-count      : -               : -               : 10
Free Disk Space Retention Policy : free-disk-space : -               : 500 mb          : -
Size Limit Retention Policy      : size-limit      : 500 mb          : -               : -
```

To list the log retention policy properties run the following `dsconfig` command

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -X -n \
  get-log-retention-policy-prop --policy-name "Free Disk Space Retention Policy"
```

#### 29.3.1.2.2 To Create a Log Retention Policy

To create a log retention policy, and to set it as enabled, type:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -w pwd-file -X -n
  create-log-retention-policy --policy-name MyMaxDiskSpace \
  --type size-limit --set disk-space-used:100mb
```

#### 29.3.1.2.3 To Modify a Log Retention Policy

To modify the properties of an existing log retention policy run the following `dsconfig` command:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -w pwd-file -X -n \
  set-log-retention-policy-prop --policy-name "File Count Retention Policy" \
  --set number-of-files:20
```

Instead of setting a property value, you can add, reset or remove a property value, using the `--add`, `--reset`, or `--remove` subcommands instead of the `--set` subcommand. For details, see Section A.2.4, "dsconfig".

### 29.3.1.3 Configuring Log Rotation Policies

Log rotation policies dictate how often the files are rotated, that is to say, how long log files are kept based on various criteria. Oracle Unified Directory provides the following four log rotation policies:

- 24 Hours time limit rotation policy. By default, this policy sets the rotation interval to one day. Time of day can be configured.
- 7 Days time limit rotation policy. By default, this policy sets the rotation interval to one week. Time of day can be configured.
- Fixed time limit rotation policy. By default, this policy sets the time of day that log files are to be rotated, to one minute before midnight.
- Size time limit rotation policy. By default, this policy sets a maximum size that log files can reach to 100 Mb, before the log file is rotated.

The type of log rotation policy enabled by default depends on the log type.

- For access and audit logs, the following are enabled:
  - 24 Hours time limit rotation policy
  - Size time limit rotation policy
- For error and replication repair logs, the following are enabled:
  - 7 Days time limit rotation policy
  - Size time limit rotation policy

You can create your own custom log rotation policies.

> **Note:** When multiple rotation policies are specified for the same log, the first threshold that is reached triggers the rotation.

#### 29.3.1.3.1 To View the Log Rotation Policies

To view a list of the existing log rotation policies run the following `dsconfig` command:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -w pwd-file -X -n \
  list-log-rotation-policies
```

The default output will be similar to the following:

```
Log Rotation Policy                : Type       : file-size-limit : rotation-interval : time-of-day
-----------------------------------:------------:-----------------:-------------------:------------
24 Hours Time Limit Rotation Policy : time-limit : -               : 1 d               : -
7 Days Time Limit Rotation Policy   : time-limit : -               : 1 w               : -
Fixed Time Rotation Policy          : fixed-time : -               : -                 : 2359
Size Limit Rotation Policy          : size-limit : 100 mb          : -                 : -
```

To display the log rotation policy properties, run the following command:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -w pwd-file -X -n \
  get-log-rotation-policy-prop "Fixed Time Rotation Policy"
```

#### 29.3.1.3.2 To Create a Log Rotation Policy

To create a log rotation policy run the following `dsconfig` command:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -w pwd-file -X -n \
  create-log-rotation-policy --policy-name my2DayPolicy \
  --type time-limit --set rotation-interval:2d
```

The policy type can be one of the following:

- `size-limit`
- `fixed-time`
- `time-limit`

#### 29.3.1.3.3 To Set Log Rotation or Retention for a Specific Log File

To set a rotation or retention policy on a specific log file, you must create a log publisher and set the log rotation or log retention policy.

To set log rotation or retention for a specific log file run the following `dsconfig` command:

```
$ dsconfig -h localhost -p 1444 -D "cn=Directoy manager" -j pwd-file -n -X \
  create-log-publisher --publisher-name myPublisher \
```

```
--type file-based-access --set log-file:logs/myLogs --set enabled:true \
--set retention-policy:MyMaxDiskSpace --set rotation-policy:my2DayPolicy
```

## 29.3.2 Configuring Logs by Using ODSM

This section describes how to use ODSM to configure logs. It contains the following topics:

- Section 29.3.2.1, "Modify Logger Properties"
- Section 29.3.2.2, "Modify Log Rotation Policies"
- Section 29.3.2.3, "Modify Log Retention Policies"

### 29.3.2.1 Modify Logger Properties

Oracle Unified Directory provides several log publishers, or loggers, by default. Any number of loggers of any type can be defined and active at any time. This means that you can log to different locations or different types of repositories and that you can specify various sets of criteria for what to include in the logs.

You cannot create a new log publisher with ODSM, but you can modify the properties of an existing log publisher.

To configure logger properties by using ODSM, complete the following steps:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager."
2. Select the **Configuration** tab.
3. Expand the **General Configuration** element.
4. Expand the **Logging** element.
5. Expand the **Loggers** element and click on the logger whose properties you want to modify.

   The properties of the logger are displayed in the right hand pane. The configurable properties will depend on the type of logger that you have selected. For a comprehensive list of all configurable properties and their allowed values, see the *Oracle Unified Directory Configuration Reference*.

   Oracle Unified Directory provides the following general configuration policies depending on the type of logger you have selected:

   - **Enabled.** It indicates whether the Log Publisher is enabled for use.
   - **Log Publisher File Location.** It specifies the file name to use for the log files generated by the File-Based Access Log Publisher. The path to the file is relative to the server root.
   - **Log Publisher Permissions.** It indicates the UNIX permissions of the log files created by this File-Based Access Log Publisher.
   - **Operations to Log.** It indicates which operations must be logged.

     This property is only available for the access and audit log publishers.
   - **Log Request and Response Controls.** It indicates whether the request controls and response controls should be logged along with the operations that are requested by the client applications.

     This property is only available for the access and audit log publishers.
   - **Default Severity.** It specifies the default severity levels for the logger.

This property is only available for the error log publishers.

- **Default Debug Level.** It specifies the lowest severity level of debug messages to log when none of the defined targets match the message.

  This property is only available for the debug log publishers.

For a comprehensive list of all configurable properties and their allowed values for each logger, see the *Oracle Unified Directory Configuration Reference*.

---

**Note:** You can configure the log rotation and log retention policies for the logger that you select in Step 5. For more information about configuring log rotation and log retention policies, see Section 29.3.2.2, "Modify Log Rotation Policies" and Section 29.3.2.3, "Modify Log Retention Policies."

---

### 29.3.2.2 Modify Log Rotation Policies

Log rotation policies dictate how often log files are rotated, that is to say, how long log files are kept based on various criteria.

Oracle Unified Directory provides the following four log rotation policies:

- 24 Hours time limit rotation policy. By default, this policy sets the rotation interval to one day. Time of day can be configured.
- 7 Days time limit rotation policy. By default, this policy sets the rotation interval to one week. Time of day can be configured.
- Fixed time limit rotation policy. By default, this policy sets the time of day that log files are to be rotated, to one minute before midnight.
- Size time limit rotation policy. By default, this policy sets a maximum size that log files can reach to 100 Mb, before the log file is rotated.

The type of log rotation policy that is enabled by default depends on the logger type.

You can configure log rotation policies by using ODSM, as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager."
2. Select the **Configuration** tab.
3. Expand the **General Configuration** element.
4. Expand the **Logging** element.
5. Select the **Rotation Policies** element and modify the required properties.

You can also add a new rotation policy or delete an existing rotation policy by clicking the Add or Delete icons on this page, and completing the required information.

### 29.3.2.3 Modify Log Retention Policies

Log retention policies dictate size and space limits for log files. Oracle Unified Directory provides the following three log retention policies by default:

- File count retention (file-count). By default, this policy sets the maximum number of log files to 10, for a specified type of log file.
- Free disk space retention (free-disk-space). By default, this policy sets a minimum remaining free disk space limit to 500 Mb, for a specified type of log file.

- Size limit retention (size-limit). By default, this policy sets the disk spaced used to a maximum of 500 Mb, for a specified type of log file. By default, the log retention policy enabled is File count retention.

You can configure log retention policies by using ODSM, as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager."
2. Select the **Configuration** tab.
3. Expand the **General Configuration** element.
4. Expand the **Logging** element.
5. Select the **Retention Policies** element and modify the required properties.

You can also add a new retention policy or delete an existing retention policy by clicking the Add or Delete icons on this page, and completing the required information.

## 29.3.3 Logging Operations to Access Log Publishers

Oracle Unified Directory provides a new parameter to specify the operations to log. This section describes the this new configuration parameter, and contain the following topics:

- Section 29.3.3.1, "Overview of the Admin Logger"
- Section 29.3.3.2, "Configuring Logged Operations in Access Log Publishers Using ODSM"

### 29.3.3.1 Overview of the Admin Logger

Oracle Unified Directory provides a mechanism for separating admin logs from user logs by means of Admin connector. Administration operations are now logged into a separate file that provides logging information associated with the administration traffic.

---

**Note:** Oracle Unified Directory out-of-the-box supports a dedicated access logger, namely `File-Based Admin Access Logger`, which contains only operations of the administrator connector. Therefore, you don't have to perform any action specific action to log administration operations into a separate file.

---

You can configure the access logs to specify the type of operation to log using `operations-to-log` property. This property is optional, and has the following configurable values:

- `SYNCHRONIZATION`
- `INTERNAL`
- `ADMINISTRATION`
- `USER`
- `ADMIN_BROWSING`
- `ALL`

In that sense, Oracle Unified Directory supports the following operation types:

- Synchronization Operations

  Synchronization operations, such as locks, process synchronization, attribute mapping and transformation.

- Internal Operations

  Internal operations are internal, because they are initiated not by external requests from clients, but instead internally by plug-ins. You must use internal operation calls when the plug-in needs Directory Server to perform an operation for which no client request exists.

- Administration Operations

  Administration operations are performed on the admin network group, excluding operations associated with **network group selection** control.

- User Operations

  User operations are performed on any user network group, excluding operations associated with **network group selection** control.

- Admin Browsing Operations

  Admin browsing operations are associated with the network group selection control. This excludes operations associated with network group dependency.

> **Note:** Operations handled by network group that are created by a user and accessing admin suffixes is considered as User operations.

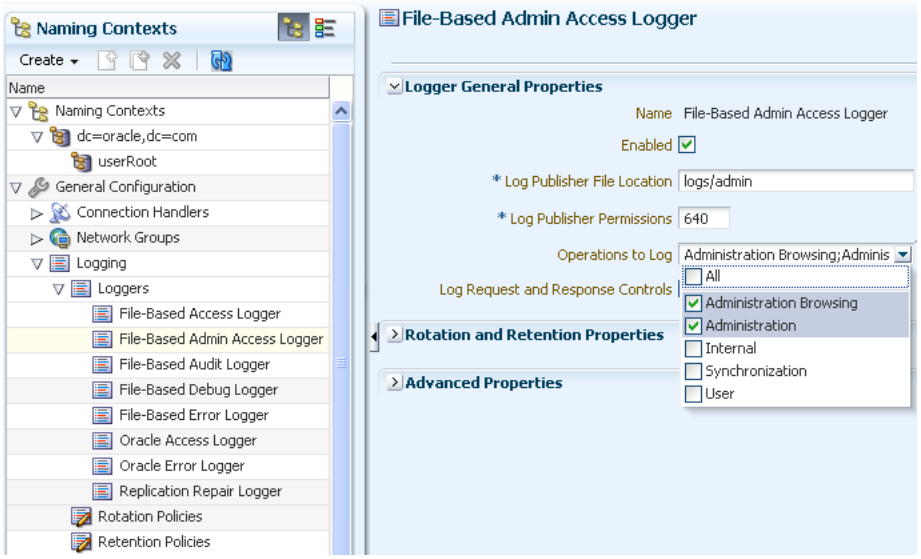
#### 29.3.3.2 Configuring Logged Operations in Access Log Publishers Using ODSM

ODSM groups the log publisher properties into three different headers, namely **Logger General Properties, Rotation and Retention Properties,** and **Advanced Properties** depending on the nature and behavior of the property. The **Logger General Properties** region is visible by default for all loggers and allows you to configure operations to log for file-based access loggers.

You can configure operations to log in Access Log Publishers using ODSM, as follows:

1. Connect to the directory server or directory proxy server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager."
2. Select the **Configuration** tab.
3. Expand the **General Configuration** element.
4. Expand the **Logging** element.
5. Expand the **Loggers** element.
6. Click the file-based access logger that you want to modify, for instance **File-Based Admin Access Logger.**
7. In the **Logger General Properties** region, perform the following step:

   From the **Operation to Log** list, select the operations to log.




8. Click **Apply.**

## 29.4 Configuring Alerts and Account Status Notification Handlers

Oracle Unified Directory provides mechanisms for transmitting alert and account status notifications by means of JMX extensions or SMTP extensions. You can configure the directory server to send alert notifications when an event occurs during processing. Typical server events include server starts and shut downs, or problems that are detected by the server, such as an attempt to write to the configuration file.

You can also receive account status notifications when an event occurs during password policy processing, such as when accounts are locked out, accounts expire, passwords expire, and so on.

Alerts and account status notification handlers are configured by using the `dsconfig` command. For more information, see Section 14.1, "Managing the Server Configuration With `dsconfig`".

For additional information about the topics in this section, see Chapter 24, "Managing Password Policies" and "The Alert Handler Configuration" in the *Oracle Unified Directory Configuration Reference*.

### 29.4.1 Managing Alert Handlers

Oracle Unified Directory provides mechanisms for transmitting alert and account status notifications by means of JMX extensions or SMTP extensions.

You can configure Oracle Unified Directory to send alert notifications when an event occurs during processing. Typical server events include server starts and shut downs, or problems that are detected by the server, such as an attempt to write to the configuration file. You can also receive account status notifications when an event occurs during password policy processing, such as when accounts are locked out, accounts expire, passwords expire, and so on.

Oracle Unified Directory supports the following alert handlers:

- JMX alert handler for JMX notifications
- SMTP alert handler for email notifications.

The following topics describe how to manage the alert handler configuration:

- Section 29.4.1.1, "Managing Alert Handlers by Using `dsconfig`"
- Section 29.4.1.2, "Managing Alert Handlers by Using ODSM"
- Section 29.4.1.3, "Supported Alert Types"

### 29.4.1.1 Managing Alert Handlers by Using `dsconfig`

The following sections describe how to manage the alert handler configuration by using dsconfig. For information about configuring alerts by using the ODSM interface, see Section 29.4.1.2, "Managing Alert Handlers by Using ODSM".

This section contains the following topics:

- Section 29.4.1.1.1, "To View the Configured Alert Handlers"
- Section 29.4.1.1.2, "To Enable an Alert Handler"
- Section 29.4.1.1.3, "To Create a New Alert Handler"
- Section 29.4.1.1.4, "To Delete an Alert Handler"
- Section 29.4.1.1.5, "To Control the Allowed Alert Types"

#### 29.4.1.1.1 To View the Configured Alert Handlers

Oracle Unified Directory stores alert handlers information in the configuration file under the `cn=Alert Handlers,cn=config` subtree. You can access the information using the `dsconfig` command.

To display a list of alert handlers, run the following `dsconfig` command:

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
 list-alert-handlers

Alert Handler     : Type : enabled
------------------:------:--------
JMX Alert Handler : jmx  : false
```

#### 29.4.1.1.2 To Enable an Alert Handler

The JMX alert handler is disabled by default. Before you begin, you must configure JMX on the server. For more information, see Section 29.5.3, "Monitoring the Server With JConsole".

1. To list the alert handler's properties, use the `dsconfig` command as follows.

   ```
   $ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
     get-alert-handler-prop --handler-name "JMX Alert Handler"

   Property            : Value(s)
   --------------------:---------------------------------------------
   disabled-alert-type : -
   enabled             : false
   enabled-alert-type  : -
   ```

2. To enable the alert handler, use `dsconfig` as follows.

   ```
   $ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
     set-alert-handler-prop --handler-name "JMX Alert Handler" --set enabled:true
   ```

3. Verify the change by using `dsconfig`.

   ```
   $ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
   ```

```
  get-alert-handler-prop --handler-name "JMX Alert Handler"

Property            : Value(s)
--------------------:-------------------------------------------
disabled-alert-type : -
enabled             : true
enabled-alert-type  : -
```

#### 29.4.1.1.3 To Create a New Alert Handler

The following example configures a new SMTP handler. Before starting this procedure, you must have configured an SMTP server for Oracle Unified Directory.

1. To create an alert handler run `dsconfig` with the `create-alert-handler` subcommand.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  create-alert-handler --handler-name "my SMTP Handler" --type smtp \
  --set enabled:true --set message-body:"Alert Type: %%alert-type%%
  \n\nAlert ID: %%alert-id%%\n\nAlert Message: %%alert-message%%" \
  --set message-subject:"Alert Message" \
  --set recipient-address:directorymanager@example.com \
  --set sender-address:OUD-Alerts@directory.example.com
```

2. View the list of alert handlers as follows.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  list-alert-handlers
```

#### 29.4.1.1.4 To Delete an Alert Handler

To delete an alert handler, use the `dsconfig delete-alert-handler` command. The following example removes the JMX alert handler.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  delete-alert-handler --handler-name "JMX Alert Handler"
```

You can simply *disable* an alert handler instead of deleting it. In this case, the alert handler is available if you need to enable it again in the future. For more information, see Section 29.4.1.1.5, "To Control the Allowed Alert Types".

#### 29.4.1.1.5 To Control the Allowed Alert Types

For a list of all supported alert types, see Section 29.4.1.3, "Supported Alert Types".

By default, all the supported alert types are allowed. If you specify a value for the `enabled-alert-type` property, only alerts with one of those types are allowed. If you specify a value for the `disabled-alert-type` property, all alert types except for the values in that property are allowed. Alert types are specified by their Java class, as shown in this example.

To disable an alert type, specify its Java class as a value of the `disabled-alert-type` property.

This command disables the startup alert from the JMX Alert Handler.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  set-alert-handler-prop --handler-name "JMX Alert Handler" \
  --set disabled-alert-type:org.opends.server.DirectoryServerStarted
```

### 29.4.1.2 Managing Alert Handlers by Using ODSM

The following sections describe how to manage the alert handler configuration by using ODSM. For information about configuring alert handlers by using `dsconfig`, see Section 29.4.1.1, "Managing Alert Handlers by Using `dsconfig`".

#### 29.4.1.2.1 Create an Alert Handler

To create an alert handler by using ODSM, follow these steps:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager."
2. Select the **Configuration** tab.
3. From the **Create** menu, select **Alert Handler**.
4. Select the type of alert handler that you want to create:
   - **JMX.** This alert handler is used to generate JMX notifications to alert administrators of significant events that occur within the server.
   - **SMTP.** This alert handler is used to send e-mail messages to notify administrators of significant events that occur within the server.
5. Enter the properties to configure the connection handler in the right hand pane.

   The configurable properties will depend on the type of alert handler that you have selected. For a comprehensive list of all configurable properties, and their allowed values, see "The Alert Handler Configuration" in the *Oracle Unified Directory Configuration Reference*.

   > **Note:** By default, all alert types are allowed. If you specify one or more values in the **Enabled Alert Type** field, only alerts with one of those types are allowed. If you specify one or more values in the **Disabled Alert Type** field, all alert types except for the values in that field are allowed.
   >
   > For a list of all supported alert types, see Section 29.4.1.3, "Supported Alert Types".

6. When you have configured the required properties for your specific alert handler type, click **Create**.

#### 29.4.1.2.2 Modify an Alert Handler

You can use ODSM to modify an existing alert handler, as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager."
2. Select the **Configuration** tab.
3. Expand the **General Configuration** element.
4. Expand the **Alert Handlers** element.
5. Select the alert handler whose properties you want to modify.
6. The properties are display in the right hand pane.
7. When you have modified the required properties, click **Apply**.

#### 29.4.1.2.3 Delete an Alert Handler

You can use ODSM to modify an existing alert handler, as follows:

1. Connect to the directory server from ODSM, as described in Section 18.2, "Connecting to the Server From Oracle Directory Services Manager."
2. Select the **Configuration** tab.
3. Expand the **General Configuration** element.
4. Expand the **Alert Handlers** element.
5. Select the alert handler that you want to delete and click the **Delete configuration** icon.
6. You are prompted to confirm the deletion. Click **Yes**.

#### 29.4.1.3 Supported Alert Types

The server sends out message alerts when an alert type event occurs in the system. The supported alert types are defined in the following table.

| Alert Type | Description |
|---|---|
| Access Control Disabled<br>Java Class: `org.opends.server.AccessControlDisabled` | Notify administrator that the access control handler has been disabled. |
| Access Control Enabled<br>Java Class: `org.opends.server.Enabled` | Notify administrator that the access control handler has been enabled. |
| Access Control Parse Failed<br>Java Class: `org.opends.server.authentication.dseecompat.ACIParseFailed` | Notify administrator if the Oracle Directory Server Enterprise Edition compatible access control subsystem failed to correctly parse one or more ACI rules when the server is first started. |
| Backend Environment Unusable<br>Java Class: `org.opends.server.BackendRunRecovery` | Notify administrator that the JE back end throws a `RunRecoveryException` and the directory server needs to be restarted. |
| Cannot Copy Schema Files<br>Java Class: `org.opends.server.CannotCopySchemaFiles` | Notify administrator if a problem occurs while attempting to create copies of the existing schema configuration before making a schema update, and the schema configuration is left in a potentially inconsistent state. |
| Cannot Find Recurring Task<br>Java Class: `org.opends.server.CannotFindRecurringTask` | Notify administrator if the directory server is unable to locate a recurring task definition in order to schedule the next iteration once the previous iteration has completed. |
| Cannot Rename Current Task File<br>Java Class: `org.opends.server.CannotRenameCurrentTaskFile` | Notify administrator if the directory server is unable to rename the current tasks backing file in the process of trying to write an updated version. |
| Cannot Rename New Task File<br>Java Class: `org.opends.server.CannotRenameNewTaskFile` | Notify administrator if the directory server is unable to rename the new tasks backing file into place. |
| Cannot Schedule Recurring Iteration<br>Java Class: `org.opends.server.CannotScheduleRecurringIteration` | Notify administrator if the directory server is unable to schedule an iteration of a recurring task. |

| Alert Type | Description |
|---|---|
| Cannot Write Configuration<br>Java Class:<br>`org.opends.server.CannotWriteConfig` | Notify administrator if the directory server is unable to write its updated configuration for some reason and so the server cannot exhibit the new configuration if it is restarted. |
| Cannot Write New Schema Files<br>Java Class:<br>`org.opends.server.CannotWriteNewSchemaFiles` | Notify administrator if a problem occurs while attempting to write new versions of the server schema configuration files, and the schema configuration is left in a potentially inconsistent state. |
| Cannot Write Task File<br>Java Class:<br>`org.opends.server.CannotWriteTaskFile` | Notify administrator if the directory server is unable to write an updated tasks backing file for some reason. |
| Distribution Backend Does Not Support PreRead Control<br>Java Class:<br>`com.sun.dps.server.distribution.globalindex.UnsupportedDirectoryBackend` | Notify administrators if the distribution is unable to maintain the content of the global index catalog. This will happen \ if one or more servers do not support the Pre-Read Entry Control (RFC 4527) |
| Entering Lockdown Mode<br>Java Class:<br>`org.opends.server.EnteringLockdownMode` | Notify administrator that the directory server is entering lockdown mode, in which only root users will be allowed to perform operations and only over the loopback address. |
| LDAP Connection Handler Consecutive Failures<br>Java Class:<br>`org.opends.server.LDAPHandlerDisabledByConsecutiveFailures` | Notify administrator of consecutive failures that have occurred in the LDAP connection handler that have caused it to become disabled. |
| LDAP Connection Handler Uncaught Error<br>Java Class:<br>`org.opends.server.LDAPHandlerUncaughtError` | Notify administrator of uncaught errors in the LDAP connection handler that have caused it to become disabled. |
| LDAP Server Extension Failed<br>Java Class:<br>`com.sun.dps.server.workflowelement.proxyldap.LDAPServerExtension.LDAPServerExtensionDown` | Notify administrator that the LDAP Server Extension has been detected as Down. |
| LDAP Server Extension is Up<br>Java Class:<br>`com.sun.dps.server.workflowelement.proxyldap.LDAPServerExtension.LDAPServerExtensionUp` | Notify administrator that the LDAP Server Extension has been detected as UP. |
| LDIF Backend Cannot Write Update<br>Java Class:<br>`org.opends.server.LDIFBackendCannotWriteUpdate` | Notify administrator that an LDIF back end was unable to store an updated copy of the LDIF file after processing a write operation. |
| LDIF ConnHandler Parse Error<br>Java Class:<br>`org.opends.server.LDIFConnectionHandlerParseError` | Notify administrator that the LDIF connection handler encountered an unrecoverable error while attempting to parse an LDIF file. |
| LDIF ConnHandler IO Error<br>Java Class:<br>`org.opends.server.LDIFConnectionHandlerIOError` | Notify administrator that the LDIF connection handler encountered an I/O error that prevented it from completing its processing. |

| Alert Type | Description |
|---|---|
| Leaving Lockdown Mode<br>Java Class:<br>`org.opends.server.LeavingLockdownMode` | Notify administrator that the directory server is leaving lockdown mode. |
| Manual Config Edit Handled<br>Java Class:<br>`org.opends.server.ManualConfigEditHandled` | Notify administrator if the directory server detects that its configuration has been manually edited with the server online and those changes were overwritten by another change made through the server. The manually-edited configuration will be copied off to another location. |
| Manual Config Edit Lost<br>Java Class:<br>`org.opends.server.ManualConfigEditLost` | Notify administrator if the directory server detects that its configuration has been manually edited with the server online and those changes were overwritten by another change made through the server. The manually-edited configuration could not be preserved due to an unexpected error. |
| New route elected by the SaturationLoadBalancingAlgorithm<br>Java Class:<br>`com.sun.dps.server.SaturationLoadBalancer` | Notify administrator that a new route has been elected as active route by the saturation load balancing algorithm. |
| New route elected by the FailoverLoadBalancingAlgorithm<br>Java Class:<br>`com.sun.dps.server.FailoverLoadBalancer` | Notify administrator that a new route has been elected as the active route by the failover load balancing algorithm. |
| Replication Unresolved Conflict<br>Java Class:<br>`org.opends.server.replication.UnresolvedConflict` | Notify administrator if the multimaster replication cannot automatically resolve a conflict. |
| Server Started<br>Java Class:<br>`org.opends.server.DirectoryServerStarted` | Notify administrator that the directory server has completed its startup process. |
| Server Shutdown<br>Java Class:<br>`org.opends.server.DirectoryServerShutdown` | Notify administrator that the directory server has begun the process of shutting down. |
| State change for a Saturation Load Balancing Route<br>Java Class:<br>`com.sun.dps.server.SaturationLoadBalancer` | Notify administrator that the saturation load balancing route state has changed (either from saturated to not saturated or from not saturated to saturated). |
| Uncaught Exception<br>Java Class:<br>`org.opends.server.UncaughtException` | Notify administrator if a directory server thread has encountered an uncaught exception that caused the thread to terminate abnormally. The impact that this problem has on the directory server depends on which thread was impacted and the nature of the exception. |
| Unique Attr Sync Conflict<br>Java Class:<br>`org.opends.server.UniqueAttributeSynchronizationConflict` | Notify administrator that a unique attribute conflict has been detected during synchronization processing. |

| Alert Type | Description |
|---|---|
| Unique Attr Sync Error<br>Java Class:<br>`org.opends.server.UniqueAttributeSynchronizationError` | Notify administrator that an error occurred while attempting to perform unique attribute conflict detection during synchronization processing. |
| Unsupported Directory Backend<br>Java Class:<br>`com.sun.dps.server.distribution.globalindex.UnsupportedDirectoryBackend` | Notify administrator that the distribution is unable to maintain the content of the global index catalog. This will happen if one or more servers do not support the Pre-Read Entry Control (RFC 4527). |

## 29.4.2 Managing Account Status Notification Handlers

Account status notification handlers provide alerts on events during password policy processing. By default, the Error Log Account Status Notification handler is set to enabled upon initial configuration. The server writes a message to the server error log when one of the following events has been configured in the password policy and occurs during the course of password policy processing:

- `account-temporarily-locked`
- `account-permanently-locked`
- `account-unlocked`
- `account-idle-locked`
- `account-reset-locked`
- `account-disabled`
- `account-expired`
- `password-expired`
- `password expiring`
- `password-reset`
- `password-changed`

The error log is located at `instance-dir/OUD/logs/errors`.

### 29.4.2.1 To View the Configured Account Status Notification Handlers

Use `dsconfig` with the `list-account-status-notification-handlers` subcommand.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  list-account-status-notification-handlers

Account Status Notification Handler : Type      : enabled
------------------------------------:-----------:--------
Error Log Handler                   : error-log : true
SMTP Handler                        : smtp      : false
```

### 29.4.2.2 To Enable Account Status Notification Handlers

You can enable an existing account status notification handler using the `dsconfig` command. By default, the directory server enables the Error Log Handler when the server is initially configured. This example enables the SMTP notification handler.

1. To view the `enabled` property use `dsconfig` with the `get-account-status-notification-handler-prop` subcommand.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  get-account-status-notification-handler-prop --handler-name "SMTP Handler" \
  --property enabled

Property : Value(s)
---------:---------
enabled  : false
```

2. To enable the notification handler use `dsconfig` with the `set-account-status-notification-handler-prop` subcommand.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  set-account-status-notification-handler-prop --handler-name "SMTP Handler" \
  --set property:enabled
```

#### 29.4.2.3 To Create a New Account Status Notification Handler

1. Use `dsconfig` with the `create-account-status-notification-handler` subcommand to create the handler.

   When you specify the type, you can use either `error-log` or `generic` (default).

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  create-account-status-notification-handler \
  --handler-name "My Password Reset Logger" --type error-log \
  --set enabled:true --set account-status-notification-type:password-reset
```

2. Use `dsconfig` to view the list of account status notification handlers.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  list-account-status-notification-handlers

Account Status Notification Handler : Type      : enabled
------------------------------------:-----------:--------
Error Log Handler                   : error-log : true
my Password Reset Logger            : error-log : true
SMTP Handler                        : smtp      : false
```

#### 29.4.2.4 To Delete an Account Status Notification Handler

You can disable an account status notification handler instead of deleting it. In this case, the alert handler is available if you need to enable it again in the future.

You can remove an account status notification handler entirely by using `dsconfig`.

Use `dsconfig` with the `delete-account-status-notification-handler` subcommand.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
  delete-account-status-notification-handler \
  --handler-name "My Password Reset Logger"
```

## 29.5 Monitoring the Server With LDAP

Oracle Unified Directory provides a variety of methods to monitor the current state of the server for debugging or troubleshooting purposes.

The topics in this section assume that you have configured monitoring providers on the server. For more information, see Section 29.2, "Configuring Monitor Providers".

You can monitor the server over LDAP in several ways. These are described in the following sections:

## 29.5.1 Viewing Monitoring Information Using the cn=monitor Entry

The directory server records system, performance, and version information as an entry with the base DN of cn=monitor. This entry provides useful performance metrics and server state information that you can use to monitor and debug a directory server instance.

You can access the cn=monitor suffix over the administration port only. There are advantages to using the administration port to access monitoring information. The main advantage of the administration connector is the separation of user traffic and administration traffic.

For example, if you monitor the number of connections on the LDAP Connection Handler ("cn=Client Connections,cn=LDAP Connection Handler 0.0.0.0 port port-number,cn=monitor") over the regular LDAP port, your monitoring data are "polluted" by the monitoring request itself. All of the examples in this section use the administration port, over SSL. For more information, see Section 14.3, "Managing Administration Traffic to the Server".

### 29.5.1.1 Monitored Attributes in the Proxy

Monitoring information related to the proxy can be collected at the level under cn=Monitor for dozens of attributes, including those relating to the following:

- Workflows: cn=workflow,cn=monitor
- Network Groups: cn=Network Groups,cn=monitor
- Load balancers: cn=load balancing,cn=monitor
- Distributions: cn=distribution,cn=monitor
- Global Index Catalogs: cn=Global Index Catalogs,cn=monitor
- Client Connections: cn=Client Connections,cn=monitor or under cn=Client Connections,cn=LDAP Connection Handler *0.0.0.0* port *port number*,cn=monitor
- LDAP Connection Handler: cn=LDAP Connection Handler *0.0.0.0* port *port number*,cn=monitor
- LDAP Connection Handler Statistics: cn=LDAP Connection Handler *0.0.0.0* port*port number* statistics,cn=monitor
- SNMP Connection Handler: cn=SNMP Connection Handler,cn=Monitor
- JMX Connection Handler: cn=JMX Connection Handler *port number*,cn=monitor
- Administration Connector: cn=Administration Connector *0.0.0.0* port *port number*,cn=monitor
- System Information: cn=System Information,cn=monitor
- Version: cn=Version,cn=monitor
- Back-end LDAP servers: cn=LDAP Servers,cn=monitor

- JVM stack traces: `cn=JVM Stack Trace,cn=monitor`
- JVM memory usage: `cn=JVM Memory Usage,cn=Monitor`
- SNMP: `cn=SNMP,cn=Monitor`
- Backend Backup: `cn=backup Backend,cn=monitor`
- Monitoring of back-end data: `cn=monitor Backend,cn=monitor`
- Tasks on the Backend Backup: `cn=backup Backend,cn=monitor`
- Entry caches: `cn=Entry Caches,cn=monitor`
- Work queues: `cn=Work Queue,cn=monitor`

Other attributes are monitored under each of the above in the dn tree. For example, client connections are monitored under both `cn=Client Connections,` *0.0.0.0* `port` *port number*`,cn=monitor` and under `cn=Client Connections,cn=Administration Connector` *0.0.0.0* `port` *port number*`, cn=monitor`.

A workflow element is monitored under the part of the tree to which that workflow element relates. For example, a load balancing workflow element can be monitored as `cn=load-bal-route1,cn=load balancing,cn=monitor`

Hundreds of statistics are collected by the proxy for monitoring. For example, for the persistent search function, `psearchCount` lists the number of persistent search operations and `psearchTotalCount` lists the number of persistent search operations since the last server restart.

You can list all of these statistics by using the `ldapsearch` command on the `cn=monitor` entry, as described in Section 29.5.1.2, "To View the Available Monitoring Information". Note that access to the `cn=monitor` entry is restricted to users who have the bypass ACI privilege.

The following procedures use the `ldapsearch` command at the command line interface.

To view status information on the replication of global indexes, you can use the `gicadm status-replication` command. For more information, see Section 15.1.7.2.5, "To View the Status of a Replicated Global Index Catalog Configuration".

### 29.5.1.2 To View the Available Monitoring Information

Use the `ldapsearch` command to inspect the attributes of `cn=monitor`. This example lists the base DNs of each monitor entry.

Run the `ldapsearch` command with a search scope of `sub` and the search attribute `1.1`.

This search attribute indicates that no attributes should be included in the matching entries.

```
$ ldapsearch -h localhost -p 4444 -D "cn=directory manager" -j pwd-file --useSSL \
  --trustAll -s sub -b "cn=monitor" "(objectclass=*)" "1.1"
dn: cn=monitor
dn: cn=Client Connections,cn=monitor
dn: cn=ads-truststore Backend,cn=monitor
dn: cn=Network Groups,cn=monitor
dn: cn=internal,cn=Network Groups,cn=monitor
dn: cn=default,cn=Network Groups,cn=monitor
dn: cn=LDAP Connection Handler 0.0.0.0 port 1389 Statistics,cn=monitor
```

```
dn: cn=Administration Connector 0.0.0.0 port 4444,cn=monitor
dn: cn=Client Connections,cn=Administration Connector 0.0.0.0 port 4444,cn=monitor
dn: cn=backup Backend,cn=monitor
dn: cn=Version,cn=monitor
dn: cn=Work Queue,cn=monitor
dn: cn=System Information,cn=monitor
dn: cn=userRoot Database Environment,cn=monitor
dn: cn=tasks Backend,cn=monitor
dn: cn=adminRoot Backend,cn=monitor
dn: cn=userRoot Backend,cn=monitor
dn: cn=schema Backend,cn=monitor
dn: cn=LDAP Connection Handler 0.0.0.0 port 1389,cn=monitor
dn: cn=admin,cn=Network Groups,cn=monitor
dn: cn=Client Connections,cn=LDAP Connection Handler 0.0.0.0 port 1389,cn=monitor
dn: cn=JVM Memory Usage,cn=monitor
dn: cn=Administration Connector 0.0.0.0 port 4444 Statistics,cn=monitor
dn: cn=JVM Stack Trace,cn=monitor
dn: cn=Entry Caches,cn=monitor
dn: cn=monitor Backend,cn=monitor
```

#### 29.5.1.3 To Monitor General-Purpose Server Information

Use the `ldapsearch` command with a base DN of `"cn=monitor"`.

```
$ ldapsearch -h localhost -p 4444 -D "cn=directory manager" -j pwd-file --useSSL \
  --trustAll -s base -b "cn=monitor" "(objectclass=*)"
```

Output will be similar to the following:

```
dn: cn=monitor
startTime: 20120110110156Z
objectClass: extensibleObject
objectClass: top
objectClass: ds-monitor-entry
cn: monitor
vendorName: Oracle Corporation
currentTime: 20120111082026Z
vendorVersion: Oracle Unified Directory 11.1.2.0
maxConnections: 1
productName: Oracle Unified Directory
currentConnections: 1
totalConnections: 8
upTime: 57 days 21 hours 18 minutes 30 seconds
```

#### 29.5.1.4 To Monitor System Information

Use the `ldapsearch` command with the base DN `"cn=System Information,cn=monitor"`.

```
$ ldapsearch -h localhost -p 4444 -D "cn=directory manager" -j pwd-file --useSSL \
  --trustAll -s base -b "cn=System Information,cn=monitor" "(objectclass=*)"
```

Depending on your configuration, output will be similar to the following:

```
dn: cn=System Information,cn=monitor
instancePath: /local/asinst_2/OUD
javaVersion: 1.6.0_10
jvmArchitecture: 32-bit
jvmArguments: "-Dorg.opends.server.scriptName=start-ds"
jvmVersion: 11.0-b15
classPath: /local/instances/OUD/classes:
/local/asinst_2/OUD/resources/resources.jar:
```

```
/local/asinst_2/OUD/lib/activation.jar:
/local/asinst_2/OUD/lib/aspectjrt.jar:
/local/asinst_2/OUD/lib/je.jar:
/local/asinst_2/OUD/lib/mail.jar:
/local/asinst_2/OUD/lib/OUD_de.jar:
/local/asinst_2/OUD/lib/OUD_es.jar:
/local/asinst_2/OUD/lib/OUD_fr.jar:
/local/asinst_2/OUD/lib/OUD_ja.jar:
/local/asinst_2/OUD/lib/OUD.jar:
/local/asinst_2/OUD/lib/OUD_zh_CN.jar:
/local/asinst_2/OUD/lib/quicksetup.jar
usedMemory: 83361792
freeUsedMemory: 21020432
objectClass: extensibleObject
objectClass: top
objectClass: ds-monitor-entry
javaVendor: Oracle Corporation
operatingSystem: SunOS 5.11 x86
cn: System Information
systemName: llandudno
workingDirectory: /local/asinst_2/OUD/bin
maxMemory: 518717440
availableCPUs: 2
javaHome: /usr/jdk/instances/jdk1.6.0/jre
jvmVendor: Oracle Corporation
```

#### 29.5.1.5 To Monitor Version Information

Use the `ldapsearch` command with base DN `"cn=Version,cn=monitor"`.

```
$ ldapsearch -h localhost -p 4444 -D "cn=directory manager" -j pwd-file --useSSL \
  --trustAll -b "cn=Version,cn=Monitor" "(objectclass=*)"
```

The output will be similar to the following:

```
dn: cn=Version,cn=monitor
shortName: OUD
labelNumber: 1112231410
objectClass: extensibleObject
objectClass: top
objectClass: ds-monitor-entry
cn: Version
pointVersion: 2
compactVersion: OUD-11.1
buildID: 20111224012512Z
majorVersion: 11
productName: Oracle Unified Directory
minorVersion: 1
versionQualifier: 0
fullVersion: Oracle Unified Directory 11.1.2.0
```

#### 29.5.1.6 To Monitor the User Root Back End

The `userRoot` back end is the back-end database (the JE environment) for your data. The monitor displays the back end's general properties, such as writability mode, base DN, back-end IDs, entry count, and other properties.

Use the `ldapsearch` command with base DN `"cn=userRoot Backend,cn=monitor"`.

```
$ ldapsearch -h localhost -p 4444 -D "cn=directory manager" -j pwd-file --useSSL \
  --trustAll -s base -b "cn=userRoot Backend,cn=monitor" "(objectclass=*)"
```

Depending on your configuration, output will be similar to the following:

```
dn: cn=userRoot Backend,cn=monitor
objectClass: top
objectClass: ds-monitor-entry
objectClass: ds-backend-monitor-entry
ds-backend-is-private: FALSE
cn: userRoot Backend
ds-backend-writability-mode: enabled
ds-backend-entry-count: 2002
ds-backend-id: userRoot
ds-base-dn-entry-count: 2002 dc=example,dc=com
ds-backend-base-dn: dc=example,dc=com
```

#### 29.5.1.7 To Monitor the Backup Back End

Use the `ldapsearch` command with base DN `"cn=backup Backend,cn=monitor"`.

```
$ ldapsearch -h localhost -p 4444 -D "cn=directory manager" -j pwd-file --useSSL \
  --trustAll -s base -b "cn=backup Backend,cn=monitor" "(objectclass=*)"
```

Depending on your configuration, output will be similar to the following:

```
dn: cn=backup Backend,cn=monitor
objectClass: top
objectClass: ds-monitor-entry
objectClass: ds-backend-monitor-entry
ds-backend-is-private: TRUE
cn: backup Backend
ds-backend-writability-mode: disabled
ds-backend-entry-count: 1
ds-backend-id: backup
ds-base-dn-entry-count: 1 cn=backups
ds-backend-base-dn: cn=backups
```

#### 29.5.1.8 To Monitor the Tasks Back End

Tasks are administrative functions (such as `import-ldif`, `export-ldif`, `backup`, and `restore`) that can be scheduled for processing at some future date or on a recurring basis. The monitor displays the tasks back end's general properties, such as writability mode, base DN, back-end IDs, entry count, and other properties.

Use the `ldapsearch` command with base DN `"cn=Tasks Backend,cn=monitor"`.

```
$ ldapsearch -h localhost -p 4444 -D "cn=directory manager" -j pwd-file --useSSL \
  --trustAll -s base -b "cn=Tasks Backend,cn=monitor" "(objectclass=*)"
```

Depending on your configuration, output will be similar to the following:

```
dn: cn=tasks Backend,cn=monitor
objectClass: top
objectClass: ds-monitor-entry
objectClass: ds-backend-monitor-entry
ds-backend-is-private: TRUE
cn: tasks Backend
ds-backend-writability-mode: enabled
ds-backend-entry-count: 3
ds-backend-id: tasks
ds-base-dn-entry-count: 3 cn=tasks
```

```
ds-backend-base-dn: cn=tasks
```

#### 29.5.1.9 To Monitor the `monitor` Back End

This monitor displays the back end's general properties, such as writability mode, base DN, back-end IDs, entry count, and other properties.

Use the `ldapsearch` command with base DN `"cn=monitor Backend,cn=monitor"`.

```
$ ldapsearch -h localhost -p 4444 -D "cn=directory manager" -j pwd-file --useSSL \
  --trustAll -s base -b "cn=monitor Backend,cn=monitor" "(objectclass=*)"
```

Depending on your configuration, output will be similar to the following:

```
dn: cn=monitor Backend,cn=monitor
objectClass: top
objectClass: ds-monitor-entry
objectClass: ds-backend-monitor-entry
ds-backend-is-private: TRUE
cn: monitor Backend
ds-backend-writability-mode: disabled
ds-backend-entry-count: 25
ds-backend-id: monitor
ds-base-dn-entry-count: 25 cn=monitor
ds-backend-base-dn: cn=monitor
```

#### 29.5.1.10 To Monitor the Schema Back End

This monitor displays the schema back end's general properties, such as writability mode, base DN, back-end IDs, entry count, and other properties.

Use the `ldapsearch` command with base DN `"cn=schema Backend,cn=monitor"`.

```
$ ldapsearch -h localhost -p 4444 -D "cn=directory manager" -j pwd-file --useSSL \
  --trustAll -s base -b "cn=schema Backend,cn=monitor" "(objectclass=*)"
```

Depending on your configuration, output will be similar to the following:

```
dn: cn=schema Backend,cn=monitor
objectClass: top
objectClass: ds-monitor-entry
objectClass: ds-backend-monitor-entry
ds-backend-is-private: TRUE
cn: schema Backend
ds-backend-writability-mode: enabled
ds-backend-entry-count: 1
ds-backend-id: schema
ds-base-dn-entry-count: 1 cn=schema
ds-backend-base-dn: cn=schema
```

#### 29.5.1.11 To Monitor the `adminRoot` Back End

This monitor displays the `adminRoot` back end's general properties, such as writability mode, base DN, back-end IDs, entry count, and other properties.

Use the `ldapsearch` command with base DN `"cn=adminRoot Backend,cn=monitor"`.

```
$ ldapsearch -h localhost -p 4444 -D "cn=directory manager" -j pwd-file --useSSL \
  --trustAll -s base -b "cn=adminRoot Backend,cn=monitor" "(objectclass=*)"
```

Depending on your configuration, output will be similar to the following:

```
dn: cn=adminRoot Backend,cn=monitor
objectClass: top
objectClass: ds-monitor-entry
objectClass: ds-backend-monitor-entry
ds-backend-is-private: TRUE
cn: adminRoot Backend
ds-backend-writability-mode: enabled
ds-backend-entry-count: 7
ds-backend-id: adminRoot
ds-base-dn-entry-count: 7 cn=admin data
ds-backend-base-dn: cn=admin data
```

#### 29.5.1.12 To Monitor the `ads-truststore` Back End

The `ads-truststore` holds a mirror, or copy, of the remote Administrative Directory Service (ADS) host's ADS key entry, so that the new instance can establish trust with existing servers in the ADS domain. The monitor displays the back end's general properties, such as writability mode, base DN, back-end IDs, entry count, and other properties.

Use the `ldapsearch` command with base DN `"cn=ads-truststore Backend,cn=monitor"`.

```
$ ldapsearch -h localhost -p 4444 -D "cn=directory manager" -j pwd-file --useSSL \
  --trustAll -s base -b "cn=ads-truststore Backend,cn=monitor" "(objectclass=*)"
```

Depending on your configuration, output will be similar to the following:

```
dn: cn=ads-truststore Backend,cn=monitor
objectClass: top
objectClass: ds-monitor-entry
objectClass: ds-backend-monitor-entry
ds-backend-is-private: TRUE
cn: ads-truststore Backend
ds-backend-writability-mode: enabled
ds-backend-entry-count: 3
ds-backend-id: ads-truststore
ds-base-dn-entry-count: 3 cn=ads-truststore
ds-backend-base-dn: cn=ads-truststore
```

#### 29.5.1.13 To Monitor Client Connections

This monitor represents *all* of the open client connections. Its contents are different to those of the DN `"cn=Client Connections,cn=LDAP Connection Handler 0.0.0.0 port 1389,cn=monitor"`, which describes the open client connections on the LDAP connection handler only.

Use the `ldapsearch` command with base DN `"cn=Client Connections,cn=monitor"`.

```
$ ldapsearch -h localhost -p 4444 -D "cn=directory manager" -j pwd-file --useSSL \
  --trustAll -s base -b "cn=Client Connections,cn=monitor" "(objectclass=*)"
```

Depending on your configuration, output will be similar to the following:

```
dn: cn=Client Connections,cn=monitor
connection: connID="11" connectTime="20090702125632Z" source="198.51.100.0:54044"
destination="198.51.100.23:1389" ldapVersion="3" authDN="cn=Directory
Manager,cn=Root DNs,
cn=config" security="none" opsInProgress="1"
```

```
cn: Client Connections
objectClass: extensibleObject
objectClass: top
objectClass: ds-monitor-entry
```

#### 29.5.1.14 To Monitor the LDAP Connection Handler

The LDAP connection handler is used to interact with clients over LDAP.

Use the `ldapsearch` command with base DN `"cn=LDAP Connection Handler 0.0.0.0 port port-number,cn=monitor"`.

```
$ ldapsearch -h localhost -p 4444 -D "cn=directory manager" -j pwd-file --useSSL \
  --trustAll -s base \
  -b "cn=LDAP Connection Handler 0.0.0.0 port 1389,cn=monitor" \
  "(objectclass=*)"
```

Depending on your configuration, output will be similar to the following:

```
dn: cn=LDAP Connection Handler 0.0.0.0 port 1389,cn=monitor
ds-connectionhandler-listener: 0.0.0.0:1389
ds-connectionhandler-num-connections: 1
ds-connectionhandler-protocol: LDAP
objectClass: top
objectClass: ds-monitor-entry
objectClass: ds-connectionhandler-monitor-entry
ds-mon-config-dn: cn=ldap connection handler,cn=connection handlers,cn=config
cn: LDAP Connection Handler 0.0.0.0 port 1389
ds-connectionhandler-connection: connID="22" connectTime="20120302133936Z"
source="198.51.100.0:39574" destination="198.51.100.23:1389" ldapVersion="3"
authDN="cn=Directory Manager,cn=Root DNs,cn=config" security="none"
opsInProgress="1"
```

#### 29.5.1.15 To Monitor LDAP Connection Handler Statistics

Use the `ldapsearch` command with base DN `"cn=LDAP Connection Handler 0.0.0.0 port port-number Statistics,cn=monitor"`.

```
$ ldapsearch -h localhost -p 4444 -D "cn=directory manager" -j pwd-file --useSSL \
  --trustAll -s base \
  -b "cn=LDAP Connection Handler 0.0.0.0 port 1389 Statistics,cn=monitor" \
  "(objectclass=*)"
```

Depending on your configuration, output will be similar to the following:

```
dn: cn=LDAP Connection Handler 0.0.0.0 port 1389 Statistics,cn=monitor
objectClass: ds-monitor-entry
objectClass: top
objectClass: extensibleObject
operationsCompleted: 37
compareRequests: 0
bytesWritten: 99488
extendedRequests: 0
addRequests: 0
bindRequests: 19
...(more output)
```

#### 29.5.1.16 To Monitor Connections on the LDAP Connection Handler

This monitor represents the open client connections on the LDAP connection handler.

Use the `ldapsearch` command with base DN `"cn=Client Connections,cn=LDAP Connection Handler 0.0.0.0 port port-number,cn=monitor"`.

```
$ ldapsearch -h localhost -p 4444 -D "cn=directory manager" -j pwd-file \
--useSSL --trustAll \
-b "cn=Client Connections,cn=LDAP Connection Handler 0.0.0.0 port 1389 \
cn=monitor" \
"(objectclass=*)"
```

Depending on your configuration, output will be similar to the following:

```
dn: cn=Client Connections,cn=LDAP Connection Handler 0.0.0.0 port 1389,cn=monitor
connection: connID="0" connectTime="20090706084747Z" source="198.51.100.0:57523"
destination="198.51.100.0:1389" ldapVersion="3" authDN="" security="none"
opsInProgress="0"
connection: connID="1" connectTime="20090706084747Z" source="198.51.100.0:57524"
destination="198.51.100.0:1389" ldapVersion="3" authDN="" security="none"
opsInProgress="0"
connection: connID="2" connectTime="20090706084747Z" source="198.51.100.0:57525"
destination="198.51.100.0:1389" ldapVersion="3" authDN="" security="none"
opsInProgress="0"
connection: connID="3" connectTime="20090706084747Z" source="198.51.100.0:57526"
destination="198.51.100.0:1389" ldapVersion="3" authDN="" security="none"
opsInProgress="0"
connection: connID="4" connectTime="20090706084747Z" source="198.51.100.0:57527"
destination="198.51.100.0:1389" ldapVersion="3" authDN="" security="none"
opsInProgress="0"
```

#### 29.5.1.17 To Monitor the Administration Connector

This monitor provides basic information about the administration connector. For more information, see Section 14.3, "Managing Administration Traffic to the Server".

Use the `ldapsearch` command with base DN `"cn=Administration Connector 0.0.0.0 port admin-port-number,cn=monitor"`.

```
$ ldapsearch -h localhost -p 4444 -D "cn=directory manager" -j pwd-file --useSSL \
  --trustAll -b "cn=Administration Connector 0.0.0.0 port 4444,cn=monitor" \
  "(objectclass=*)"
```

Depending on your configuration, output will be similar to the following:

```
objectClass: top
objectClass: ds-monitor-entry
objectClass: ds-connectionhandler-monitor-entry
dn: cn=Administration Connector 0.0.0.0 port 4444,cn=monitor
ds-connectionhandler-listener: 0.0.0.0:4444
ds-connectionhandler-num-connections: 0
ds-connectionhandler-protocol: LDAPS
cn: Administration Connector 0.0.0.0 port 4444
ds-mon-config-dn: cn=administration connector,cn=config
```

#### 29.5.1.18 To Monitor Administration Connector Statistics

This monitor provides extensive statistical information about operations that are performed through the administration connector. For more information, see Section 14.3, "Managing Administration Traffic to the Server".

Use the `ldapsearch` command with base DN `"cn=Administration Connector 0.0.0.0 port admin-port-number Statistics,cn=monitor"`.

```
$ ldapsearch -h localhost -p 4444 -D "cn=directory manager" -j pwd-file --useSSL \
```

```
  --trustAll \
  -b "cn=Administration Connector 0.0.0.0 port 4444 Statistics,cn=monitor" \
  "(objectclass=*)"
```

Depending on your configuration, output will be similar to the following:

```
dn: cn=Administration Connector 0.0.0.0 port 4444 Statistics,cn=monitor
compareResponses: 0
connectionsClosed: 1
searchResultsDone: 4
ds-mon-resident-time-mod-operations-total-time: 92257568
extendedResponses: 0
bindRequests: 2
operationsAbandoned: 0
bytesWritten: 45056
addResponses: 0
addRequests: 0
ds-mon-resident-time-moddn-operations-total-time: 0
ds-mon-extended-operations-total-count: 0
ds-mon-moddn-operations-total-count: 0
modifyResponses: 1
operationsCompleted: 7
...(more output)...
```

### 29.5.1.19 To Monitor Connections on the Administration Connector

This monitor represents the open client connections on the Administration Connector.

Use the `ldapsearch` command with base DN `"cn=Client Connections,cn=Administration Connector 0.0.0.0 port port-number,cn=monitor"`.

```
$ ldapsearch -h localhost -p 4444 -D "cn=directory manager" -j pwd-file \
--useSSL --trustAll \
-b "cn=Client Connections,cn=Administration Connector 0.0.0.0 \
port 4444,cn=monitor" \
"(objectclass=*)"
```

Depending on your configuration, output will be similar to the following:

```
objectClass: top
objectClass: ds-monitor-entry
objectClass: extensibleObject
dn: cn=Client Connections,cn=Administration Connector 0.0.0.0 port 4444,cn=monitor
connection: connID="339" connectTime="20120307075218Z" source="198.51.100.0:48213"
destination="198.51.100.0:4444" ldapVersion="3" authDN="" security="TLS"
 opsInProgress="1"
cn: Client Connections
```

### 29.5.1.20 To Monitor the LDIF Connection Handler

The LDIF connection handler is used to process changes that are read from an LDIF file, using internal operations. Monitoring information for the LDIF connection handler is only available if the connection handler is enabled.

Use the `ldapsearch` command with base DN `"cn=LDIF Connection Handler,cn=monitor"`.

```
$ ldapsearch -h localhost -p 4444 -D "cn=directory manager" -j pwd-file --useSSL \
  --trustAll -s base -b "cn=LDIF Connection Handler,cn=monitor" "(objectclass=*)"
```

Depending on your configuration, output will be similar to the following:

```
objectClass: top
objectClass: ds-monitor-entry
objectClass: ds-connectionhandler-monitor-entry
dn: cn=LDIF Connection Handler,cn=monitor
ds-connectionhandler-num-connections: 0
ds-connectionhandler-protocol: LDIF
ds-mon-config-dn: cn=ldif connection handler,cn=connection handlers,cn=config
cn: LDIF Connection Handler
```

#### 29.5.1.21 To Monitor the Work Queue

The work queue keeps track of outstanding client requests and ensures that they are processed.

Use the `ldapsearch` command with base DN `"cn=Work Queue,cn=monitor"`.

```
$ ldapsearch -h localhost -p 4444 -D "cn=directory manager" -j pwd-file --useSSL \
  --trustAll -s base -b "cn=Work Queue,cn=monitor" "(objectclass=*)"
```

Depending on your configuration, output will be similar to the following:

```
dn: cn=Work Queue,cn=monitor
currentRequestBacklog: 0
objectClass: extensibleObject
objectClass: top
objectClass: ds-monitor-entry
requestsSubmitted: 25
cn: Work Queue
maxRequestBacklog: 0
averageRequestBacklog: 0
requestsRejectedDueToQueueFull: 0
```

#### 29.5.1.22 To Monitor JVM Stack Trace Information

You can access JVM Stack Trace information for your directory server instance. This resource monitor is implemented in the `org.opends.server.monitors.StackTraceMonitorProvider` class and requires no custom configuration.

Use the `ldapsearch` command with the base DN `"cn=JVM Stack Trace,cn=monitor"`.

```
$ ldapsearch -h localhost -p 4444 -D "cn=directory manager" -j pwd-file --useSSL \
  --trustAll -s base -b "cn=JVM Stack Trace,cn=monitor" "(objectclass=*)"
```

Depending on your configuration, the beginning of the output will be similar to the following:

```
dn: cn=JVM Stack Trace,cn=monitor
cn: JVM Stack Trace
jvmThread: id=2 ---------- Reference Handler ----------
jvmThread: id=2 frame[0]=java.lang.Object.wait(Object.java:native)
jvmThread: id=2 frame[1]=java.lang.Object.wait(Object.java:485)
jvmThread: id=2 frame[2]=java.lang.ref.Reference$ReferenceHandler.run(Reference.
java:116)
jvmThread: id=3 ---------- Finalizer ----------
jvmThread: id=3 frame[0]=java.lang.Object.wait(Object.java:native)
jvmThread: id=3 frame[1]=java.lang.ref.ReferenceQueue.remove(ReferenceQueue.java
:116)
jvmThread: id=3 frame[2]=java.lang.ref.ReferenceQueue.remove(ReferenceQueue.java
:132)
jvmThread: id=3 frame[3]=java.lang.ref.Finalizer$FinalizerThread.run(Finalizer.j
```

```
ava:159)
jvmThread: id=4 ---------- Signal Dispatcher ----------
jvmThread: id=10 ---------- Time Thread ----------
jvmThread: id=10 frame[0]=sun.misc.Unsafe.park(Unsafe.java:native)
jvmThread: id=10 frame[1]=java.util.concurrent.locks.LockSupport.parkNanos(LockS
upport.java:198)
...(more output)...
```

### 29.5.1.23 To Monitor the JVM Memory Usage

Use the `ldapsearch` command with base DN `"cn=JVM Memory Usage,cn=monitor"`.

```
$ ldapsearch -h localhost -p 4444 -D "cn=directory manager" -j pwd-file --useSSL \
  --trustAll -s base -b "cn=JVM Memory Usage,cn=monitor" "(objectclass=*)"
```

Depending on your configuration, output will be similar to the following:

```
dn: cn=JVM Memory Usage,cn=monitor
ps-eden-space-bytes-used-after-last-collection: 0
ps-mark-sweep-total-collection-count: 0
code-cache-bytes-used-after-last-collection: 0
ps-old-gen-current-bytes-used: 25260472
ps-perm-gen-bytes-used-after-last-collection: 0
ps-scavenge-recent-collection-duration: 3
ps-scavenge-total-collection-count: 17
ps-eden-space-current-bytes-used: 32001992
ps-perm-gen-current-bytes-used: 21179960
ps-old-gen-bytes-used-after-last-collection: 0
ps-mark-sweep-total-collection-duration: 0
ps-mark-sweep-average-collection-duration: 0
ps-scavenge-average-collection-duration: 26
ps-scavenge-total-collection-duration: 443
objectClass: extensibleObject
objectClass: top
objectClass: ds-monitor-entry
ps-mark-sweep-recent-collection-duration: 0
ps-survivor-space-bytes-used-after-last-collection: 622592
cn: JVM Memory Usage
code-cache-current-bytes-used: 2143680
ps-survivor-space-current-bytes-used: 622592
```

### 29.5.1.24 To Monitor the `userRoot` Database Environment

The `userRoot` database environment utilizes the Berkeley DB Java Edition back end. JE monitoring data (data under `cn=*Database Environment,cn=monitor`) is reliable only in the short term. During high server activity (for example, anywhere from an hour to several days depending on the counter), this data can overflow. In such cases, the JE monitoring data can reflect negative values or positive but incorrect values. This is a known issue and is expected to be fixed in the next major release of the Berkeley DB Java Edition. Oracle SR numbers 15979 and 15985 correspond to this issue.

Use the `ldapsearch` command with base DN `"cn=userRoot Database Environment,cn=monitor"`.

```
$ ldapsearch -h localhost -p 4444 -D "cn=directory manager" -j pwd-file --useSSL \
  --trustAll -s base -b "cn=userRoot Database Environment,cn=monitor" \
  "(objectclass=*)"
dn: cn=userRoot Database Environment,cn=monitor
```

Depending on your configuration, output will be similar to the following:

```
EnvironmentNTempBufferWrites: 0
EnvironmentNNodesExplicitlyEvicted: 0
EnvironmentCleanerBacklog: 0
EnvironmentTotalLogSize: 5386067
EnvironmentLockBytes: 2000
EnvironmentNFullBINFlush: 2
EnvironmentNBINsStripped: 0
EnvironmentLastCheckpointEnd: 5385359
TransactionNCommits: 24
EnvironmentNCleanerEntriesRead: 0
EnvironmentNRepeatFaultReads: 2
TransactionNXACommits: 0
EnvironmentNClusterLNsProcessed: 0
TransactionNBegins: 24
LockNOwners: 25
...(more output)...
```

#### 29.5.1.25 To Monitor the Database Cache

The database (DB) cache is used to store Java Edition nodes. The DB cache is the critical component of your directory server's overall performance. Ensure that you tune and monitor the DB cache carefully. The DB cache includes the following nodes:

- Upper node
- Inner node
- Leaf node

The upper and inner nodes represents the internal B-tree structure and the leaf node represent the user entries. For best possible performance, it is recommended to have all the DB cache nodes in the DB cache. It is recommend to size the dbcache such that it contains at minimum the B-tree internal structure (the upper and inner nodes). If the dbcache is too short this can result in having lots of misses and frequent evictions which will badly affect directory server performance.

Tuning the size of the cache is done by:

- Setting the `dbcache-percent`
- Sizing appropriately the OUD JVM heap and especially the old generation.

You can monitor the DB cache by using the `ldapsearch` command with base DN `cn=userRoot Database Environment,cn=monitor`:

```
$ ldapsearch -h localhost -p 4444 -D "cn=directory manager" -j pwd-file --useSSL \
  --trustAll -s base -b "cn=userRoot Database Environment,cn=monitor" \
  "(objectclass=*)"
dn: cn=userRoot Database Environment,cn=monitor
```

Depending on your configuration, output will be similar to the following:

```
EnvironmentNTempBufferWrites: 0
EnvironmentNNodesExplicitlyEvicted: 0
EnvironmentCleanerBacklog: 0
EnvironmentTotalLogSize: 5386067
EnvironmentLockBytes: 2000
EnvironmentNFullBINFlush: 2
EnvironmentNBINsStripped: 0
EnvironmentLastCheckpointEnd: 5385359
TransactionNCommits: 24
EnvironmentNCleanerEntriesRead: 0
```

```
EnvironmentNRepeatFaultReads: 2
TransactionNXACommits: 0
EnvironmentNClusterLNsProcessed: 0
TransactionNBegins: 24
LockNOwners: 25
...(more output)...
```

The following DB cache hits and miss counters are described below:

| Counters | Description |
|---|---|
| `EnvironmentNUpperINsFetch` | Accumulated number of upper inner nodes fetched from the cache. |
| `EnvironmentNUpperINsFetchMiss` | Accumulated number of upper inner nodes miss. |
| `EnvironmentNBINsFetch` | Accumulated number of bottom inner nodes fetched from the cache. |
| `EnvironmentNBINsFetchMiss` | Accumulated number of upper inner nodes miss. |
| `EnvironmentNLNsFetch` | Accumulated number of leaf nodes fetched from the cache. |
| `EnvironmentNLNsFetchMiss` | Accumulated number of leaf nodes miss. |

For OUD to perform well it is recommended to have all the nodes in the dbcache or at least to have all the inner nodes in the dbcache.

As the values in cn=monitor are accumulations, it is important to compute deltas at regular interval (1mn for instance) and monitor the evolution of deltas over time. You must update the following:

```
DeltaNUpperINsMiss=EnvironmentNUpperINsFetchMiss -
EnvironmentNUpperINsFetchMissPrev
DeltaNUpperINsFetch=EnvironmentNUpperINsFetch - EnvironmentNUpperINsFetchPrev
DeltaBINsMiss=EnvironmentNBINsFetchMiss - EnvironmentNBINsFetchMissPrev
DeltaBINsFetch=EnvironmentNBINsFetch - EnvironmentNBINsFetchPrev
DeltaNLNsMiss=EnvironmentNLNsFetchMiss - EnvironmentNLNsFetchMissPrev
DeltaNLNsFetch=EnvironmentNLNsFetch - EnvironmentNLNsFetchPrev
```

You can run the Oracle Unified Directory with a minimal level of performance. It is recommend to have the B-Tree structure in the dbcache, as described below:

```
((DeltaNUpperINsMiss/DeltaNUpperINsFetch)*100) as close to 0 as possible
((DeltaNBINsMiss/DeltaNBINsFetch)*100) as close to 0 as possible (< 5% remains
acceptable)
```

To have the best possible performance, it is recommended for OUD to also have user entries in the dbcache, i-e:

`((DeltaNLNsMiss/DeltaNLNsFetch)*100)` as close to `0` as possible.

Start with Deltas ratio close to `0` after the import is complete (and data primed) and with time the Deltas ratio grows due to the database growth (bc of replication metadata, clean-min-utilizat° impact, growth of the entry (new apps) as well as the nb of entries ). Due to this, it is recommended to monitor the dbcache (via custom scripts or UI) and take appropriate actions such as increase the dbcache-percent or the OUD JVM heap.

#### 29.5.1.26 To Monitor the Entry Cache

You can access the aggregated state of all active entry caches for your directory server instance by accessing the `cn=Entry Caches,cn=Monitor` entry. The server can also request the "per cache" monitor data for a given instance if the entry cache instances are enabled in the directory server configuration:

- `cn=FIFO Entry Cache,cn=Monitor`
- `cn=Soft Reference Entry Cache,cn=Monitor`
- `cn=File System Entry Cache,cn=Monitor`

Additionally, any arbitrarily named active entry cache instance should provide a monitor, which can be accessed by that instance name, for example `cn=Any Arbitrary Name Entry Cache,cn=Monitor`.

Use the `ldapsearch` command with base DN `"cn=Entry Caches,cn=monitor"`.

```
$ ldapsearch -h localhost -p 4444 -D "cn=directory manager" -j pwd-file --useSSL \
  --trustAll -s base -b "cn=Entry Caches,cn=monitor" "(objectclass=*)"
```

Depending on your configuration, output will be similar to the following:

```
dn: cn=Entry Caches,cn=monitor
entryCacheHits: 0
entryCacheTries: 0
currentEntryCacheCount: 0
objectClass: extensibleObject
objectClass: top
objectClass: ds-monitor-entry
entryCacheHitRatio: 0
cn: Entry Caches
...
```

#### 29.5.1.27 To Monitor Network Groups

Use the `ldapsearch` command with the base DN `"cn=Network Groups,cn=monitor"`.

```
$ ldapsearch -h localhost -p 4444 -D "cn=directory manager" -j pwd-file \
--useSSL --trustAll -b "cn=Network Groups,cn=monitor" "(objectclass=*)"
```

Depending on your configuration, output will be similar to the following:

```
objectClass: top
objectClass: ds-monitor-entry
objectClass: ds-mon-branch
dn: cn=Network Groups,cn=monitor
dn: cn=admin,cn=Network Groups,cn=monitor
ds-mon-compare-operations-total-count: 0
ds-mon-failed-referrals-total-count: 15
ds-mon-unbind-operations-total-count: 13
ds-mon-followed-referrals-total-count: 34
ds-mon-violations-schema-total-count: Not implemented
ds-mon-bind-operations-total-count: 98
ds-mon-persistent-searchs-count: Not implemented
ds-mon-add-operations-total-count: 37
ds-mon-abandon-operations-total-count: 0
ds-mon-moddn-operations-total-count: 0
ds-mon-extended-operations-total-count: 0
ds-mon-searchsubtree-operations-total-count: 310
objectClass: top
objectClass: ds-monitor-entry
```

```
objectClass: extensibleObject
ds-mon-discarded-referrals-total-count: Not implemented
ds-mon-mod-operations-total-count: 1
ds-mon-forwarded-referrals-total-count: Not implemented
cn: admin
ds-mon-searchonelevel-operations-total-count: 92966
ds-mon-delete-operations-total-count: 0

dn: cn=default,cn=Network Groups,cn=monitor
...
```

### 29.5.1.28 To Monitor Distribution

Use the `ldapsearch` command with the base DN `"cn=Distribution,cn=monitor"`.

```
$ ldapsearch -h localhost -p 4444 -D "cn=directory manager" -j pwd-file \
  --useSSL --trustAll -b "cn=Distribution,cn=monitor" "(objectclass=*)"
```

Depending on your configuration, output will be similar to the following:

```
objectClass: top
objectClass: ds-monitor-entry
objectClass: ds-mon-branch
dn: cn=distribution,cn=monitor

cn: distrib-we
ds-mon-searchonelevel-operations-total-count: 0
ds-mon-residenttime-bind-operations-max-time: 0
...

ds-mon-delete-operations-total-count: 0

dn: cn=algorithm,cn=distrib-we,cn=distribution,cn=monitor
ds-mon-residenttime-total-time: 0
ds-mon-residenttime-max-time: 0
cn: algorithm
ds-mon-runs-total-count: 0
ds-mon-residenttime-min-time: 0
objectClass: top
objectClass: ds-monitor-entry
objectClass: extensibleObject

dn: cn=partitions,cn=algorithm,cn=distrib-we,cn=distribution,cn=monitor
objectClass: top
objectClass: ds-monitor-entry
objectClass: ds-mon-branch

dn: cn=distrib-part1,cn=partitions,cn=algorithm,cn=distrib-we,cn=distribution,cn
 =monitor
...
objectClass: top
objectClass: ds-monitor-entry
objectClass: extensibleObject
ds-mon-modify-operations-total-count: 0
cn: distrib-part1
ds-mon-searchonelevel-operations-total-count: 0
ds-mon-delete-operations-total-count: 0

dn: cn=distrib-part2,cn=partitions,cn=algorithm,cn=distrib-we,cn=distribution,cn
 =monitor
```

```
...
```

### 29.5.1.29 To Monitor Load Balancing

Use the `ldapsearch` command with the base DN `"cn=load balancing,cn=monitor"`.

```
$ ldapsearch -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file \
  --useSSL --trustAll -b "cn=load balancing,cn=monitor" "(objectclass=*)"
```

Depending on your configuration, output will be similar to the following:

```
objectClass: top
objectClass: ds-monitor-entry
objectClass: ds-mon-branch
dn: cn=load balancing,cn=monitor
dn: cn=load-bal-we1,cn=load balancing,cn=monitor
ds-mon-aborted-add-operations-total-count: 0
...
dn: cn=algorithm,cn=load-bal-we1,cn=load balancing,cn=monitor

dn: cn=routes,cn=algorithm,cn=load-bal-we1,cn=load balancing,cn=monitor
...
dn: cn=load-bal-route1,cn=routes,cn=algorithm,cn=load-bal-we1,cn=load
balancing,cn=monitor
...
dn: cn=load-bal-we2,cn=load balancing,cn=monitor
...
dn: cn=algorithm,cn=load-bal-we2,cn=load balancing,cn=monitor
...
dn: cn=routes,cn=algorithm,cn=load-bal-we2,cn=load balancing,cn=monitor

dn: cn=load-bal-route1,cn=routes,cn=algorithm,cn=load-bal-we2,cn=load
balancing,cn=monitor
...
cn: load-bal-route1

dn: cn=load-bal-route2,cn=routes,cn=algorithm,cn=load-bal-we1,cn=load
balancing,cn=monitor
...
cn: load-bal-route2

dn: cn=load-bal-route2,cn=routes,cn=algorithm,cn=load-bal-we2,cn=load
balancing,cn=monitor

cn: load-bal-route2
ds-mon-searchonelevel-operations-total-count: 9
ds-mon-delete-operations-total-count: 0
```

### 29.5.1.30 To Monitor Remote LDAP Servers

Use the `ldapsearch` command with the base DN `"cn=LDAP Servers,cn=monitor"`.

```
$ ldapsearch -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file \
  --useSSL --trustAll -b "cn=LDAP Servers,cn=monitor" "(objectclass=*)"
```

Depending on your configuration, output will be similar to the following:

```
objectClass: top
objectClass: ds-monitor-entry
```

```
objectClass: ds-mon-branch
dn: cn=LDAP Servers,cn=monitor

dn: cn=proxy1,cn=LDAP Servers,cn=monitor
ds-mon-aborted-add-operations-total-count: 0
...
cn: proxy1
ds-mon-searchonelevel-operations-total-count: 0
...
objectClass: top
objectClass: ds-monitor-entry
objectClass: extensibleObject

dn: cn=proxy2,cn=LDAP Servers,cn=monitor
ds-mon-aborted-add-operations-total-count: 0
...
cn: proxy2
ds-mon-searchonelevel-operations-total-count: 0
...
objectClass: top
objectClass: ds-monitor-entry
objectClass: extensibleObject
...
dn: cn=proxy3,cn=LDAP Servers,cn=monitor
...
cn: proxy3
ds-mon-searchonelevel-operations-total-count: 0
...
objectClass: top
objectClass: ds-monitor-entry
objectClass: extensibleObject
...
dn: cn=proxy4,cn=LDAP Servers,cn=monitor
...
cn: proxy4
...
objectClass: top
objectClass: ds-monitor-entry
objectClass: extensibleObject
```

#### 29.5.1.31 To Monitor a Global Index

Use the `ldapsearch` command with the base DN `"cn=givenname,cn=gi-catalog,cn=Global Index Catalogs,cn=monitor"`.

Ensure that `givenname` corresponds to the name of the indexed attribute (for example `cn`, if you indexed `cn`), and that `gi-catalog` corresponds to the name of the global index catalog.

```
$ ldapsearch -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file --useSSL \
  --trustAll -b "cn=givenname,cn=gi-catalog,cn=Global Index Catalogs,cn=monitor"
  "(objectclass=*)"
```

Depending on your configuration, output will be similar to the following:

```
dn: cn=givenname,cn=gi-catalog,cn=Global Index Catalogs,cn=monitor
ds-mon-add-operations-min-time: 0
ds-mon-add-operations-aborted-count: 0
ds-mon-lookup-operations-min-time: 0
ds-mon-getpartitions-operations-total-count: 0
```

```
ds-mon-add-operations-max-time: 0
ds-mon-lookup-operations-total-count: 0
ds-mon-memorized-remove-operations-count: 0
ds-mon-remove-operations-aborted-count: 0
ds-mon-add-operations-total-time: 0
ds-mon-getpartitions-operations-aborted-count: 0
ds-mon-lookup-operations-total-time: 0
ds-mon-index-entries: 0
ds-mon-remove-operations-failed-count: 0
ds-mon-getpartitions-operations-min-time: 0
ds-mon-lookup-operations-max-time: 0
ds-mon-getpartitions-operations-average-time: 0
ds-mon-index-creation-date: 1252483187019
ds-mon-getpartitions-operations-last-access-date: 0
ds-mon-remove-operations-total-count: 0
ds-mon-lookup-operations-failed-count: 0
ds-mon-add-operations-failed-count: 0
ds-mon-remove-operations-min-time: 0
ds-mon-add-operations-average-time: 0
ds-mon-lookup-operations-aborted-count: 0
ds-mon-getpartitions-operations-total-time: 0
ds-mon-remove-operations-max-time: 0
ds-mon-getpartitions-operations-max-time: 0
ds-mon-lookup-operations-last-access-date: 0
ds-mon-add-operations-total-count: 0
ds-mon-remove-operations-total-time: 0
ds-mon-remove-operations-average-time: 0
ds-mon-getpartitions-operations-failed-count: 0
objectClass: ds-monitor-entry
objectClass: top
objectClass: extensibleObject
ds-mon-lookup-operations-average-time: 0
ds-mon-remove-operations-last-access-date: 0
cn: givenname
ds-mon-add-operations-last-access-date: 0
```

#### 29.5.1.32 To Monitor a Global Index Catalog

Use the `ldapsearch` command with the base DN `"cn=gi-catalog,cn=Global Index Catalogs,cn=monitor"`.

Ensure that `givenname` corresponds to the name of the indexed attribute (for example cn, if you indexed cn), and that `gi-catalog` corresponds to the name of the global index catalog.

```
$ ldapsearch -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file --useSSL \
  --trustAll -b "cn=gi-catalog,cn=Global Index Catalogs,cn=monitor" \
  "(objectclass=*)"
```

Depending on your configuration, output will be similar to the following:

```
dn: cn=gi-catalog,cn=Global Index Catalogs,cn=monitor
ds-mon-replication-received-update-message-errors: 0
ds-mon-configured-index-number: 1
ds-mon-replication-full-update-pending-attribute:
ds-mon-replication-full-update-status: NONE
ds-mon-state: RUNNING_STANDALONE
ds-mon-replication-published-update-message-number: 0
ds-mon-replication-active: false
ds-mon-replication-auto-sync-retries: 0
ds-mon-replication-published-update-message-errors: 0
```

```
ds-mon-replication-full-update-errors: 0
ds-mon-replication-received-update-message-number: 0
ds-mon-replication-auto-sync-is-running: false
objectClass: ds-monitor-entry
objectClass: top
objectClass: extensibleObject
ds-mon-replication-configured: false
cn: gi-catalog
```

## 29.5.2 Monitoring Using the `manage-tasks` Command

Oracle Unified Directory provides a tasks back end that provides a mechanism for scheduling and processing certain tasks, such as `import-ldif`, `export-ldif`, `backup`, and `restore`. You can schedule a task to run at specific times and at recurring periods. To monitor scheduled tasks, use the `manage-tasks` command. For more information, see Section 14.4, "Configuring Commands As Tasks".

## 29.5.3 Monitoring the Server With JConsole

The JConsole (`jconsole`) Java utility is a JMX-compliant, graphical tool that connects to a running Java Virtual Machine that has been started with the management agent. This generic tool can be used to access server monitoring information.

### 29.5.3.1 To Configure JMX on a Server Instance

1. Start the server.
2. Enable the JMX Connection Handler and set the port number to be used with JMX.

   Choose a port that is not in use and to which the user that is running the server has access rights.

   ```
   $ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
     set-connection-handler-prop --handler-name "JMX Connection Handler" \
     --set enabled:true --set listen-port:1689
   ```

3. Add the JMX read, write, and notify privileges to the root DN.

   ```
   $ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -X -n \
     set-root-dn-prop \
     --add default-root-privilege-name:jmx-read \
     --add default-root-privilege-name:jmx-write \
     --add default-root-privilege-name:jmx-notify
   ```

4. Restart the server.

### 29.5.3.2 Starting JConsole

Start the console by typing `jconsole` in a terminal window.

To run `jconsole` from the command line, you might have to add *JAVA_HOME*/bin to your path, where `JAVA_HOME` is the directory containing the JDK. Alternatively, you can enter the full path when you type the command.

For more information about using JConsole, see Using JConsole (`http://download.oracle.com/docs/cd/E17409_01/javase/6/docs/technotes/guides/management/jconsole.html`).

#### 29.5.3.3 Accessing a Server Instance From JConsole

To connect JConsole to a server instance, use the Remote Process fields. The following fields are required:

- **JMX URL**:

  ```
  service:jmx:rmi:///jndi/rmi://''host'':''port''/org.opends.server.protocols.jmx.client-unknown
  ```

  - *host* is a host name, an IPv4 numeric host address, or an IPv6 numeric address enclosed in square brackets.
  - *port* is the decimal port number of the JMX connector. (See Section 29.4, "Configuring Alerts and Account Status Notification Handlers").

  The default JMX URL is:

  ```
  service:jmx:rmi:///jndi/rmi://198.51.100.0:1689/org.opends.server.protocols.jmx.client-unknown
  ```

- **User Name.** A valid LDAP user name.

  The default Directory Manager user name is `cn=Directory Manager`.

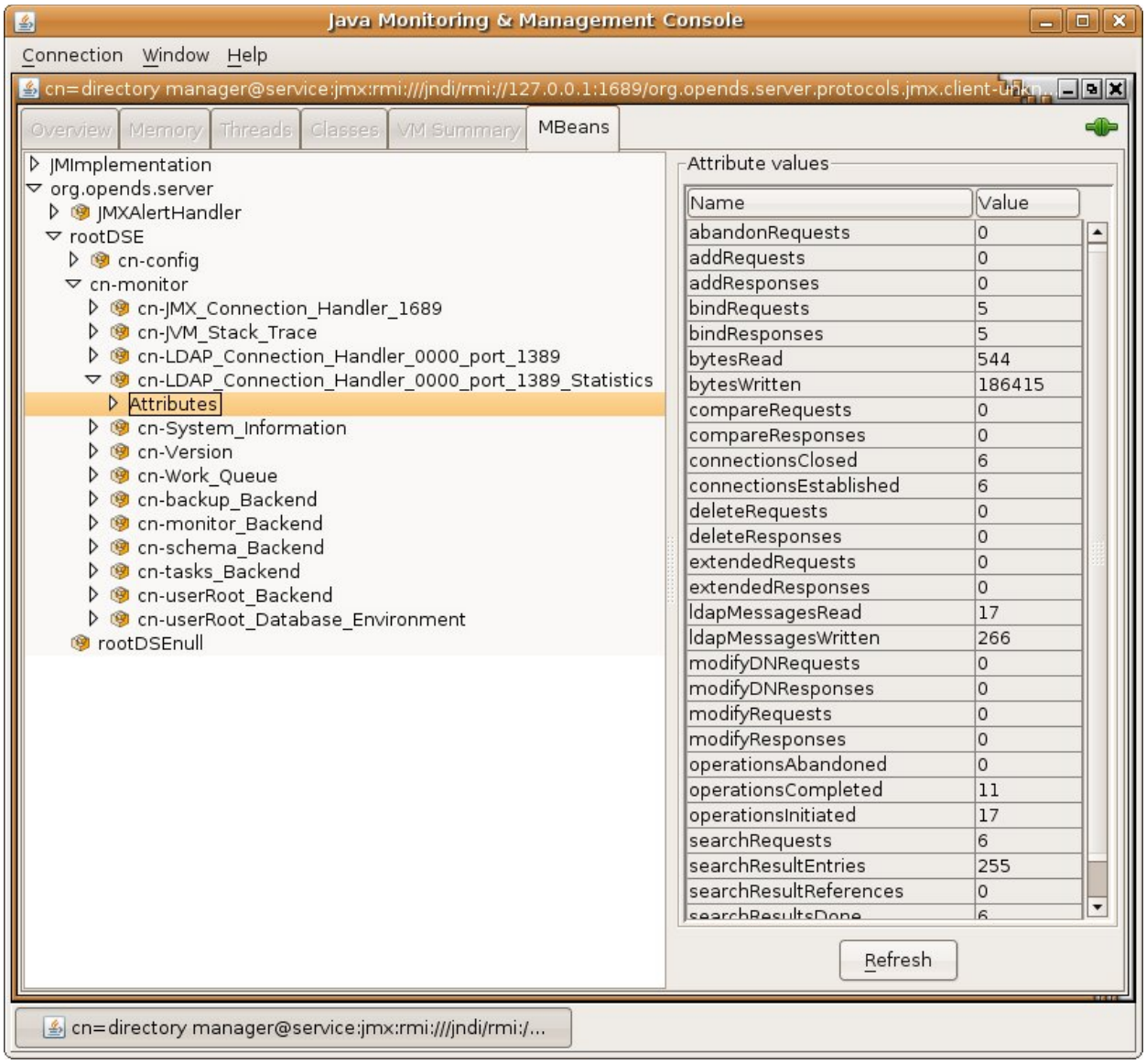
- **Password.** The user's LDAP password.

#### 29.5.3.4 Viewing Monitoring Information With JConsole

When JConsole is connected to a server instance, it displays management objects (MBeans). The tree on the left pane shows all MBeans currently available. You can access server monitoring information in the right hand pane by selecting the associated MBean.

The following figure shows the attribute list for a server `cn=LDAP Connection Handler 0.0.0.0 port 1389 Statistics,cn=monitor`.

***Figure 29–1 Java Monitoring and Management Console***




## 29.5.4 Accessing Logs

The server provides logging mechanisms to record access, error, or debugging information for the server instance. Multiple loggers of a given type can be active at any time, which makes it possible to create logs for specific subtrees or different

repositories. The server does not currently provide logging filters to restrict the type of information in the logs.

The following logs are provided:

- **Access logs**. Access logs record information about the types of operations processed by the directory server. Access logs are provided by default.
- **Audit logs**. Audit logs are a type of access log and record all activity on the directory server. Audit logs are not enabled by default.
- **Debug logs**. Debug logs record information that can be used for troubleshooting directory server problems or for providing detailed information about the directory server's processing. Debug logs are not enabled by default.
- **Error logs**. Error logs record all warnings, errors, or significant events that occur during directory server processing.
- **Replication repair logs**. Replication repair logs record inconsistencies on a single directory server in a topology.

  The replication repair log is read-only and its use is restricted to enabling replication conflict resolution.
- **oud-setup logs**. The setup logs record the equivalent command line arguments executed during the installation of an Oracle Unified Directory proxy server instance or replication gateway instance. This enables you to perform a "silent install" of the proxy server or gateway server, based on a previous installation.

  This file is not output for directory server instances.
- **server.out logs**. The server.out logs record the bootstrapping configuration process, list extensions loaded from jar files, and indicate connection and alert notification activity.

#### 29.5.4.1 To View the Access Logs

1. Change to the logs directory of the server instance.

   ```
   $ cd INSTANCE_DIR/OUD/logs
   ```

2. Open the `access` file by using a text editor or the UNIX `cat` command.

   ```
   $ cat access | more
   [10/Jan/2012:12:02:11 +0100] CONNECT conn=0 from=198.51.100.0:55416
   to=198.51.100.0:5444 protocol=LDAPS
   [10/Jan/2012:12:02:12 +0100] BIND REQ conn=0 op=0 msgID=1 type=SIMPLE
   dn="cn=Directory Manager"
   [10/Jan/2012:12:02:12 +0100] BIND RES conn=0 op=0 msgID=1 result=0
   authDN="cn=Directory Manager,cn=Root
    DNs,cn=config" etime=36
   [10/Jan/2012:12:02:12 +0100] UNBIND REQ conn=0 op=1 msgID=2
   [10/Jan/2012:12:02:12 +0100] DISCONNECT conn=0 reason="Client Disconnect"
   ...(more output)...
   ```

#### 29.5.4.2 To View the Audit Logs

1. If the audit log publisher is not already enabled, enable it as described in Section 29.3.1.1.2, "To Enable a Log Publisher".
2. Change to the logs directory of the server instance.

   ```
   $ cd INSTANCE_DIR/OUD/logs
   ```

3. Open the `audit` file by using a text editor or the UNIX `cat` command.

```
$ cat audit | more
# 11/Jan/2012:11:20:00 +0100; conn=10; op=18
dn: cn=File-Based Audit Logger,cn=Loggers,cn=config
changetype: modify
replace: ds-cfg-enabled
ds-cfg-enabled: true
-
replace: modifiersName
modifiersName: cn=directory manager
-
replace: modifyTimestamp
modifyTimestamp: 20120111102000Z

# 11/Jan/2012:11:20:20 +0100; conn=11; op=6
dn: cn=File-Based Debug Logger,cn=Loggers,cn=config
changetype: modify
replace: ds-cfg-enabled
ds-cfg-enabled: true
-
replace: modifiersName
modifiersName: cn=directory manager
-
replace: modifyTimestamp
modifyTimestamp: 20120111102020Z
...(more output)...
```

#### 29.5.4.3 To View the Debug Logs

1. If the debug log publisher is not already enabled, enable it as described in Section 29.3.1.1.2, "To Enable a Log Publisher".
2. Change to the logs directory of the server instance.

```
$ cd INSTANCE_DIR/OUD/logs
```

3. Open the `debug` file by using a text editor or the UNIX `cat` command.

```
$ cat debug | more
[11/Jan/2012:11:39:48 +0100] 0 caught error thread={Worker Thread 43(118)}
threadDetail={parentThread=main(1) isDaemon=false clientConnection=LDAP client
connection from 198.51.100.0:56288 to 198.51.100.0:2389
operation=SearchOperation(connID=13, opID=1, baseDN=dc=example,dc=com,
scope=wholeSubtree, filter=(objectclass=*)) } method={run(SearchOperationBas
is.java:1513)} caught={org.opends.server.types.CanceledOperationException:
Client Disconnect}
...(more output)...
```

#### 29.5.4.4 To View the Error Logs

1. Change to the logs directory of the server instance.

```
$ cd INSTANCE_DIR/OUD/logs
```

2. Open the `errors` file by using a text editor or the UNIX `cat` command.

```
$ cat errors | more
[11/Jan/2012:15:14:13 +0100] category=RUNTIME_INFORMATION severity=NOTICE
msgID=20381717 msg=Installation Directory:  /local/OUD_BASE/Oracle_OUD1
[11/Jan/2012:15:14:13 +0100] category=RUNTIME_INFORMATION severity=NOTICE
msgID=20381719 msg=Instance Directory:      /local/OUD_BASE/asinst_4/OUD
[11/Jan/2012:15:14:13 +0100] category=RUNTIME_INFORMATION severity=NOTICE
```

```
msgID=20381713 msg=JVM Information: 1.6.0_30-b12 by Sun Microsystems Inc.,
32-bit architecture, 957743104 bytes heap size
[11/Jan/2012:15:14:13 +0100] category=RUNTIME_INFORMATION severity=NOTICE
msgID=20381714 msg=JVM Host: host1, running SunOS 5.10 sparc, 103079215104
bytes physical memory size, number of processors available 24
[11/Jan/2012:15:14:13 +0100] category=RUNTIME_INFORMATION severity=NOTICE
msgID=20381715 msg=JVM Arguments: "-Dorg.opends.server.scriptName=start-ds"
[11/Jan/2012:15:14:16 +0100] category=PROTOCOL severity=NOTICE msgID=2556180
msg=Started listening for new connections on Administration Connector 0.0.0.0
port 7444
[11/Jan/2012:15:14:16 +0100] category=PROTOCOL severity=NOTICE msgID=2556180
msg=Started listening for new connections on LDAP Connection Handler 0.0.0.0
port 4389
[11/Jan/2012:15:14:16 +0100] category=CORE severity=NOTICE msgID=458887 msg=The
Directory Server has started successfully
...(more output)...
```

#### 29.5.4.5 To View the Replication Repair Logs

1. Change to the logs directory of the server instance.

   ```
   $ cd INSTANCE_DIR/OUD/logs
   ```

2. Open the `replication` file by using a text editor or the UNIX `cat` command.

   ```
   $ cat replication | more
   [13/Jan/2012:15:00:50 +0100] category=SYNC severity=NOTICE msgID=15139035
   msg=The replication server database has version 2 format
   [13/Jan/2012:15:00:50 +0100] category=SYNC severity=NOTICE msgID=15138878
   msg=Replication is up and running for domain cn=admin data with replicati
   on server id 18049 host1/198.51.100.0:8989 - local server id is 9338 - data
   generation is 93408
   [13/Jan/2012:15:00:52 +0100] category=SYNC severity=NOTICE msgID=15138878
   msg=Replication is up and running for domain dc=example,dc=com with repli
   cation server id 18049 host1/198.51.100.0:8989 - local server id is 25340 -
   data generation is 19449577
   [13/Jan/2012:15:00:53 +0100] category=SYNC severity=NOTICE msgID=15138878
   msg=Replication is up and running for domain cn=schema with replication s
   erver id 18049 host1/198.51.100.0:8989 - local server id is 13881 - data
   generation is 8408
   [13/Jan/2012:15:08:28 +0100] category=SYNC severity=NOTICE msgID=15138893
   msg=On suffix cn=admin data, replication server 3844 presented generation
    ID=-1 when expected generation ID=93408
   [13/Jan/2012:15:08:28 +0100] category=SYNC severity=MILD_ERROR msgID=14876753
   msg=In RS 18049 for dn cn=admin data, update 00000134d765d4b1247a0000
   0001 will not be sent to RS 3844 with generation id -1 different from local
   generation id 93408
   [13/Jan/2012:15:08:28 +0100] category=SYNC severity=MILD_ERROR msgID=14876753
   msg=In RS 18049 for dn cn=admin data, update 00000134d765d4b1247a0000
   0002 will not be sent to RS 3844 with generation id -1 different from local
   generation id 93408
   ...(more output)...
   ```

#### 29.5.4.6 To View the server.out Logs

1. Change to the logs directory of the server instance.

   ```
   $ cd INSTANCE_DIR/OUD/logs
   ```

2. Open the `server.out` file by using a text editor or the UNIX `cat` command.

   ```
   $ cat server.out | more
   ```

```
[23/May/2011:02:27:59 -0700] category=CORE severity=INFORMATION msgID=132
  msg=The Directory Server is beginning the configuration bootstrapping process
[23/May/2011:02:28:00 -0700] category=EXTENSIONS severity=INFORMATION
msgID=1049147
  msg=Loaded extension from file
'/OUD_BASE/ORACLE_HOME/lib/extensions/globalindex.jar'
  (build 1.0.0)
[23/May/2011:02:28:00 -0700] category=EXTENSIONS severity=INFORMATION
msgID=1049147
  msg=Loaded extension from file
'/OUD_BASE/ORACLE_HOME/lib/extensions/replication-gateway.jar'
  (build 1.0.0)
[23/May/2011:02:28:00 -0700] category=EXTENSIONS severity=INFORMATION
msgID=1049147
  msg=Loaded extension from file
'/OUD_BASE/ORACLE_HOME/lib/extensions/snmp-mib2605.jar'
  (build 11.1.1.5.0)
[23/May/2011:02:28:00 -0700] category=EXTENSIONS severity=INFORMATION
msgID=1049147
  msg=Loaded extension from file
'/OUD_BASE/ORACLE_HOME/lib/extensions/loadbalancing.jar'
  (build 1.0.0)
[23/May/2011:02:28:00 -0700] category=EXTENSIONS severity=INFORMATION
msgID=1049147
  msg=Loaded extension from file
'/OUD_BASE/ORACLE_HOME/lib/extensions/virtualization.jar'
  (build 1.0.0)
[23/May/2011:02:28:00 -0700] category=EXTENSIONS severity=INFORMATION
msgID=1049147
  msg=Loaded extension from file
'/OUD_BASE/ORACLE_HOME/lib/extensions/distribution.jar'
  (build 1.0.0)
...
more output
...
```

#### 29.5.4.7 To View the Setup Logs

This log file is available for proxy server and replication gateway instances only.

1. Change to the logs directory of the server instance.

   ```
   $ cd INSTANCE_DIR/OUD/logs
   ```

2. Open the `oud-setup` file by using a text editor or the UNIX `cat` command.

   ```
   $ cat oud-setup | more
   May 19, 2011 2:24:36 AM com.sun.dps.ui.deploy.SetupLog initLogFileHandler
   INFO: oud-setup application launched May 19, 2011 2:24:36 AM PDT
   ...(more output)...
   ```

## 29.6 Monitoring the Server With SNMP

Oracle Unified Directory provides a jar file extension that contains a Simple Network Management Protocol (SNMP) connection handler for Management Information Base (MIB) 2605 support. The extension contains the SNMP connection handler, the required classes to support MIB 2605 objects and SNMP requests, and the SNMP adapter that allows an SNMP manager to access the server monitoring information.

Before you start on the procedures in this section, ensure that you have set up an SNMP-managed network for your particular system.

## 29.6.1 Configuring the SNMP Connection Handler and Its Dependencies

Oracle Unified Directory provides an SNMP connection handler that you can enable and configure. The SNMP connection handler is provided as a jar file extension and is located in `install-dir/lib/extensions/snmp-mib2605.jar`.

### 29.6.1.1 To Configure SNMP in the Server

Oracle Unified Directory can be configured for monitoring through the Simple Network Management Protocol (SNMP). The server uses the Java Dynamic Management Kit (JDMK) to create smart agents for the SNMP connection handler.

1. Verify that the SNMP connection handler is displayed under the list of current connection handlers by using `dsconfig` as follows.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -n \
  list-connection-handlers

Connection Handler       : Type : enabled : listen-port : use-ssl
-------------------------:------:---------:-------------:--------
JMX Connection Handler   : jmx  : false   : 1689        : false
LDAP Connection Handler  : ldap : true    : 1389        : false
LDAPS Connection Handler : ldap : false   : 636         : true
LDIF Connection Handler  : ldif : true    : -           : -
SNMP Connection Handler  : snmp : false   : 161         : -
```

2. Use the `dsconfig` command to enable SNMP for the server and to set the listen port.

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j pwd-file -n -X \
  set-connection-handler-prop --handler-name "SNMP Connection Handler" \
  --set enabled:true --set listen-port:8085
```

### 29.6.1.2 To View the SNMP Connection Handler Properties

Run the following command to display the list of SNMP connection handler properties.

```
$ dsconfig -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -n \
  get-connection-handler-prop --handler-name "SNMP Connection Handler"
```

The connection handler properties are listed with their values, as follows.

```
Property            : Value(s)
--------------------:-----------------------------------------
allowed-client      : -
allowed-manager     : *
allowed-user        : *
community           : OUD
denied-client       : -
enabled             : false
listen-port         : 161
opendmk-jarfile     : -
registered-mbean    : false
security-agent-file : config/snmp/security/oud-snmp.security
security-level      : authnopriv
trap-port           : 162
traps-community     : OUD
traps-destination   : -
```

#### 29.6.1.3 To Access SNMP on a Server Instance

1. Restart the server by using `stop-ds` and `start-ds`.

   If the server was started and no modifications were made to the configuration, the restart operation is not required.

2. Check that the SNMP Connection Handler is up and running.

```
$ snmpwalk -v 2c -c OUD@OUD localhost:8085 mib-2.66
SNMPv2-SMI::mib-2.66.1.1.1.1 = STRING: "Oracle Unified Directory Server
11.1.1.5.0 -
  20090310152800Z"
SNMPv2-SMI::mib-2.66.1.1.2.1 = STRING: "INSTANCE_DIR/bin"
SNMPv2-SMI::mib-2.66.1.1.3.1 = Gauge32: 35
SNMPv2-SMI::mib-2.66.1.1.4.1 = Gauge32: 1
SNMPv2-SMI::mib-2.66.1.1.5.1 = Gauge32: 0
SNMPv2-SMI::mib-2.66.1.1.6.1 = Counter32: 0
SNMPv2-SMI::mib-2.66.1.1.7.1 = Counter32: 1
SNMPv2-SMI::mib-2.66.2.1.1.1.1 = INTEGER: 1
SNMPv2-SMI::mib-2.66.2.1.1.1.2 = INTEGER: 2
SNMPv2-SMI::mib-2.66.2.1.1.1.3 = INTEGER: 3
SNMPv2-SMI::mib-2.66.2.1.2.1.1 = OID: SNMPv2-SMI::internet.27.3.8085
SNMPv2-SMI::mib-2.66.2.1.2.1.2 = OID: SNMPv2-SMI::internet.27.3.1389
SNMPv2-SMI::mib-2.66.2.1.2.1.3 = OID: SNMPv2-SMI::enterprises.42
SNMPv2-SMI::mib-2.66.2.1.3.1.1 = Counter32: 1
SNMPv2-SMI::mib-2.66.2.1.3.1.2 = Counter32: 1
SNMPv2-SMI::mib-2.66.2.1.3.1.3 = Counter32: 1
SNMPv2-SMI::mib-2.66.2.1.4.1.1 = Counter32: 1
SNMPv2-SMI::mib-2.66.2.1.4.1.2 = Counter32: 1
SNMPv2-SMI::mib-2.66.2.1.4.1.3 = Counter32: 1
SNMPv2-SMI::mib-2.66.2.1.5.1.1 = Counter32: 1
SNMPv2-SMI::mib-2.66.2.1.5.1.2 = Counter32: 1
...
```

   The managed objects included in the MIB 2605 are divided into three tables: `dsTable`, `dsAppliIfOpsTable`, and `dsIntTable`. Currently, the `dsIntTable` table is not implemented.

#### 29.6.1.4 SNMP Security Configuration

SNMP security configuration depends on the version of SNMP as you are using. This topic discusses security configuration for SNMP V1 and V2c, and V3.

**29.6.1.4.1 SNMP Security Configuration: V1 and V2c** Under SNMP v1 and SNMP v2c, agents act as information servers, and the IP-based access control protects this information from unauthorized access. By default, the MIB 2605 is accessible in v1 and v2c by using the community string `OUD@OUD`. All managers are allowed to read the monitoring information exposed by the MIB 2605.

> **Note:** Only read access is authorized on the MIB 2605.

You can configure SNMP v1 and SNMP v2c by setting the SNMP connection handler properties with the `dsconfig` command. Properties related to the SNMP v1 and SNMP v2c security configuration include:

- `allowed-manager`
- `community`

SNMP v1 traps are sent on server startup and server shutdown. By default, these traps are sent to `localhost` and use the trap community string "OUD".

> **Note:** The default trap port might have to be changed to a value that is allowed by the system.

SNMP traps are also configured by setting the SNMP connection properties with the `dsconfig` command. Properties related to SNMP traps include:

- `trap-port`
- `traps-community`
- `traps-destination`

The ACL file that corresponds to the default values of the SNMP connection handler would be represented as follows:

```
acl = {
{
communities = OUD
access = read-only
managers = all
}
}
trap = {
{
traps-community = OUD
hosts = localhost
}
}
```

**29.6.1.4.2 SNMP Security Configuration: V3** The SNMP v3 protocol provides more sophisticated security mechanisms than SNMP v1 and SNMP v2c. SNMP v3 implements a user-based security model (USM) that authenticates and encrypts the requests sent between agents and their managers, and provides user-based access control. A `defaultUser` template is provided for adding authorized users in the agent engine using the SNMP cloning mechanism.

Under SNMP v3, the community string described in the previous section is used as the "context" from which the MIB 2605 is registered. By default, the MIB2605 is accessible in v3 by using the context "OUD". All users have access to it.

The SNMP v3 UACL is configured by setting the SNMP connection handler properties with the `dsconfig` command-line utility. The properties related to SNMP v3 UACL configuration include:

- `community`
- `allowed-user`
- `security-level`

The UACL file corresponding to the default values of the SNMP connection handler would be represented as follows:

```
uacl = {
{
context-names = OUD
access = read-only
security-level = authNoPriv
users = *
```

```
}
}
```

**29.6.1.4.3 SNMP USM Configuration: V3** The USM MIB (that is, the MIB that defines allowed users) is registered in the `null` context and only a `snmpAdmin` user with a security level `authNoPriv` has `read-write` access to it. This `snmpAdmin` user can add additional users who can access the MIB 2605 information.

The SNMP v3 USM configuration is read from a template file that is located at `INSTANCE_DIR/OUD/config/snmp/security/oud-snmp.security`. The template file is not encrypted.

To access the MIB 2605 in the server agent, use the SNMP clone mechanism to add a user in the security file. Use `snmpAdmin` to send the SNMP request for the clone mechanism as shown here. The user to clone is `defaultUser`. The `snmpAdmin` and `defaultUser` users cannot access the MIB 2605 information.

- Admin User to add and configure other users.

  ```
  userEntry=localEngineID,snmpAdmin,null,usmHMACMD5AuthProtocol,passadmin
  ```

- Template user to be cloned with no read or write access.

  ```
  userEntry=localEngineID,defaultUser,,usmHMACMD5AuthProtocol,password,,3,true
  ```

**Note:** The security file is also used to make the users persistent.

# 29.7 Monitoring a Replicated Topology

These topics describe how to monitor a replicated topology by using the `dsreplication status` command, and how to use the `ldapsearch` command to obtain more advanced monitoring information.

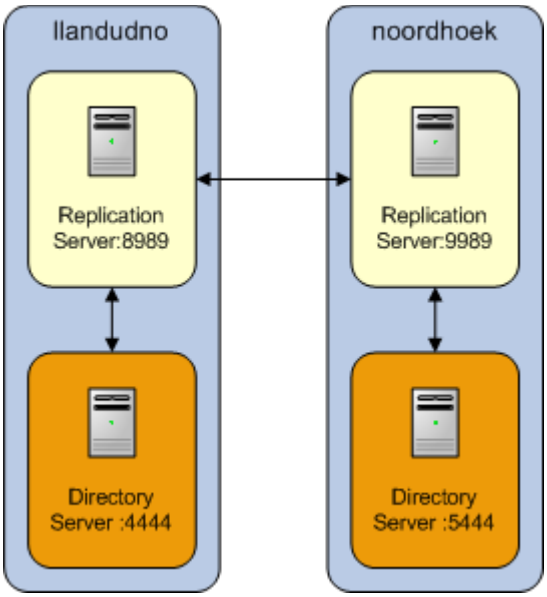
## 29.7.1 Monitoring Replication Status With `dsreplication`

The simplest way to monitor replication is to use the `dsreplication status` command. This command provides a tabular view of the replication status, including the following information:

- The topology and its connections
- The latency between replicated servers
- The data consistency across replicated servers
- The security configuration between replicated servers
- The replication protocol peer to peer

The examples in the remainder of this section assume the following simple replication topology.

***Figure 29–2 Simple Replication Topology***




To obtain the replication status, run the following command:

```
$ dsreplication status --adminUID admin --adminPasswordFile pwd.txt -X \
 --hostname host1 --port 4444
```

The output of this command includes the following:

- **Server.** Lists the LDAP servers in the topology and the port on which they are listening for LDAP connections.
- **Entries.** Indicates the number of entries on each server for the specified base DN. If the information in this column is not the same across all the servers, the replication topology is not synchronized.
- **M.C.** Indicates the number of updates already pushed by the other LDAP servers in the topology, but not yet replayed on the specified LDAP server. If this number is high on a particular server, investigate the latency of that server.
- **A.O.M.C.** Specifies the approximate date of the oldest update pushed by the other directory servers in the topology, but not yet processed on the specified LDAP server.
- **Port.** Indicates the port of the replication server to which the specified LDAP server is directly connected.
- **Encryption.** Indicates whether SSL encryption is enabled between the LDAP server and its replication server.
- **Trust.** Indicates whether this server is configured as a trusted or untrusted server. For more information, see Section 26.10, "Using Isolated Replicas".
- **U.C.** Specifies the number of changes that have been made on an untrusted server, and not yet replicated to the topology. For more information, see Section 26.10, "Using Isolated Replicas".
- **Status.** Indicates the status of the replication domain on this directory server. The status can be one of the following:

- **Normal.** The connection to a replication server is established with the correct data set. Replication is working. If assured mode is used, then acknowledgements from this directory server are sent.
- **Degraded.** The connection to a replication server is established with the correct data set. Replication is working in degraded mode as the directory server has numerous changes that are pending in the replication server queue. If assured mode is used, then acknowledgements from this directory server are not expected.
- **Full Update.** The connection to a replication server is established and a new data set is received from this connection (online import), to initialize the local back end.
- **Bad Data Set.** The connection to a replication server is established with a data set that is different from the rest of the topology. Replication is not working. Either the other directory servers of the topology should be initialized with a compatible data set, or this server should be initialized with another data set that is compatible with the other servers.
- **Not Connected.** The directory server is not connected to any replication server.

- **Change Log**. Indicates whether the external change log is enabled for the base DN on this server. For more information, see Section 26.5, "Using the External Change Log".
- **Group ID**. The ID of the replication group to which the server belongs. For more information, see Section 6.6, "Replication Groups".
- **Connected To.** Displays the name, IP address and replication port of the replication server to which this directory server is connected.

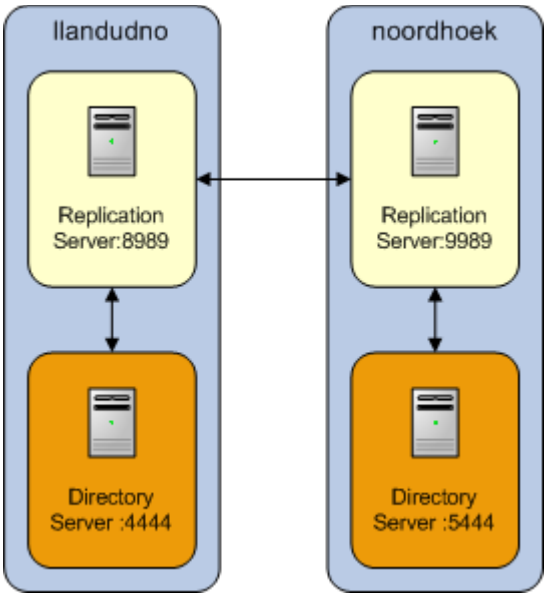
Additional replication monitoring information is available under the `cn=monitor` entry. You can use the `ldapsearch` command to track specific monitoring attributes, which will provide you with a comprehensive view of the replication status. For more information, see Section 29.7.2, "Advanced Replication Monitoring".

## 29.7.2 Advanced Replication Monitoring

The easiest way to monitor replication status is by using the `dsreplication status` command. However, in depth replication monitoring information is available under the `cn=monitor` entry. You can use the `ldapsearch` command to track specific monitoring attributes, which provide you with a comprehensive view of the replication status. Monitoring information is consolidated by replication servers. Therefore, monitoring information can only be retrieved by searching a directory server that hosts a running replication server.

The examples in the remainder of this section assume the following simple replication topology.

*Figure 29–3 Simple Replication Topology*




These examples access the `cn=monitor` entry on the administration port over SSL (`--useSSL`) and automatically trust the certificate that is presented by the server (`--trustAll`).

The information under `cn=monitor` can be filtered to include a single replicated base DN. You can do this in two ways:

- Specify the `domain-name` attribute as a filter, for example:

```
$ ldapsearch -p 4444 -D "cn=directory manager" -j pwd-file --useSSL \
  --trustAll -b "cn=monitor" "(domain-name=dc=example,dc=com)"
```

- Include the base DN in the search base, for example:

```
$ ldapsearch -p 4444 -D "cn=directory manager" -j pwd-file --useSSL \
  --trustAll -b "cn=dc_example_dc_com,cn=replication,cn=monitor" \
  "(objectclass=*)"
```

This section covers the following monitoring topics:

- Section 29.7.2.1, "To Monitor the Topology and Its Connections"
- Section 29.7.2.2, "To Monitor Replication Latency"
- Section 29.7.2.3, "To Monitor Data Consistency"
- Section 29.7.2.4, "To Monitor Replication Security"
- Section 29.7.2.5, "To Monitor Replicated Updates"
- Section 29.7.2.6, "To Monitor Replication Conflicts"

### 29.7.2.1 To Monitor the Topology and Its Connections

Each directory server contains a list of candidate replication servers for each replicated base DN. However, a directory server is *connected* to only one replication server at a time.

To obtain an overview of the replication topology and its connections, run the following search on any directory server in the topology that hosts a replication server:

```
$ ldapsearch -p 4444 -D "cn=directory manager" -j pwd-file --useSSL \
  --trustAll -b "cn=monitor" "(connected-to=*)" "connected-to" "lost-connections"
dn: cn=Replication Domain 30839,cn=dc_example_dc_com,cn=replication,cn=monitor
lost-connections: 0
connected-to: llandudno/0:0:0:0:0:0:0:1:8989

dn: cn=Replication Domain 14142,cn=cn_schema,cn=replication,cn=monitor
lost-connections: 0
connected-to: llandudno/0:0:0:0:0:0:0:1:8989

dn: cn=Connected Replica llandudno 27742,cn=Replication Server 8989  1740,cn=cn_
admin data,cn=replication,cn=monitor
connected-to: Replication Server 8989 1740

dn: cn=Connected Replica llandudno 30839,cn=Replication Server 8989  1740,cn=dc_
example_dc_com,cn=replication,cn=monitor
connected-to: Replication Server 8989 1740

dn: cn=Connected Replica llandudno 14142,cn=Replication Server 8989  1740,cn=cn_
schema,cn=replication,cn=monitor
connected-to: Replication Server 8989 1740

dn: cn=Undirect Replica  22052,cn=Connected Replication Server noordhoek:9989 71
64,cn=Replication Server 8989  1740,cn=cn_schema,cn=replication,cn=monitor
connected-to: Connected Replication Server noordhoek:9989 7164,cn=Replication Se
rver 8989  1740,cn=cn_schema,cn=replication

dn: cn=Undirect Replica  19984,cn=Connected Replication Server noordhoek:9989 71
64,cn=Replication Server 8989  1740,cn=dc_example_dc_com,cn=replication,cn=moni
tor
connected-to: Connected Replication Server noordhoek:9989 7164,cn=Replication Se
rver 8989  1740,cn=dc_example_dc_com,cn=replication

dn: cn=Undirect Replica  30030,cn=Connected Replication Server noordhoek:9989 71
64,cn=Replication Server 8989  1740,cn=cn_admin data,cn=replication,cn=monitor
connected-to: Connected Replication Server noordhoek:9989 7164,cn=Replication Se
rver 8989  1740,cn=cn_admin data,cn=replication

dn: cn=Replication Domain 27742,cn=cn_admin data,cn=replication,cn=monitor
lost-connections: 0
connected-to: llandudno/0:0:0:0:0:0:0:1:8989
```

The `connected-to` attribute specifies the replication server to which each directory server is currently connected for a particular base DN. If a directory server is directly connected to the replication server, its DN includes `cn=Connected Replica`. A directory server that is in the topology but is connected to a different replication server has `cn=Undirect Replica` in its DN. Because all replication servers are permanently connected to all other replication servers, the `connected-to` attribute does not exist for replication servers.

The `lost-connections` attribute indicates the number of connection breaks between directory servers and replication servers. The value of this attribute on each directory server should be close to the number of times that replication has been stopped on that server. If the value of this attribute is much higher, there are unexpected connection losses that must be investigated.

#### 29.7.2.2 To Monitor Replication Latency

Monitoring replication latency enables you to establish whether a specific replication server is lagging behind other servers in the topology. This provides a complete view of any replication delays and the current quality of service.

To monitor replication latency, run the following search on any server in the topology that hosts a replication server:

```
$ ldapsearch -p 4444 -D "cn=directory manager" -j pwd-file --useSSL \
  --trustAll -b "cn=monitor" "domain-name=dc=example,dc=com" "missing-changes" \
  "approx-older-change-not-synchronized"
dn: cn=Replication Domain 30839,cn=dc_example_dc_com,cn=replication,cn=monitor

dn: cn=Replication Server 8989 1740,cn=dc_example_dc_com,cn=replication,cn=monitor
missing-changes: 0

dn: cn=Connected Replica llandudno 30839,cn=Replication Server 8989  1740,cn=dc_
example_dc_com,cn=replication,cn=monitor
missing-changes: 0

dn: cn=Connected Replication Server noordhoek:9989 7164,cn=Replication Server 8989
  1740,cn=dc_example_dc_com,cn=replication,cn=monitor
missing-changes: 0

dn: cn=Undirect Replica 19984,cn=Connected Replication Server noordhoek:9989
7164,cn=Replication Server 8989
1740,cn=dc_example_dc_com,cn=replication,cn=monitor
missing-changes: 0
```

The `missing-changes` attribute specifies the number of updates already pushed by the other directory servers in the topology, but not yet replayed on the specified directory server.

The `approx-older-change-not-synchronized` attribute specifies the approximate date of the oldest update pushed by the other directory servers in the topology, but not yet processed on the specified directory server.

> **Note:** If the replication latency, as defined by these attributes, is high, look at the number of updates sent and received to identify the servers in the topology that are causing the latency. These attributes are described later in this document.

#### 29.7.2.3 To Monitor Data Consistency

Monitoring data consistency enables you to establish whether each replication server in the topology is synchronized and up-to-date with the latest changes that have occurred in the topology.

To monitor the data consistency across the directory servers in the topology, run the following search on any server in the topology that hosts a replication server:

```
$ ldapsearch -p 4444 -D "cn=directory manager" -j pwd-file --useSSL --trustAll \
  -b "cn=monitor" "(generation-id=*)" "generation-id"
dn: cn=Replication Server 8989  1740,cn=cn_admin data,cn=replication,cn=monitor
generation-id: cn=admin data 94310

dn: cn=Connected Replication Server noordhoek:9989 7164,cn=Replication Server 8989
  1740,cn=cn_admin data,cn=replication,cn=monitor
generation-id: 94310
```

```
dn: cn=Replication Domain 30839,cn=dc_example_dc_com,cn=replication,cn=monitor
generation-id: 19399981

dn: cn=Replication Domain 14142,cn=cn_schema,cn=replication,cn=monitor
generation-id: 8468

dn: cn=Connected Replica llandudno 27742,cn=Replication Server 8989  1740,cn=cn_
admin data,cn=replication,cn=monitor
generation-id: 94310

dn: cn=Replication Server 8989  1740,cn=cn_schema,cn=replication,cn=monitor
generation-id: cn=schema 8468

dn: cn=Replication Server 8989
1740,cn=dc_example_dc_com,cn=replication,cn=monitor
generation-id: dc=example,dc=com 19399981

dn: cn=Connected Replica llandudno 30839,cn=Replication Server 8989  1740,cn=dc_
example_dc_com,cn=replication,cn=monitor
generation-id: 19399981

dn: cn=Connected Replication Server noordhoek:9989 7164,cn=Replication Server 8989
  1740,cn=cn_schema,cn=replication,cn=monitor
generation-id: 8468

dn: cn=Connected Replica llandudno 14142,cn=Replication Server 8989  1740,cn=cn_
schema,cn=replication,cn=monitor
generation-id: 8468

dn: cn=Connected Replication Server noordhoek:9989 7164,cn=Replication Server 8989
  1740,cn=dc_example_dc_com,cn=replication,cn=monitor
generation-id: 19399981

dn: cn=Replication Domain 27742,cn=cn_admin data,cn=replication,cn=monitor
generation-id: 94310
```

The `generation-id` attribute indicates the *version* of the data in each replicated base DN, for each directory server. Note that the generation ID on all servers for the base DN `dc=example,dc=com` is `19399981`. The consistency of the generation IDs means that the data on those servers is the same for that base DN.

Each directory server is also aware of the generation ID of the replication server to which it is connected. The generation ID of a replication server relates to the updates that are stored in its change log database for that base DN.

Replication is considered to be working correctly between two directory servers, for a specified base DN, when those servers and their replication server all have the same generation ID.

#### 29.7.2.4 To Monitor Replication Security

A secure replication topology has SSL encryption enabled between servers, for a particular base DN.

To monitor replication security, run the following search on any server in the topology that hosts a replication server:

```
$ ldapsearch -p 4444 -D "cn=directory manager" -j pwd-file --useSSL --trustAll \
  -b "cn=monitor" "(ssl-encryption=*)" "ssl-encryption"
dn: cn=Connected Replication Server noordhoek:9989 7164,cn=Replication Server 89
```

```
89  1740,cn=cn_admin data,cn=replication,cn=monitor
ssl-encryption: true

dn: cn=Replication Domain 30839,cn=dc_example_dc_com,cn=replication,cn=monitor
ssl-encryption: true

dn: cn=Replication Domain 14142,cn=cn_schema,cn=replication,cn=monitor
ssl-encryption: true

dn: cn=Connected Replica llandudno 27742,cn=Replication Server 8989  1740,cn=cn_
admin data,cn=replication,cn=monitor
ssl-encryption: true

dn: cn=Connected Replica llandudno 30839,cn=Replication Server 8989  1740,cn=dc_
example_dc_com,cn=replication,cn=monitor
ssl-encryption: true

dn: cn=Connected Replication Server noordhoek:9989 7164,cn=Replication Server 89
89  1740,cn=cn_schema,cn=replication,cn=monitor
ssl-encryption: true

dn: cn=Connected Replica llandudno 14142,cn=Replication Server 8989  1740,cn=cn_
schema,cn=replication,cn=monitor
ssl-encryption: true

dn: cn=Connected Replication Server noordhoek:9989 7164,cn=Replication Server 8989
  1740,cn=dc_example_dc_com,cn=replication,cn=monitor
ssl-encryption: true

dn: cn=Replication Domain 27742,cn=cn_admin data,cn=replication,cn=monitor
ssl-encryption: true
```

The `ssl-encryption` attribute specifies whether the replication protocol is encrypted between two servers for a specified base DN. This information is available for each directory server or replication server. Authentication of replication sessions is not monitored.

### 29.7.2.5 To Monitor Replicated Updates

Monitoring the number of updates that have been sent and received by the servers in a topology provides an indication of how well replication is working.

To monitor sent and received updates, type the following command:

```
$ ldapsearch -p 4444 -D "cn=directory manager" -j pwd-file --useSSL --trustAll \
  -b "cn=monitor" "(&(sent-updates=*)(received-updates=*))" \
  "sent-updates" "received-updates"
dn: cn=Connected Replication Server noordhoek:9989 7164,cn=Replication Server 8989
  1740,cn=cn_admin data,cn=replication,cn=monitor
sent-updates: 7
received-updates: 0

dn: cn=Replication Domain 30839,cn=dc_example_dc_com,cn=replication,cn=monitor
received-updates: 28
sent-updates: 0

dn: cn=Replication Domain 14142,cn=cn_schema,cn=replication,cn=monitor
received-updates: 0
sent-updates: 0

dn: cn=Connected Replica llandudno 27742,cn=Replication Server 8989  1740,cn=cn_
```

```
admin data,cn=replication,cn=monitor
sent-updates: 0
received-updates: 0

dn: cn=Connected Replica llandudno 30839,cn=Replication Server 8989  1740,cn=dc_
example_dc_com,cn=replication,cn=monitor
sent-updates: 28
received-updates: 0

dn: cn=Connected Replication Server noordhoek:9989 7164,cn=Replication Server 8989
  1740,cn=cn_schema,cn=replication,cn=monitor
sent-updates: 0
received-updates: 0

dn: cn=Connected Replica llandudno 14142,cn=Replication Server 8989  1740,cn=cn_
schema,cn=replication,cn=monitor
sent-updates: 0
received-updates: 0

dn: cn=Connected Replication Server noordhoek:9989 7164,cn=Replication Server 8989
  1740,cn=dc_example_dc_com,cn=replication,cn=monitor
sent-updates: 0
received-updates: 28

dn: cn=Replication Domain 27742,cn=cn_admin data,cn=replication,cn=monitor
received-updates: 0
sent-updates: 0
```

The `sent-updates` attribute indicates the number of updates that have been sent by this directory server or replication server.

The `received-updates` attribute indicates the number of updates that have been received by this directory server or replication server.

The values of these attributes assist in determining the flow of updates within a topology. When replication appears to be very slow, it is helpful to monitor these attributes. If the number of updates sent by one server is consistently much higher than the number of updates received by another server, it is likely that the second server is a bottleneck in the topology.

The replication protocol controls the flow of updates between two servers. This ensures that when a high number of updates is exchanged between two servers, the servers are not prevented from processing operations with a higher priority. This functionality relies on a window mechanism where the recipient server periodically provides the sending server with the number of updates that the sending server can send.

You can specify the size of the send and receive windows, by setting the `max-send-window` and `max-rcv-window` configuration attributes. For more information, see Section 26.3, "Modifying the Replication Configuration With `dsconfig`".

The `current-send-window` monitoring attribute indicates how many changes can be sent by the sending server to the recipient server at that specific time. If the value of the `current-send-window` attribute is often equal to 0, transmission is stopped and the recipient server is probably a bottleneck in the topology. If the value of the `current-send-window` attribute is often equal to the value of the `max-send-window` attribute, and you are experiencing high replication latency, it is likely that the sending server is a bottleneck in the topology.

To obtain the value of the `current-send-window` property, type the following command:

```
$ ldapsearch -p 4444 -D "cn=directory manager" -j pwd-file --useSSL --trustAll \
  -b "cn=monitor" "(current-send-window=*)" "current-send-window"
dn: cn=Connected Replication Server noordhoek:9989 7164,cn=Replication Server 8989
  1740,cn=cn_admin data,cn=replication,cn=monitor
current-send-window: 93

dn: cn=Replication Domain 30839,cn=dc_example_dc_com,cn=replication,cn=monitor
current-send-window: 100

dn: cn=Replication Domain 14142,cn=cn_schema,cn=replication,cn=monitor
current-send-window: 100

dn: cn=Connected Replica llandudno 27742,cn=Replication Server 8989  1740,cn=cn_
admin data,cn=replication,cn=monitor
current-send-window: 100

dn: cn=Connected Replica llandudno 30839,cn=Replication Server 8989  1740,cn=dc_
example_dc_com,cn=replication,cn=monitor
current-send-window: 72

dn: cn=Connected Replication Server noordhoek:9989 7164,cn=Replication Server 8989
  1740,cn=cn_schema,cn=replication,cn=monitor
current-send-window: 100

dn: cn=Connected Replica llandudno 14142,cn=Replication Server 8989  1740,cn=cn_
schema,cn=replication,cn=monitor
current-send-window: 100

dn: cn=Connected Replication Server noordhoek:9989 7164,cn=Replication Server 8989
  1740,cn=dc_example_dc_com,cn=replication,cn=monitor
current-send-window: 100

dn: cn=Replication Domain 27742,cn=cn_admin data,cn=replication,cn=monitor
current-send-window: 100
```

#### 29.7.2.6 To Monitor Replication Conflicts

When multiple operations are performed on the same entry at the same time, replication conflicts can occur. In some cases, the replication mechanism is able to resolve these conflicts. In other cases, manual conflict resolution is required.

Three types of conflict attributes can be monitored:

- `unresolved-naming-conflicts`. Indicates the number of naming conflicts that could not be resolved by the replication mechanism.
- `resolved-naming-conflicts`. Indicates the number of naming conflicts that have been resolved.
- `resolved-modify-conflicts`. Indicates the number of modify conflicts that have been resolved.

To monitor resolved and unresolved replication conflicts, run the following command:

```
$ ldapsearch -p 4444 -D "cn=directory manager" -j pwd-file --useSSL --trustAll \
  -b "cn=monitor" "(&(unresolved-naming-conflicts=*) \
  (resolved-naming-conflicts=*) (resolved-modify-conflicts=*))" \
  "unresolved-naming-conflicts" "resolved-naming-conflicts" \
  "resolved-modify-conflicts"
```

```
dn: cn=Replication Domain 30839,cn=dc_example_dc_com,cn=replication,cn=monitor
resolved-naming-conflicts: 0
unresolved-naming-conflicts: 0
resolved-modify-conflicts: 0

dn: cn=Replication Domain 14142,cn=cn_schema,cn=replication,cn=monitor
resolved-naming-conflicts: 0
unresolved-naming-conflicts: 0
resolved-modify-conflicts: 0

dn: cn=Replication Domain 27742,cn=cn_admin data,cn=replication,cn=monitor
resolved-naming-conflicts: 0
unresolved-naming-conflicts: 0
resolved-modify-conflicts: 0
```

## 29.8 General Purpose Enterprise Monitoring Solutions

You can use a variety of general UNIX tools to monitor your server environment. For information about these tools, see the man pages on your UNIX system.

### 29.8.1 General UNIX Monitoring Tools

The following general purpose UNIX monitoring tools can be used with Oracle Unified Directory.

| Tool | Description |
|---|---|
| `iostat` | Provides information about disk I/O and CPU usage. |
| `lsof` | Provides information about open file descriptors. |
| `lslk` | Provides information about file system locks. |
| `netstat` | Provides statistics about network functions. |
| `nslookup` | Allows you to query DNS servers for information about hosts and domains. |
| `ping` | Allows you to query the status of a remote host or network gateway. |
| `sar` | UNIX System V performance monitoring tool. |
| `tcpdump` | Allows you to debug and monitor network traffic. |
| `top` | Provides quick, easy monitoring of processes and CPU activities. |
| `trace` | Provides information about which system calls a process makes. |
| `traceroute` | Provides the path a packet takes throughout the Internet to reach its final destination. |
| `vmstat` | Provides statistics about process, virtual memory, disk, trap, and CPU activity. |

### 29.8.2 Solaris Monitoring Tools

The following Solaris monitoring tools can be used with Oracle Unified Directory.

| Tool | Description |
|---|---|
| `lockstat` | Provides information about OS and application locking. Requires DTrace privileges. |
| `mpstat` | Provides statistics about each processor on the system. |
| `pmap` | Provides a breakdown of how much memory a process is using. |
| `proctool` | Monitors processes and threads. |
| `snoop` | Monitors network traffic. Indispensable when debugging low-level packets. |
| `SymbEL/Virtual\\Adrian` | Provides functionality of the above listed tools and more. |
| `truss` | Provides information about which system calls a process makes. |

### 29.8.3 HP-UX Monitoring Tools

The following HP-UX monitoring tools can be used with Oracle Unified Directory.

| Tool | Description |
|---|---|
| `glance` | Provides detailed system information about open file descriptors, locks, and threads. |
| `gpm` | GlancePlus is a graphical real-time performance diagnostic tool. Glance is the character-based component. |
| `tusc` | Provides a system call trapper. |
| `sysdef` | Provides information about kernel parameters. |
| `landiag` | Monitors network statistics. |
| `sam` | Provides a general system administration tool. |

# 30 Tuning Performance

Oracle Unified Directory aims to be high-performing and highly-scalable. Although the server can achieve impressive results with the "out-of-the-box" server configuration and default JVM settings, performance can often be improved significantly through some basic tuning.

The default settings of Oracle Unified Directory are targeted at evaluators and developers who are running equipment with limited resources. When you deploy Oracle Unified Directory in a production environment, it useful to do some initial tuning of the Java Virtual Machine (JVM) and of the server configuration to improve scalability and performance (particularly for write operations).

This chapter covers the following topics:



## 30.1 Assessing Performance Problems

You can obtain a quick idea of whether performance issues are related to problems with the server or with the client by examining the access log at `INSTANCE_DIR/OUD/logs/access`. This log contains entries of the form:

```
[09/Sep/2009:15:36:18 +0200] SEARCH RES conn=1 op=16 msgID=17
  result=0 nentries=1 etime=1
```

The value of the `etime` field is the time (in milliseconds) that the server spent processing the request. Large `etimes` generally indicate an issue on the server side (which can usually be resolved by appropriate performance tuning or indexing. If you are experiencing performance problems but the `etimes` are small, the issue is more likely to be with your client application.

Comprehensive monitoring information is available under the `cn=monitor` entry. For more information, see Chapter 29, "Monitoring Oracle Unified Directory." Oracle Unified Directory performance can also be monitored by using the Enterprise Manager Grid Control plugin. For more information, see the *System Monitoring Plug-in for Oracle Unified Directory User's Guide*.

## 30.2 General Performance Tuning

Note that performance tuning strategies differ depending on whether you are running a directory server or a proxy server.

The following items can improve performance in specific deployment scenarios.

- **Java Version**. Use the most recent Java Runtime Environment (JRE) release available. Oracle Unified Directory is designed to work with Java SE 6 and 7.
- **Environment Variables**. The server uses the *OPENDS_JAVA_HOME* environment variable to point to your installed JRE. If you have multiple versions of Java installed on a system, set the *JAVA_HOME* environment variable to point to the root of the desired installation. In this way, the version of the JRE specified by the *JAVA_HOME* variable can be used by other applications but not by Oracle Unified Directory.

  To specify a JRE installation for the server, do one of the following:

  - Use the `dsjavaproperties` command to set the appropriate environment variables.

    For more information, see dsjavaproperties.
  - Set the *OPENDS_JAVA_BIN* environment variable (with the JAVA binary path).
  - Set the *OPENDS_JAVA_HOME* environment variable (with the JAVA installation path).

## 30.3 Tuning Java Virtual Machine Settings

You can use the `JAVA_ARGS` environment variable to provide global configuration arguments that can be passed to the JVM, or you can use the `java.properties` file. Any argument that can be used with the `java` command can be used with both methods.

It is recommended to tune the JVM for optimal performance and ensures that Oracle Unified Directory applications are robust and responsive. You can tune the JVM by tuning the heap size. The heap size is divided into the following:

- Young generation: Includes operations like PDUs and local variables.
- Old generation: Includes Oracle Unified Directory caches like the JE database cache and the entry cache.
- Permanent generation: Includes constants and classes.

When Oracle Unified Directory is in Directory Server mode, you can perform one of the following database caching option:

- Cache the entire database in database cache. This will give optimal performance but will lead to long cache warmup and larger heap size.
- Cache only the internal nodes of the database Btree (Upper and inner nodes) in database cache and keep remaining RAM for file system cache. This will give good performance, short cache warmup, smaller heap size and is recommended for very large deployments (Above 50MBytes entries). It is recommended for small and medium deployments.

For more information, see Section 30.4, "Determining the Database Cache Size".

**Note:** For proxy mode, use large old generation for distribution with global index.

For more information, see dsjavaproperties.

For additional information about tuning the JVM, see the Java Performance Documentation (`http://java.sun.com/docs/performance/`). The Java Tuning White Paper (`http://java.sun.com/performance/reference/whitepapers/tuning.html`) and the Garbage Collection Tuning (`http://www.oracle.com/technetwork/java/javase/tech/index-jsp-136373.html`) documents are particularly useful.

The following table describes the main JVM tunable options.

| Parameter | Description |
|---|---|
| `-server` | Always use the server JVM instead of the client JVM. The client VM is better optimized for processes that run for a short period of time and need to start as quickly as possible. The server VM can take longer to warm up but is faster in the long run. |
| `-d32` or `-d64` | Select the 32-bit or 64-bit version of the JVM as follows:<br>■ `-d32` provides better performance for JVM heaps smaller than 3.5Gbytes.<br>■ `-XX:+UseCompressedOops` should be used for JVM heaps between 3.5Gbytes and 31Gbytes.<br>■ `-d64`: should be used for JVM heaps over 32Gbytes. |
| `-XX:+UseCompressedOops` | Use this option if you use the 64-bit JVM and if the heap size is less than 32 Gbytes. |

| Parameter | Description |
|---|---|
| `-Xms2g` and `-Xmx2g` | This parameter sets the initial and maximum heap size available to the JVM. Increasing the heap size can improve performance, but setting it too high can have a detrimental effect in the form of longer pauses for full garbage collection runs. The initial and maximum sizes should generally be set to the same values.<br><br>For maximum performance, size the heap so that the entire DB can be cached in memory. In general, you should allocate enough heap for the server runtime and the rest to the DB cache.<br><br>For example, if you want to modify the heap size of an Oracle Unified Directory instance with only one JE backend named `userRoot`. Then you must decide the space needed for the new generation, the old generation and the perm generation. To size the different generations, you must consider the following:<br><br>■ The size of the database impacting the old generation<br>■ Determine the need to use an entry cache impacting the old generation.<br>■ The type of GC used impacting the old generation.<br>■ The type of usage impacting the new generation.<br><br>If you use CMS as the garbage collector of the oldgen, you must take into account the `-XX:CMSInitiatingOccupancyFraction` property when you calculate the heap size so that it is coherent with the size (or percent of the heap) occupied by the dbcache.<br><br>If you set the `CMSInitiatingOccupancyFraction` to 55, the dbcache percent should be set to 50. Then if you have a database on disk that is 10GB, you need at least a heap of 22GB if you want the entire database to fit into the dbcache. |
| `-XX:NewSize=512M` | The total heap space is divided into the old generation and the young generation. This parameter sets the size of the young generation. The remaining memory (old generation) must be sufficient to hold the DB cache plus some overhead. |
| `-XX:+UseConcMarkSweepGC` | Use the Concurrent Mark Sweep (CMS) garbage collector. This option allows the JVM to minimize the response time of LDAP operations, but it can have a small impact on the overall performance (throughput) of the server. Use this option of long pause times are not tolerated. |
| `-XX:CMSInitiatingOccupancyFraction=<percentage>` | Specify the level at which the CMS garbage collection is started. The default value is approximately 68%. Use this value if you want to set the percentage to something other than the default value. |
| `-XX:+UseBiasedLocking` | Improve locking performance in the server in cases where there is not expected to be a high degree of contention. |
| `-XX:LargePageSizeInBytes=256m` | Use large pages for the information it stores in memory. This argument applies primarily to systems using the UltraSPARC T1 processor. |
| `-XX:+UseParallelGC` | Specify that the system should use parallel garbage collection, which is particularly useful on systems with a large number of CPUs. |

| Parameter | Description |
|---|---|
| -XX:+UseParallelOldGC | Specify that the JVM should use parallel garbage collection for the old (tenured) generation. |
| `-XX:ParallelGCThreads=8` | Specify that the JVM should use 8 threads when performing parallel garbage collection. The default is to use a number of threads equal to the number of CPUs, but this can be inappropriate on systems with a very large number of CPUs or on CMT-based systems like those using the UltraSPARC T1 processor. |

## 30.4 Determining the Database Cache Size

If you have installed or configured and initialized an Oracle Unified Directory instance then you can determine the database cache size requirements by measuring the size of `<OUD_INSTANCE_DIR>/OUD/db/userRoot` directory (Assuming there is only one database for the Oracle Unified Directory instance named `userRoot`).

If an Oracle Unified Directory instance is not configured or initialized, then you can determine the memory required to store internal nodes for one index file or the file containing user data, by running the DbCacheSize utility (`com.sleepycat.je.util`).

For more information on using the DbCacheSize utility, see this Javadoc page: http://docs.oracle.com/cd/E17277_02/html/java/com/sleepycat/je/util/DbCacheSize.html.

For example, 10 million entries of 4Kbytes with an index and average key size of 10 bytes are as follows:

```
[oud@oel5 bin]$ java -jar -XX:+UseCompressedOops
/space/Middleware/Oracle_OUD1/lib/je.jar DbCacheSize -records 10000000 -key 10
-data 4000

=== Database Cache Size ===
 Minimum Bytes    Maximum Bytes   Description
---------------  ---------------  -----------
    259,725,752      317,907,896  Internal nodes only
 40,721,011,192   40,779,193,336  Internal nodes and leaf nodes
=== Internal Node Usage by Btree Level ===
 Minimum Bytes    Maximum Bytes      Nodes    Level
---------------  ---------------  ----------  -----
    256,180,800      313,709,120     112,360    1
      3,503,312        4,149,456       1,262    2
         38,864           46,032          14    3
          2,776            3,288           1    4
```

A 10 million entries deployment with 4 Kbytes will require 37 Gbytes to store the full user data in the database cache (4Kbytes entries and the internal nodes of the Database Btree). If you want to store only the internal nodes in the database cache, then 303 Mbytes are required per indexes (3 Gbytes for 10 indexes).

## 30.5 Tuning the Server Configuration

Various components of the server can be tuned to provide performance improvements in specific scenarios. Most performance tuning recommendations depend on several variables, including the anticipated workload, the types of data that are stored, and the hardware and resources available.

The following general tuning recommendations can improve performance in specific deployments.

### 30.5.1 Back End Tuning Parameters

The following Berkeley DB JE tuning parameters can be used to tune performance:

| Parameter | Description |
|---|---|
| `je.checkpointer.highPriority` | If true, the checkpointer uses more resources in order to complete the checkpoint in a shorter time interval. Btree latches are held and other threads are blocked for a longer period. Log cleaner record migration is performed by cleaner threads instead of lazily during eviction and checkpoints (see CLEANER_LAZY_MIGRATION). When set to true, application response time may be longer during a checkpoint, and more cleaner threads may be required to maintain the configured log utilization.<br>Setting that property to false is a way to achieve better throughput and lower response times. |
| `preload-time-limit` | You can configure the server to preload some of the database contents into memory on startup. For large databases, preloading the database cache avoids a long *warmup* period after server startup. For more information, see "Local DB Backend Configuration" in the *Oracle Unified Directory Configuration Reference*. |
| `db-cache-percent` and `db-cache-size` | Use these properties to configure the amount of memory that the database cache uses. For best performance, consider configuring the server so that the whole database fits into the database cache.<br>Determine the approximate size of the database after an import. For example, after doing an import into the `userRoot` back end, run the following command (on UNIX systems) to determine the size of the database:<br>`$ cd INSTANCE_DIR/OUD/db`<br>`$ du -sk userRoot/`<br>`910616 userRoot/`<br>On Windows systems, use an equivalent procedure to determine the database size. Remember that the database size is not static and can increase after an initial import when modifications are made.<br>Setting the JVM heap to 2 Gbytes (`-Xms2g -Xmx2g`), and the `db-cache-percent` to 50, will cause the DB cache to use 1 Gbyte of memory. To monitor the DB cache size, observe the following properties under the `"dn:cn=userRoot Database Environment,cn=monitor"` entry through Jtrace and JMX:<br>Check that `EnvironmentCacheDataBytes` has a value that is consistent with the expected size of the DB cache.<br>Check that `EnvironmentNCacheMiss` does not have unexpected growth when loading the server.<br>As the database grows very large over time due to replication metadata, users, and applications. This may effect the performance after the import. It is recommended that you tune the Oracle Unified Directory JVM heap size (Primarily the old generation). |
| `db-directory` | Ensure that the database is held on a fast file system with adequate storage. The file system should be different to the location of the access logs. By default, the database will grow to twice its original size. For example, if the database is 1 Gbyte after an import, the file system should have at least 2 Gbytes available. |

| Parameter | Description |
|---|---|
| db-evictor-lru-only | Use this property can be used to control how the database cache retains information. Setting this value to false ensures that the internal nodes are maintained in cache, which provides better performance when the JE cache holds only a small percentage of the database contents. |
| db-txn-durability | Use this property to configure durability for write operations. Reducing durability can increase write performance, but it can also increase the chance of data loss in the event of a JVM crash or a system crash. This property takes the following values:<br>**write-to-disk.** All data are written synchronously to disk.<br>**write-to-fs.** Data are written to the file system immediately but might stay in the file system before being flushed to disk.<br>**write-to-cache.** Data are written to an internal buffer and flushed to the file system, then to disk when necessary. |
| db-log-file-max | Use this property to control the size of JE log files. Increasing the file size can improve write performance, but it can also make it harder to maintain the desired utilization percentage. |
| db-num-cleaner-threads<br>and<br>db-cleaner-min-utilization | These properties control how the cleaner works, which keeps the database size down and keeps up with high write throughput. |
| db-num-lock-tables | On systems with a large number of CPUs, this property can improve concurrency within the database lock manager. |

## 30.5.2 Core Server Tuning Parameters

The following core server tuning parameters can be used to tune performance:

- num-request-handlers

  This property can be configured so that the LDAP connection handler (and the LDAPS connection handler, if it is enabled) use multiple threads for decoding client requests. Increasing the number of threads on systems with a larger number of CPUs can improve performance. As a rule of thumb, you should set this property to a quarter the number of CPUs, with a maximum of twelve.

  In some cases disabling the keep-stats property can help reduce lock contention in the connection handlers. For more information, see "LDAP Connection Handler Configuration" in the *Oracle Unified Directory Configuration Reference*.

- num-worker-threads

  The default value of this property is two times the number of CPUs. This value is sufficient in most deployments.

- log-file

  Ensure that the access log publisher is on a fast file system, or turn it off altogether by setting the enabled property to false. For more information see "File Based Access Log Publisher Configuration" in the *Oracle Unified Directory Configuration Reference*.

### 30.5.3 Additional Tuning Recommendations

The following additional recommendations can improve performance in specific scenarios.

- **Enable an Entry Cache**. In some cases, particularly those involving relatively small directories (for example, up to a few hundred thousand entries), it can be useful to enable an entry cache. In general the FIFO entry cache provides better results than the soft reference entry cache. For more information, see "Entry Cache Configuration" in the *Oracle Unified Directory Configuration Reference*.

  For large database, it is recommended that you store only a specific set of the data in the cache, by using the `include-filter` property. Storing static groups in the entry cache can greatly improve the overall performance of the server. This reduces the time required to perform group membership lookup, which is necessary in evaluating ACIs, for example.

- **Disable Unused Virtual Attributes**. If the functionality needed by one or more of the virtual attributes is not required, they can be disabled for a slight performance improvement when decoding entries. For more information, see "Virtual Attribute Configuration" in the *Oracle Unified Directory Configuration Reference*.

- **Disable Unused Access Logging**. If access logging is not necessary, disabling the server access logger can help improve performance. For more information, see "Log Publisher Configuration" in the *Oracle Unified Directory Configuration Reference*.

- **Disable Unused Access Control Handlers**. If you do not need access control processing in the server, then you can disable it by setting the `enabled` configuration property to `false` for the Access Control Handler. You can set the property by using `dsconfig`.

- **Reduce Lock Contention**. On systems with large numbers of CPUs (for example, chip multi-threading (CMT) systems with several hardware threads per core), you can reduce lock contention by setting the `org.opends.server.LockManagerConcurrencyLevel` system property to be equal to the number of worker threads you intend to use.

  **Note:** This property must be set as a JVM system property, because it can be required very early in the server startup process, even before accessing the server configuration.

# A

# Oracle Unified Directory Command Line Interface

Oracle Unified Directory includes a number of command-line utilities that are used to interact with the directory server and the proxy server. Utilities are also provided to prepare a server to be part of a multi-version topology using the replication gateway.

This appendix describes all of the commands that are provided with Oracle Unified Directory 11*g* Release 2 (11.1.2). Some of these commands are specific to a directory server instance and cannot be used to configure a proxy server. Similarly, a number of the commands are specific to the proxy and cannot be used to configure a directory server.

This appendix covers the following topics:



## A.1 General Command-Line Usage Information

The following sections provide general information about command usage:



### A.1.1 Summary of Server Commands and Their Use

The tables in this section provide a summary of the server commands and how they can be used. The tables use the following legend:

**Remote**
The command can be launched on a remote server

**Offline**
The command can be launched when the server is stopped

**Online**
The command connects to a running server instance

**Administration Port Only**
The command *must* use the administration connector to access the server (on port 4444 by default)

> **Note:** Not all the commands listed in the following tables are supported for a proxy server instance.

***Table A–1 Server Administration Commands***

| Command | Remote | Offline | Online | Administration Connector |
|---|---|---|---|---|
| create-rc-script | | | | |
| dsconfig | X | | X | X |
| dsjavaproperties | | X | | |
| dsreplication | X | | X | X |
| gicadm | X | | X | X |
| oudExtractMovePlan | | X | X | |
| oudCopyConfig | | X | X | |
| oudPasteConfig | | X | | |
| start-ds | | X | | |
| status | X | X | X | X |
| stop-ds | X | | X | X |
| uninstall | | X | X | X |
| upgrade | | X | | |
| windows-service | | X | | |

***Table A–2 Data Administration Commands***

| Command | Remote | Offline | Online | Administration Connector |
|---|---|---|---|---|
| backup | X * | X | X | X |
| base64 | | X | | |
| dbtest | | X | | |
| encode-password | | X | | |
| export-ldif | X * | X | X | X |
| import-ldif | X * | X | X | X |
| ldapcompare | X | | X | |
| ldapdelete | X | | X | |
| ldapmodify | X | | X | |
| ldappasswordmodify | X | | X | |
| ldapsearch | X | | X | |
| ldif-diff | | X | | |

*Table A–2 (Cont.) Data Administration Commands*

| Command | Remote | Offline | Online | Administration Connector |
|---|---|---|---|---|
| ldifmodify | | X | | |
| ldifsearch | | X | | |
| list-backends | | X | | |
| make-ldif | | X | | |
| manage-account | X | | X | X |
| manage-tasks | X | | X | X |
| rebuild-index | | X | | |
| restore | X * | X | X | X |
| split-ldif | | X | X | |
| verify-index | | X | | |

* The command can be launched remotely but the data files must be on the host on which the server is running.

## A.1.2 Using a Properties File With Server Commands

Certain command-line utilities can use a common properties file to provide default values for options such as the following:

- The host name and port number of the server
- Whether to use SSL or StartTLS to communicate with the server
- The bind DN to use when connecting to the server

The following utilities can use a properties file:

- `backup`
- `dsconfig`
- `dsreplication`
- `export-ldif`
- `gicadm`
- `import-ldif`
- `split-ldif`
- `ldapcompare`
- `ldapdelete`
- `ldapmodify`
- `ldappasswordmodify`
- `ldapsearch`
- `manage-tasks`
- `oud-setup`
- `oud-proxy-setup`

- `oud-replication-gateway-setup`
- `restore`
- `status`
- `stop-ds`
- `uninstall`

The following mutually exclusive options are used with the command-line utilities to indicate whether a properties files is used:

`--propertiesFilePath` ***path***
Specify the path to the file that contains default values for command-line options.

`--noPropertiesFile`
Indicates that the properties file is not used to obtain default values for command-line options.

### A.1.2.1 Locating the Properties File

Utilities that use the common properties file have the following default behavior:

- If the `--noPropertiesFile` option is specified, the command-line interface does not try to locate a properties file. Only options specified on the command line are evaluated.
- If the `--propertiesFilePath` option is specified, property values are read from this file.
- If neither `--propertiesFilePath` nor `--noPropertiesFile` is specified, the command-line interface attempts to find a properties file in the following locations:
  - *USERDIRECTORY*`/.opends/tools.properties`
  - *INSTANCE_DIR*`/OUD/config/tools.properties`
- If no properties file is found in either of these locations, the default behavior is applied (only arguments specified on the command line are evaluated).

### A.1.2.2 Order of Precedence of Options and Properties

If an option is provided on the command line, this option and its corresponding value are used by the command-line interface. In other words, options specified on the command line take precedence over the properties defined in the properties file.

The properties file has the standard JAVA properties file format (*property-name=value*). As such, the file supports variations on property names to enable them to be overridden according to the command that uses them. For example, the properties file might contain the following:

```
hostname=localhost
port=4444
bindDN=cn=Directory Manager
bindPasswordFile=/path/pwd-file
baseDN=dc=example,dc=com
searchScope=sub
sortOrder=givenName
virtualListView=0:2:1:0
```

If a command-line interface uses the `port` property, the command first tries to locate a *toolname*`.port` definition. If this is not defined, the command tries to locate a `port`

definition. For example, the properties file might have several port options defined for different utilities:

```
port=4444
ldapsearch.port=1389
ldapcompare.port=1389
ldapmodify.port=1389
ldapdelete.port=1389
```

**Note:** Do **not** use quotation marks around the values in the properties file (for example, `port="4444"`).

## A.2 Server Administration Commands

The following sections describe the server administration commands:



### A.2.1 create-rc-script

The `create-rc-script` command generates a shell script to start, stop, and restart the directory server.

#### A.2.1.1 Synopsis

`create-rc-script` [*options*]

### A.2.1.2 Description

The `create-rc-script` command can be used to generate a shell script to start, stop, and restart the directory server. You can update the resulting script to suit the needs of your directory service. This command is available for UNIX or Linux systems only.

The `create-rc-script` command uses the `OPENDS_JAVA_*` and `JAVA_*` variables.

### A.2.1.3 Options

The `create-rc-script` command accepts an option in either its short form (for example, `-f` *filename*) or its long form equivalent (for example, `--outputFile` *filename*).

`-f, --outputFile` ***filename***
Specify the path to the output file.

`-j, --javaHome` ***javaHomePath***
Specify the path to the Java installation that should be used to run the server.

`-J, --javaArgs` ***javaArgs***
Specify the set of arguments that should be passed to the JVM when running the server.

`-u, --userName` ***userName***
Specify the name of the user account under which the server should run. The user account must have the appropriate permissions to run the script.

### A.2.1.4 General Options

`--version`
Display the version information for the directory server.

`-?, -H, --help`
Display command-line usage information for the `create-rc-script` command.

### A.2.1.5 Examples

The following examples show how to use the `create-rc-script` command.

#### *Example A–1 Creating the Script*

The following command generates the script to start, stop, and restart the directory server. It creates the file called `myscript`, specified by the `-f` option:

```
$ create-rc-script -f myscript
```

#### *Example A–2 Starting the Directory Server by Using the New Script*

The following command uses the newly created script (see previous example) to start the directory server.

```
$ myscript start
```

#### *Example A–3 Stopping the Directory Server by Using the New Script*

The following command uses the newly created script (see first example) to stop the directory server.

```
$ myscript stop
```

***Example A–4 Restarting the Directory Server by Using the New Script***

The following command uses the newly created script (see first example) to restart the directory server.

```
$ myscript restart
```

***Example A–5 Specifying `JAVA_HOME` and `JAVA_ARGS` in the Script***

The following command uses the `-u` (`--userName`), `-j` (`--javaHome`) and `-J` (`--javaArgs`) options.

```
$ create-rc-script -f myscript -u sysAdmin -j /usr/java -J "-Xms128m -Xmx128m"
```

### A.2.1.6 Code Generated by the `create-rc-script` Command

The `create-rc-script` command from the example above generates the following code:

```
# /bin/sh
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License, Version 1.0 only
# (the "License").  You may not use this file except in compliance
# with the License.
#
# You can obtain a copy of the license at
# https://OpenDS.dev.java.net/OpenDS.LICENSE.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at
# trunk/opends/resource/legal-notices/OpenDS.LICENSE.  If applicable,
# add the following below this CDDL HEADER, with the fields enclosed
# by brackets "[]" replaced with your own identifying information:
#      Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END

# Set the path to the OpenDS instance to manage
INSTANCE_ROOT="/usr/local/opends/standalone/ds-server-1"
export INSTANCE_ROOT

# Specify the path to the Java installation to use
OPENDS_JAVA_HOME="/usr/java"
export OPENDS_JAVA_HOME

# Specify arguments that should be provided to the JVM
JAVA_ARGS="-Xms128m -Xmx128m"
export JAVA_ARGS

# Determine what action should be performed on the server
case "${1}" in
start)
/bin/su sysAdmin "${INSTANCE_ROOT}/bin/start-ds" --quiet
exit ${?}
;;
stop)
/bin/su sysAdmin "${INSTANCE_ROOT}/bin/stop-ds" --quiet
```

```
exit ${?}
;;
restart)
/bin/su sysAdmin "${INSTANCE_ROOT}/bin/stop-ds" --restart --quiet
exit ${?}
;;
*)

echo "Usage:  $0 { start | stop | restart }"
exit 1
;;
esac
```

#### A.2.1.7 Exit Codes

An exit code of 0 indicates success. A non-zero exit code indicates that an error occurred.

#### A.2.1.8 Location

The `create-rc-script` command is located at this path:

UNIX and Linux: *INSTANCE_DIR*`/OUD/bin`

#### A.2.1.9 Related Commands

Section A.2.15, "start-ds"

Section A.2.17, "stop-ds"

### A.2.2 dps2oud

The `dps2oud` command allows you to migrate a Directory Proxy Server (DPS) configuration to an Oracle Unified Directory configuration.

#### A.2.2.1 Synopsis

`dps2oud` [*options*]

#### A.2.2.2 Description

The `dps2oud` command allows you to migrate a DPS configuration to an Oracle Unified Directory configuration. The `dps2oud` command takes a DPS configuration as the input and generates a batch file that comprises `dsconfig` commands, which are used to create an equivalent Oracle Unified Directory configuration. The `dps2oud` command reads the DPS configuration either through a file or through the LDAP protocol on a running DPS instance.

#### A.2.2.3 Options

The `dps2oud` command accepts the following options.

`-o, --outputFile` ***file***
The output file for `dsconfig` commands.

`-f, --dpsConfigFile` ***file***
Specifies the name of the DPS config file to use.

`-c, --createDisabledObjects`
Creates DPS-disabled objects.

`-P, --printDsConfigCmds`
Prints `dsconfig` commands.

### A.2.2.4 LDAP Connection Options

`-h, --hostname` ***host***
DPS server hostname or IP address.

`-j, --bindPasswordFile` ***filename***
The full path to the file containing the bind password.

`-p, --port` ***port***
DPS server port number.

`-D, --BindDN` ***bindDN***
DN to use to bind to the DPS server.

### A.2.2.5 General Options

`-?, -H, --help`
Displays command-line usage information for the command and exit without making any attempt to stop or restart the directory server.

`-V, --version`
Displays the version information for the directory server.

### A.2.2.6 Examples

The following examples show how to use the `dps2oud` command.

***Example A–6 Viewing the Global Help Subcommands***

The following command displays the available global Help subcommands:

```
$ dps2oud --help
```

***Example A–7 Migrating a Directory Proxy Server Configuration to an Oracle Unified Directory Configuration***

You can migrate a DPS configuration to an Oracle Unified Directory configuration using one of the following methods:

Method 1: Reading a DPS configuration from an LDIF file

The following command displays how to read a DPS configuration from an LDIF file:

```
$ dps2oud -f dse.ldif -o oud_conf_cmds
```

The following command provides the path to a batch file containing a set of `dsconfig` commands to be executed:

```
$ dsconfig -F oud_conf_cmds
```

Method 2: Reading a DPS configuration from a running DPS instance

The following command displays how to read a DPS configuration from a DPS instance:

```
$ dps2oud -h dpsHost -p 389 -D "cn=Proxy Manager"  -j /path/pwd-file -o
oud_conf_cmds
```

The following command provides the path to a batch file containing a set of `dsconfig` commands to be executed:

```
$ dsconfig -F oud_conf_cmds
```

#### A.2.2.7 Exit Codes

An exit code of 0 indicates that the operation completed successfully. A nonzero exit code indicates that an error occurred during processing.

#### A.2.2.8 Location

- UNIX and Linux: *INSTANCE_DIR*`/OUD/bin/dps2oud`
- Windows: *INSTANCE_DIR*`\OUD\bat\dps2oud.bat`

#### A.2.2.9 Related Commands

Section A.2.4, "dsconfig"

### A.2.3 ds2oud

The `ds2oud` command manages the migration from an Oracle Directory Server Enterprise Edition directory server instance to Oracle Unified Directory.

#### A.2.3.1 Synopsis

`ds2oud` [*options*]

#### A.2.3.2 Description

The `ds2oud` command enables you to manage the migration from an Oracle Directory Server Enterprise Edition directory server instance to Oracle Unified Directory. The `ds2oud` command first allows you to diagnose the targeted Oracle Directory Server Enterprise Edition directory server, and then performs the migration task. It is based on the premise that the existing Oracle Unified Directory instance is modified to be compatible with the Oracle Directory Server Enterprise Edition directory server to be migrated. The `ds2oud` command runs in interactive mode, if you do not specify options. Interactive mode works much like a wizard, walking you through every aspect of the migration.

You can also run the `ds2oud` command in batch mode. In batch mode, a batch file that comprises `dsconfig` commands is generated. These commands are used to create an equivalent Oracle Unified Directory configuration. So, you can run `ds2oud` once, and create a single batch file that can be used to configure any number of Oracle Unified Directory instances.

You must ensure while running the `ds2oud` command that the Oracle Unified Directory instance (to which the Oracle Directory Server Enterprise Edition instance is being migrated) is configured without any suffixes.

#### A.2.3.3 Options

The `ds2oud` command accepts the following options.

`-d, --diagnose`
Diagnoses the targeted Oracle Directory Server Enterprise Edition directory server.

`-f, --ldifDBFile` ***file***
Diagnoses the Oracle Directory Server Enterprise Edition directory server LDIF database file.

`-u, --userSchemaFile` ***file***
Specifies the user schema to be taken into consideration. It applies to `-f` subcommand.

`-a, --migrateAll`
Propagates schema and configuration elements from Oracle Directory Server Enterprise Edition directory server to Oracle Unified Directory server.

`-s, --migrateUserSchema`
Propagates the User schema from Oracle Directory Server Enterprise Edition directory server to Oracle Unified Directory server.

You must migrate the schema *before* you migrate the configuration, otherwise the migration can produce unpredictable results.

`-c, --migrateConfiguration`
Propagates configuration elements from Oracle Directory Server Enterprise Edition directory server to Oracle Unified Directory server.

You must migrate the schema *before* you migrate the configuration, otherwise the migration can produce unpredictable results.

`-w, --uniqueWorkflowElement`
Use a unique workflow element for all the naming contexts to migrate. This applies to `-c` subcommand.

### A.2.3.4 Oracle Directory Server Enterprise Edition LDAP Connection Options

`-D, --odseeBindDN` ***bindDN***
DN to use to bind to the Oracle Directory Server Enterprise Edition server.

`-j, --odseeBindPasswordFile` ***filename***
Oracle Directory Server Enterprise Edition bind password file.

`-h, --odseeHostname` ***host***
Oracle Directory Server Enterprise Edition server hostname. The default value is localhost.

`-p, --odseePort` ***port***
Oracle Directory Server Enterprise Edition server port number. The default value is 389.

`-Z, --odseeUseSSL`
Establishes an Oracle Directory Server Enterprise Edition SSL-encrypted connection.

`-P, --odseeTrustStorePath` ***trustStorePath***
Use the Oracle Directory Server Enterprise Edition trust store certificate in the specified path. This option is not needed if `-X` is used, although a trust store should be used when working in a production environment.

`-U, --odseeTrustStorePasswordFile` ***filename***
Use the password in the specified file to access the certificates in the Oracle Directory Server Enterprise Edition trust store. This option is only required if `--odseeTrustStorePath` is used and the specified trust store requires a password to access its contents (most trust stores do not require this).

`-X, --odseeTrustAll`
Trust all certificate that the Oracle Directory Server Enterprise Edition server presents. This option can be used for testing purposes, but for security reasons, a trust store should be used to determine whether the Oracle Directory Server Enterprise Edition should accept the server certificate.

### A.2.3.5 Oracle Unified Directory LDAP Connection Options

`--oudBindDN` ***bindDN***
DN to use to bind to the Oracle Unified Directory server.

`--oudBindPasswordFile` ***filename***
Oracle Unified Directory bind password file.

`--oudHostname` ***host***
Oracle Unified Directory server hostname. The default value is localhost.

`--oudPort` ***port***
Oracle Unified Directory server port number. The default value is 389.

`--oudAdminPort` ***port***
Oracle Unified Directory server administration port. The default value is 444.

`--oudUseSSL`
Establishes an Oracle Unified Directory SSL-encrypted connection.

`--oudTrustStorePath` ***trustStorePath***
Use the Oracle Unified Directory trust store certificate in the specified path.

`--oudTrustStorePasswordFile` ***filename***
Use the password in the specified file to access the certificates in the Oracle Unified Directory trust store. This option is only required if `--oudTrustStorePath` is used and the specified trust store requires a password to access its contents (most trust stores do not require this).

`--oudTrustAll`
Trust all certificate that the Oracle Unified Directory server presents. This option can be used for testing purposes, but for security reasons, a trust store should be used to determine whether the Oracle Unified Directory should accept the server certificate.

### A.2.3.6 Command Input/Output Options

`-n, --no-prompt`
Use the non-interactive mode. If data in the command is missing, the user is not prompted and the tool fails.

`-o, --outputFile` ***filename***
Redirects the output into the specified output file.

`-F, --batchFilePath` ***filename***
This option specifies the name of the output file that contains a set of `dsconfig` commands to execute to migrate the configuration.

When you run `ds2oud` with this option, a batch file is generated that includes all of the `dsconfig` commands required to create the equivalent Oracle Unified Directory configuration. So, you can run `ds2oud` once, and create a single batch file that can be used to configure any number of Oracle Unified Directory instances.

`--displayCommand`
Display the equivalent non-interactive dsconfig commands (for the migration of Oracle Directory Server Enterprise Edition configuration parameters).

### A.2.3.7 General Options

`-?, -H, --help`
Displays command-line usage information for the command and exit without making any attempt to stop or restart the directory server.

`-V, --version`
Displays the version information for the directory server.

### A.2.3.8 Examples

The following examples show how to use the `ds2oud` command.

***Example A–8 Viewing the Global Help Subcommands***

The following command displays the available global Help subcommands:

```
$ ds2oud --help
```

***Example A–9 Running `ds2oud` in Interactive Mode From the Command Line***

The `ds2oud` command can be run in interactive mode, where you are prompted for migration options. To run `ds2oud` in interactive mode, type the following command:

```
$ ds2oud
What do you want to do ?

1)  Diagnose an ODSEE directory server instance
2)  Diagnose an ODSEE LDIF data file
3)  Migrate all ( user schema + configuration )
4)  Migrate the user schema
5)  Migrate global configuration parameters

c)  cancel
```

For each preceding action, you must first provide the connection options for the Oracle Directory Server Enterprise Edition server (for diagnosis) or both the Oracle Directory Server Enterprise Edition and Oracle Unified Directory servers (for migration).

***Example A–10 Running `ds2oud` for Diagnosing Data***

The following command is run to diagnose the data present in the Oracle Directory Server Enterprise Edition directory server:

```
$ ds2oud -f odseeDataFile.ldif -u 99user.ldif

*******************************************************************************
* Diagnose ODSEE LDIF data file :  odseeDataFile.ldif
*******************************************************************************
The data were validated successfully regarding the OUD schema
```

***Example A–11 Migrating an Existing Oracle Directory Server Enterprise Edition Configuration to an Oracle Unified Directory Configuration***

Use the following commands to migrate an existing Oracle Directory Server Enterprise Edition Configuration to a new Oracle Unified Directory Configuration

The following command migrates an existing Oracle Directory Server Enterprise Edition configuration and schema:

```
$ ds2oud --migrateAll -D "cn=directory manager"
-j /tmp/pwd -h hostname -p ldapPort
--oudBindDN "cn=directory manager" --oudBindPasswordFile /tmp/pwd
--oudHostname hostname2 --oudPort ldapPort2 --oudAdminPort adminPort  -n
```

The following command provides the path to a batch file containing a set of `dsconfig` commands to be executed to create a new Oracle Unified Directory configuration:

```
$ ds2oud --migrateConfiguration --batchFilePath batchFile
-D "cn=directory manager" -j /tmp/pwd -h hostname
-p ldapPort --oudBindDN "cn=directory manager"
--oudBindPasswordFile /tmp/pwd  --oudHostname hostname2
--oudPort ldapPort2 --oudAdminPort adminPort -n
```

#### A.2.3.9 Exit Codes

`0`
Successful.

`1`
Unable to initialize arguments.

`2`
Cannot parse arguments because the provided arguments are not valid or there was an error checking the user data.

`3`
At least one step into the migration process has failed.

`4`
The user canceled the operation in interactive mode.

#### A.2.3.10 Location

- UNIX and Linux: *INSTANCE_DIR*`/OUD/bin/ds2oud`
- Windows: *INSTANCE_DIR*`\OUD\bat\ds2oud.bat`

#### A.2.3.11 Related Commands

- Section A.2.4, "dsconfig"

## A.2.4 dsconfig

The `dsconfig` command allows you to define a base configuration for the Directory Server.

#### A.2.4.1 Synopsis

`dsconfig` [*subcommands*] [*Options*]

### A.2.4.2 Description

The `dsconfig` command enables you to create, manage, and remove the base configuration for a server instance. The server configuration is organized as a set of components that `dsconfig` can access by using one or more subcommands. All components have zero or more configurable properties. These properties can be queried and modified to change the behavior of the component.

The `dsconfig` command accesses the server over SSL through the administration connector (described in Section 14.3, "Managing Administration Traffic to the Server").

Unless you specify all configuration parameters and the `-n` (`--no-prompt`) option, `dsconfig` runs in interactive mode. Interactive mode works much like a wizard, walking you through every aspect of the server configuration. For more information, see Section 14.1.2, "Using `dsconfig` in Interactive Mode."

### A.2.4.3 Help Subcommands

The `dsconfig` command provides help functions that list the component subcommands needed to manage your base configuration.

`--help-distribution`
Display subcommands relating to distribution.

`--help-general-configuration`
Display subcommands relating to general configuration.

`--help-load-balancing`
Display subcommands relating to load balancing.

`--help-local-storage`
Display subcommands relating to local storage.

`--help-miscellaneous-workflow-elements`
Display subcommands relating to miscellaneous workflow elements.

`--help-remote-storage`
Display subcommands relating to remote storage.

`--help-replication`
Display subcommands relating to replication.

`--help-schema`
Display subcommands relating to schema.

`--help-security`
Display subcommands relating to authentication and authorization.

`--help-virtualization`
Display subcommands relating to virtualization.

`--help-all`
Display all subcommands.

### A.2.4.4 General Subcommands

The following subcommand lists the objects and properties of the server instance.

**list-properties**
Displays the managed objects and properties. Option types are as follows:

`r` — Property values are readable.

`w` — Property values are writable.

`m` — The property is mandatory.

`s` — The property is single-valued.

`a` — Administrative action is required for changes to take effect.

Suboptions are as follows:

`-t, --type` *type*. Component type.

`-c, --category` *category*. Category of the component. The value for `type` must be one of the component types associated with the *category* that is specified using the `--category` suboption.

`--inherited`. Modifies the display output to show the inherited properties of components.

`--advanced`. Modifies the display output to show the advanced properties of components.

`--property` *property*. The name of a property to be displayed.

### A.2.4.5 Distribution Subcommands

The following subcommands allow you to define the base configuration for the directory server.

`create-distribution-algorithm`
Creates distribution algorithms. Suboptions are as follows:

`--element-name` *name*. The name of the distribution workflow element.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t,--type` *type*. The type of Distribution Algorithm that should be created. The value for *type* can be one of `capacity, dnpattern, generic, lexico,` or `numeric`.

`create-distribution-partition`
Creates distribution partitions. Suboptions are as follows:

`--element-name` *name*. The name of the distribution workflow element.

`--partition-name` *name*. The name of the new distribution partition.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t,--type` *type*. The type of Distribution Partition that should be created. The value for *type* can be one of `capacity`, `dnpattern`, `generic, lexico`, or `numeric`.

`create-workflow-element --type distribution`
Creates Workflow Elements. Suboptions are as follows:

`--element-name` *name.* The name of the new Workflow Element.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t, --type` *type.* The type of Workflow Element that should be created. The value for *type* can be one of `ad-paging, backup-local-backend, db-local-backend,distribution, dn-renaming, eus, eus-context, fa, global-index-local-backend, global-index-replication-changes-local-backend, kerberos-auth-provider, ldif-local-backend, load-balancing, memory-local-backend, monitor-local-backend, null-local-backend, pass-through-authentication, plugin, proxy-ldap, rdn-changing, transformations, trust-store-local-backend.`

`create-global-index`
Creates global indexes. Suboptions are as follows:

`--extension-name` *name.* The name of the Global Index Catalog Extension.

`--index-name` *name.* The name of the new Global Index.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`create-extension --type global-index-catalog`
Creates Extensions. Suboptions are as follows:

`--extension-name` *name*. The name of the Global Index Catalog Extension.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t,--type` *type.* The type of Extension that should be created. The value for type can be one of `global-index-catalog, global-index-catalogs-shared-cache, ldap-server.`

`create-global-index-catalog-replication-domain`
Creates global index catalog replication domains. Suboptions are as follows:

`--extension-name` *name*. The name of the Global Index Catalog Extension.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`create-extension --type global-index-catalogs-shared-cache`
Creates Extensions. Suboptions are as follows:

`--extension-name` *name*. The name of the new Extension.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t,--type` *type.* The type of Extension that should be created. The value for *type* can be one of `global-index-catalog, global-index-catalogs-shared-cache, ldap-server.`

`create-workflow-element --type global-index-local-backend`

Creates Workflow Elements. Suboptions are as follows:

`--element-name` *name.* The name of the new Workflow Element.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t, --type` *type*. The type of Workflow Element that should be created. The value for *type* can be one of `ad-paging, backup-local-backend, db-local-backend, distribution, dn-renaming, eus, eus-context, fa, global-index-local-backend, global-index-replication-changes-local-backend, kerberos-auth-provider, ldif-local-backend, load-balancing, memory-local-backend, monitor-local-backend, null-local-backend, pass-through-authentication, plugin, proxy-ldap, rdn-changing, transformations, trust-store-local-backend`.

`create-workflow-element --type global-index-replication-changes-local-backend`
Creates Workflow Elements. Suboptions are as follows:

`--element-name` *name*. The name of the new Workflow Element.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t, --type` *type*. The type of Workflow Element that should be created. The value for *type* can be one of `ad-paging, backup-local-backend, db-local-backend, distribution, dn-renaming, eus, eus-context, fa, global-index-local-backend, global-index-replication-changes-local-backend, kerberos-auth-provider, ldif-local-backend, load-balancing, memory-local-backend, monitor-local-backend, null-local-backend, pass-through-authentication, plugin, proxy-ldap, rdn-changing, transformations, trust-store-local-backend`.

`delete-distribution-algorithm`
Deletes distribution algorithms. Suboptions are as follows:

`--element-name` *name*. The name of the Distribution Workflow Element.

`-f,--force`. Ignore nonexistent distribution algorithms.

`delete-distribution-partition`
Deletes distribution partitions. Suboptions are as follows:

`--element-name` *name*. The name of the distribution workflow element.

`--partition-name` *name*. The name of the distribution partition.

`-f,--force`. Ignore nonexistent distribution partitions.

`delete-extension`
Deletes Extensions. Suboptions are as follows:

`--extension-name` *name*. The name of the Extension.

`-f,--force`. Ignore nonexistent extensions.

`delete-global-index`
Deletes global indexes. Suboptions are as follows:

`--extension-name` *name*. The name of the Global Index Catalog Extension.

`--index-name` *name*. The name of the Global Index.

`-f,--force`. Ignore nonexistent global indexes.

`delete-global-index-catalog-replication-domain`
This command is supported only for the proxy. To manage the global index see Section A.2.7, "gicadm."

Deletes global index catalog replication domains. Suboptions are as follows:

`--extension-name` *name*. The name of the Global Index Catalog Extension.

`-f,--force`. Ignore nonexistent global index catalog replication domains.

`delete-workflow-element`
Deletes Workflow Elements. Suboptions are as follows:

`--element-name` *name*. The name of the Workflow Element.

`-f,--force`. Ignore nonexistent workflow element.

`get-distribution-algorithm-prop`
Shows distribution algorithm properties. Suboptions are as follows:

`--element-name` *name*. The name of the distribution workflow element.

`--property` *property*. The name of a property to be displayed.

`-E,--record`. Modifies the display output to show one property value per line.

`-z,--unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m,--unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`get-distribution-partition-prop`
Shows distribution partition properties. Suboptions are as follows:

`--element-name` *name*. The name of the distribution workflow element.

`--partition-name` *name*. The name of the distribution partition.

`--property` *property*. The name of a property to be displayed.

`-E,--record`. Modifies the display output to show one property value per line.

`-z,--unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m,--unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`get-global-index-catalog-replication-domain-prop`
This command is supported only for the proxy. To manage the global index see Section A.2.7, "gicadm."

Shows global index catalog replication domain properties. Suboptions are as follows:

`--extension-name` *name*. The name of the Global Index Catalog Extension.

`--property` *property*. The name of a property to be displayed.

`-E,--record`. Modifies the display output to show one property value per line.

`-z,--unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`get-global-index-prop`
This command is supported only for the proxy. To manage the global index see Section A.2.7, "gicadm."

Shows Global index properties. Suboptions are as follows:

`--extension-name` *name*. The name of the Global Index Catalog Extension.

`--index-name` *name*. The name of the Global Index.

`--property` *property*. The name of a property to be displayed.

`-E, --record`. Modifies the display output to show one property value per line.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`list-distribution-algorithm`
This command is supported for only proxy.

Lists existing distribution algorithm. Suboptions are as follows:

`--element-name` *name*. The name of the distribution workflow element.

`--property` *property*. The name of a property to be displayed.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`list-distribution-partitions`
This command is supported only for the proxy.

Lists existing distribution partitions. Suboptions are as follows:

`--element-name` *name*. The name of the distribution workflow element.

`--property` *property*. The name of a property to be displayed.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`list-extensions`
Lists existing Extensions. Suboptions are as follows:

`--property` *property*. The name of a property to be displayed.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`list-global-index-catalog-replication-domain`
This command is supported only for the proxy. To manage the global index see Section A.2.7, "gicadm."

Lists existing global index catalog replication domain. Suboptions are as follows:

`--extension-name` *name*. The name of the Global Index Catalog Extension.

`--property` *property*. The name of a property to be displayed.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`list-global-indexes`
Lists existing global indexes. Suboptions are as follows:

`--extension-name` *name*. The name of the Global Index Catalog Extension.

`--property` *property*. The name of a property to be displayed.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`list-workflow-elements`
Lists existing Workflow Elements. Suboptions are as follows:

`--property` *property*. The name of a property to be displayed.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`set-distribution-algorithm-prop`
This command is supported only for the proxy.

Modifies distribution algorithm properties. Suboptions are as follows:

`--element-name` *name*. The name of the distribution workflow element.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*`:`*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*`:`*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

`set-distribution-partition-prop`
This command is supported only for the proxy.

Modifies distribution partition properties. Suboptions are as follows:

`--element-name` *name*. The name of the distribution workflow element.

`--partition-name` *name*. The name of the distribution partition.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*`:`*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*`:`*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

`set-extension-prop`
Modifies Extension properties. Suboptions are as follows:

--extension-name name. The name of the Extension.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*`:`*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*`:`*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

`set-global-index-catalog-replication-domain-prop`
This command is supported only for the proxy.

Modifies global index catalog replication domain properties. Suboptions are as follows:

`--extension-name` *name*. The name of the Global Index Catalog Extension.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*`:`*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*`:`*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

`set-global-index-prop`
This command is supported only for the proxy.

Modifies global index properties. Suboptions are as follows:

`--extension-name` *name*. The name of the Global Index Catalog Extension.

`--index-name` *name.* The name of the Global Index.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*`:`*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*`:`*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

`set-workflow-element-prop`
Modifies Workflow Element properties. Suboptions are as follows:

`--element-name` *name.* The name of the Workflow Element.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*`:`*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*`:`*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

### A.2.4.6 General Configuration Subcommands

The following subcommands configure the core server.

`create-alert-handler`
Creates alert handlers. Suboptions are as follows:

`--handler-name` *name*. The name of the new alert handler.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t`, `--type` *type*. The type of Alert Handler that should be created. The value for *type* can be one of `custom`, `jmx`, or `smtp`.

`create-certificate-mapper`
Creates certificate mappers. Suboptions are as follows:

`--mapper-name` *name*. The name of the new certificate mapper.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t, --type` *type*. The type of Certificate Mapper that should be created. The value for *type* can be one of `custom`, `fingerprint`, `subject-attribute-to-user-attribute`, `subject-dn-to-user-attribute`, or `subject-equals-dn`.

`create-connection-handler`
Creates connection handlers. Suboptions are as follows:

`--handler-name` *name*. The name of the new connection handler.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t, --type` *type*. The type of Connection Handler that should be created. The value for *type* can be one of `custom`, `jmx`, `ldap`, `snmp`, or `ldif`.

`create-debug-target`
Creates debug targets. Suboptions are as follows:

`--publisher-name` *name*. The name of the debug log publisher.

`--target-name` *java-name*. The name of the new debug target, which will also be used as the value for the `debug-scope` property. The fully-qualified Oracle Unified Directory Java package, class, or method affected by the settings in this target definition. Use the hash symbol (`#`) to separate the class name and the method name (for example, `org.opends.server.core.DirectoryServer#startUp`).

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`create-extended-operation-handler`
This command is not supported for the proxy.

Creates extended operation handlers. Suboptions are as follows:

`--handler-name` *name*. The name of the new extended operation handler.

`--set property:value`. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t, --type` *type*. The type of Extended Operation handler that should be created. The value for *type* can be one of `cancel`, `custom`, `get-connection-id`, `get-symmetric-key`, `password-modify`, `password-policy-state`, `start-tls`, or `who-am-i`.

`create-identity-mapper`
Creates identity mappers. Suboptions are as follows:

--mapper-name *name*. The name of the new identity mapper.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t, --type` *type*. The type of Identity Mapper that should be created. The value for *type* can be one of `custom`, `exact-match`, or `match-and-replace`.

`create-log-publisher`
Creates log publishers. Suboptions are as follows:

`--publisher-name` *name*. The name of the new log publisher.

`--set` *property*:*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t, --type` *type*. The type of Log Publisher that should be created. The value for *type* can be one of `custom-access, custom-debug, custom-error, file-based-access`, `file-based-debug`, or `file-based-error`.

`create-log-retention-policy`
Creates Log Retention Policies. Suboptions are as follows:

`--policy-name` *name*. The name of the new log retention policy.

`--set` *property*:*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t, --type` *type*. The type of Log Retention Policy that should be created. The value for *type* can be one of `custom`, `file-count`, `free-disk-space`, or `size-limit`.

`create-log-rotation-policy`
Creates log rotation policies. Suboptions are as follows:

`--policy-name name`. The name of the new log rotation policy.

`--set` *property*:*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t, --type` *type*. The type of Log Rotation Policy that should be created. The value for *type* can be one of `custom, fixed-time`, `size-limit`, or `time-limit`.

`create-workflow-element --type monitor-local-backend`
Creates Workflow Elements. Suboptions are as follows:

`--element-name` *name*. The name of the new Workflow Element.

`--set` *property*:*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t, --type` *type.* The type of Workflow Element that should be created. The value for *type* can be one of `ad-paging, backup-local-backend, db-local-backend, distribution, dn-renaming, eus, eus-context, fa, global-index-local-backend, global-index-replication-changes-local-backend, kerberos-auth-provider, ldif-local-backend, load-balancing, memory-local-backend, monitor-local-backend, null-local-backend, pass-through-authentication, plugin, proxy-ldap, rdn-changing, transformations, trust-store-local-backend.`

`create-network-group`
Creates network groups. Suboptions are as follows:

`--group-name` *name*. The name of the new network group.

`--set` *property*:*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`create-network-group-qos-policy`
Creates network group resource limits. Suboptions are as follows:

`--group-name` *name*. The name of the network group.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t, --type` *type*. The type of Quality of Service Policy that should be created. The value for *type* can be one of the following `affinity`, `referral`, `request-filtering`, or `resource-limits`.

`create-workflow`
Creates workflows. Suboptions are as follows:

`--workflow-name` *name*. The name of the new workflow. This name will also be used as The value for the `workflow-id` property.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`delete-alert-handler`
Deletes alert handlers. Suboptions are as follows:

`--handler-name` *name*. The name of the alert handler.

`-f,--force`. Ignore nonexistent alert handlers.

`delete-certificate-mapper`
Deletes certificate mappers. Suboptions are as follows:

`--mapper-name` *name*. The name of the certificate mapper.

`-f,--force`. Ignore nonexistent certificate mappers.

`delete-connection-handler`
Deletes connection handlers. Suboptions are as follows:

`--handler-name` *name*. The name of the connection handler.

`-f,--force`. Ignore nonexistent connection handlers.

`delete-debug-target`
Deletes debug targets. Suboptions are as follows:

`--publisher-name` *name*. The name of the debug log publisher.

`--target-name` *name*. The name of the debug target.

`-f,--force`. Ignore nonexistent debug targets.

`delete-extended-operation-handler`
Deletes extended operation handlers. Suboptions are as follows:

`--handler-name` *name*. The name of the extended operation handler.

`-f,--force`. Ignore nonexistent extended operation handlers.

`delete-identity-mapper`
Deletes identity mappers. Suboptions are as follows:

`--mapper-name` *name*. The name of the identity mapper.

`-f,--force`. Ignore nonexistent identity mappers.

`delete-log-publisher`
Deletes log publishers. Suboptions are as follows:

`--publisher-name` *name*. The name of the log publisher.

`-f,--force`. Ignore nonexistent log publishers.

`delete-log-retention-policy`
Deletes Log Retention Policies. Suboptions are as follows:

`--policy-name` *name*. The name of the log retention policy.

`-f,--force`. Ignore nonexistent Log Retention Policies.

`delete-log-rotation-policy`
Deletes log rotation policies. Suboptions are as follows:

`--policy-name` *name*. The name of the log rotation policy.

`-f,--force`. Ignore nonexistent log rotation policies.

`delete-network-group`
Deletes network group. Suboptions are as follows:

`--group-name` *name*. The name of the network group.

`-f,--force`. Ignore nonexistent network groups.

`delete-network-group-qos-policy`
Deletes network group quality of service policy. Suboptions are as follows:

`--group-name` *name*. The name of the network group.

`--policy-type` *name*. The name of the QOS policy.

`-f,--force`. Ignore nonexistent network group resource limits.

`delete-workflow`
Deletes workflow. Suboptions are as follows:

`-f,--force`. Ignore nonexistent workflow.

`--workflow-name` *name*. The name of the workflows.

`delete-workflow-element`
Deletes Workflow Elements. Suboptions are as follows:

`--element-name` *name.* The name of the Workflow Element.

`-f,--force`. Ignore nonexistent workflow elements.

`get-administration-connector-prop`
Shows administration connector properties. Suboptions are as follows:

`--property` *property*. The name of a property to be displayed.

`-E,--record`. Modifies the display output to show one property value per line.

`-z,--unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m,--unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`get-alert-handler-prop`
Shows alert handler properties. Suboptions are as follows:

`--handler-name` *name*. The name of the alert handler.

`--property` *property*. The name of a property to be displayed.

`-E, --record`. Modifies the display output to show one property value per line.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`get-certificate-mapper-prop`
Shows certificate mapper properties. Suboptions are as follows:

`--mapper-name` *name*. The name of the certificate mapper.

`--property` *property*. The name of a property to be displayed.

`-E, --record`. Modifies the display output to show one property value per line.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`get-connection-handler-prop`
Shows connection handler properties. Suboptions are as follows:

`--handler-name` *name*. The name of the connection handler.

`--property` *property*. The name of a property to be displayed.

`-E, --record`. Modifies the display output to show one property value per line.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`get-debug-target-prop`
Shows debug target properties. Suboptions are as follows:

`--publisher-name name`. The name of the debug log publisher.

`--target-name` *name*. The name of the debug target.

`--property` *property*. The name of a property to be displayed.

`-E, --record`. Modifies the display output to show one property value per line.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`get-extended-operation-handler-prop`
Shows extended operation handler properties. Suboptions are as follows:

`--handler-name` *name*. The name of the extended operation handler.

`--property` *property*. The name of a property to be displayed.

`-E, --record`. Modifies the display output to show one property value per line.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`get-global-configuration-prop`
Shows global configuration properties. Suboptions are as follows:

`--property` *property*. The name of a property to be displayed.

`-E, --record`. Modifies the display output to show one property value per line.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`get-identity-mapper-prop`
Shows identity mapper properties. Suboptions are as follows:

`--mapper-name` *name*. The name of the identity mapper.

`--property` *property*. The name of a property to be displayed.

`-E, --record`. Modifies the display output to show one property value per line.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`get-log-publisher-prop`
Shows log publisher properties. Suboptions are as follows:

`--publisher-name` *name*. The name of the log publisher.

`--property` *property*. The name of a property to be displayed.

`-E, --record`. Modifies the display output to show one property value per line.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`get-log-retention-policy-prop`
Shows log retention policy properties. Suboptions are as follows:

`--policy-name` *name*. The name of the log retention policy.

`--property` *property*. The name of a property to be displayed.

`-E, --record`. Modifies the display output to show one property value per line.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`get-log-rotation-policy-prop`
Shows log rotation policy properties. Suboptions are as follows:

`--policy-name` *name*. The name of the log rotation policy.

`--property` *property*. The name of a property to be displayed.

`-E, --record`. Modifies the display output to show one property value per line.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`get-network-group-prop`
Shows network group properties. Suboptions are as follows:

`--group-name` *name*. The name of the network group.

`--property` *property*. The name of a property to be displayed.

`-E, --record`. Modifies the display output to show one property value per line.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`get-network-group-qos-policy-prop`
Shows network group quality of service policy properties. Suboptions are as follows:

`--group-name` *name*. The name of the network group.

`--policy-type` *name*. The name of the quality of service policy.

`--property` *property*. The name of a property to be displayed.

`-E, --record`. Modifies the display output to show one property value per line.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`get-plugin-root-prop`
Shows plugin root properties.

`--property` *property*. The name of a property to be displayed.

`-E, --record`. Modifies the display output to show one property value per line.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`get-root-dse-backend-prop`
Shows root DSE backend properties. Suboptions are as follows:

`--property` *property*. The name of a property to be displayed.

`-E, --record`. Modifies the display output to show one property value per line.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`get-work-queue-prop`
Shows work queue properties. Suboptions are as follows:

`--property` *property*. The name of a property to be displayed.

`-E, --record`. Modifies the display output to show one property value per line.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`get-workflow-prop`
Shows workflow properties. Suboptions are as follows:

`--workflow-name` *name*. The name of the workflow.

`--property` *property*. The name of a property to be displayed.

`-E, --record`. Modifies the display output to show one property value per line.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`list-alert-handlers`
Lists existing alert handlers. Suboptions are as follows:

`--property` *property*. The name of a property to be displayed.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`list-certificate-mappers`
Lists existing certificate mappers. Suboptions are as follows:

`--property` *property*. The name of a property to be displayed.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`list-connection-handlers`
Lists existing connection handlers. Suboptions are as follows:

`--property` *property*. The name of a property to be displayed.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`list-debug-targets`
Lists existing debug targets. Suboptions are as follows:

`--publisher-name` *name*. The name of the Debug Log Publisher.

`--property` *property*. The name of a property to be displayed.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`list-extended-operation-handlers`
Lists existing extended operation handlers. Suboptions are as follows:

`--property` *property*. The name of a property to be displayed.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`list-identity-mappers`
Lists existing identity mappers. Suboptions are as follows:

`--property` *property*. The name of a property to be displayed.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`list-log-publishers`
Lists existing log publishers. Suboptions are as follows:

`--property` *property*. The name of a property to be displayed.

`-z,--unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m,--unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`list-log-retention-policies`
Lists existing log retention policies. Suboptions are as follows:

`--property` *property*. The name of a property to be displayed.

`-z,--unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m,--unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`list-log-rotation-policies`
Lists existing log rotation policies. Suboptions are as follows:

`--property` *property*. The name of a property to be displayed.

`-z,--unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m,--unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`list-network-group-qos-policies`
Lists existing network group QOS policies. Suboptions are as follows:

`--group-name` *name*. The name of the Network Group.

`--property` *property*. The name of a property to be displayed.

`-z,--unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m,--unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`list-network-groups`
Lists existing network groups. Suboptions are as follows:

`--property` *property*. The name of a property to be displayed.

`-z,--unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m,--unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`list-workflow-elements`
Lists existing Workflow Elements. Suboptions are as follows:

`--property` *property*. The name of a property to be displayed.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`list-workflows`
Lists existing workflows. Suboptions are as follows:

`--property` *property*. The name of a property to be displayed.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`set-administration-connector-prop`
Modifies administration connector properties. Suboptions are as follows:

`--set` *property*:*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*:*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*:*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

`set-alert-handler-prop`
Modifies alert handler properties. Suboptions are as follows:

`--handler-name` *name*. The name of the alert handler.

`--set` *property*:*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*:*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*:*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

`set-certificate-mapper-prop`
Modifies certificate mapper properties. Suboptions are as follows:

`--mapper-name` *name.* The name of the certificate mapper.

`--set` *property*:*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*`:`*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*`:`*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

`set-connection-handler-prop`
Modifies connection handler properties. Suboptions are as follows:

`--handler-name` *name.* The name of the connection handler.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*`:`*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*`:`*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

`set-debug-target-prop`
Modifies debug target properties. Suboptions are as follows:

`--publisher-name` *name*. The name of the debug log publisher.

`--target-name` *name*. The name of the debug target.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*`:`*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*`:`*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

`set-extended-operation-handler-prop`
Modifies extended operation handler properties. Suboptions are as follows:

`--handler-name` *name.* The name of the extended operation handler.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*`:`*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*`:`*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

`set-global-configuration-prop`
Modifies global configuration properties. Suboptions are as follows:

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*`:`*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*`:`*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

`set-identity-mapper-prop`
Modifies identity mapper properties. Suboptions are as follows:

`--mapper-name` *name*. The name of the identity mapper.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*`:`*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*`:`*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

`set-log-publisher-prop`
Modifies log publisher properties. Suboptions are as follows:

`--publisher-name` *name*. The name of the log publisher.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*`:`*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*`:`*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

`set-log-retention-policy-prop`
Modifies log retention policy properties. Suboptions are as follows:

`--policy-name` *name*. The name of the log retention policy.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*`:`*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*`:`*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

`set-log-rotation-policy-prop`
Modifies log rotation policy properties. Suboptions are as follows:

`--policy-name` *name*. The name of the log rotation policy.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*`:`*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*`:`*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

`set-network-group-prop`
Modifies network group properties. Suboptions are as follows:

`--group-name` *name*. The name of the network group.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*`:`*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*`:`*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

`set-network-group-qos-policy-prop`
Modifies network group quality of service policy properties. Suboptions are as follows:

`--group-name` *name*. The name of the network group.

`--policy-type` *name*. The name of the QOS policy.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*`:`*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*`:`*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

`set-plugin-root-prop`
Modifies plugin root properties. Suboptions are as follows:

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*`:`*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*`:`*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

`set-root-dse-backend-prop`
Modifies root DSE back end properties. Suboptions are as follows:

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*`:`*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*`:`*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

`set-work-queue-prop`
Modifies work queue properties. Suboptions are as follows:

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*`:`*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*`:`*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

`set-workflow-element-prop`
Modifies Workflow Element properties. Suboptions are as follows:

--element-name name. The name of the Workflow Element.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*`:`*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*`:`*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

`set-workflow-prop`
Modifies workflow properties. Suboptions are as follows:

`--workflow-name` *name*. The name of the workflow.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*`:`*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*`:`*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

### A.2.4.7 Load Balancing Subcommands

The following subcommands configure load balancing for the proxy server.

`create-load-balancing-algorithm`
This command is supported only for the proxy.

Creates load balancing algorithms. Suboptions are as follows:

`--element-name` *name*. The name of the load balancing workflow element.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t, --type` *type*. The type of Load Balancing Algorithm that should be created. The value for *type* can be `failover`, `generic`, `optimal`, `proportional`, `saturation`, or `searchfilter`. The default value is `generic`.

`create-load-balancing-route`
This command is supported only for the proxy.

Creates load balancing routes. Suboptions are as follows:

`--element-name` *name*. The name of the load balancing workflow element.

`--route-name` *name*. The name of the new load balancing route.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t, --type` *type*. The type of Load Balancing Route that should be created. The value for *type* can be `failover`, `generic`, `optimal`, `proportional`, `saturation`, or `searchfilter`. The default value is `generic`.

`create-workflow-element --type load-balancing`
Creates Workflow Elements. Suboptions are as follows:

--element-name name. The name of the new Workflow Element.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t, --type` *type*. The type of Workflow Element that should be created. The value for *type* can be one of `ad-paging, backup-local-backend, db-local-backend,`

distribution, dn-renaming,eus,eus-context, fa,
global-index-local-backend,
global-index-replication-changes-local-backend,
kerberos-auth-provider,ldif-local-backend,load-balancing,
memory-local-backend,monitor-local-backend,null-local-backend,
pass-through-authentication,plugin,proxy-ldap,
rdn-changing,transformations,trust-store-local-backend.

delete-load-balancing-algorithm
Deletes load balancing algorithm. Suboptions are as follows:

--element-name *name*. The name of the load balancing workflow element.

-f,--force. Ignore nonexistent load balancing algorithms.

delete-load-balancing-route
Deletes load balancing routes. Suboptions are as follows:

--element-name *name*. The name of the load balancing workflow element.

--route-name *name*. The name of the load balancing route.

-f,--force. Ignore nonexistent load balancing route.

delete-workflow-element
Deletes Workflow Elements. Suboptions are as follows:

--element-name *name*. The name of the workflow element.

-f,--force. Ignore nonexistent workflow element.

get-load-balancing-algorithm-prop
Shows load balancing algorithm properties. Suboptions are as follows:

--element-name *name*. The name of the load balancing workflow element.

--property *property*. The name of a property to be displayed.

-E,--record. Modifies the display output to show one property value per line.

-z,--unit-size *unit*. Displays size data using the specified unit. The value for *unit* can be one of b, kb, mb, gb, or tb (bytes, kilobytes, megabytes, gigabytes, or terabytes).

-m,--unit-time *unit*. Displays time data using the specified unit. The value for *unit* can be one of ms, s, m, h, d, or w (milliseconds, seconds, minutes, hours, days, or weeks).

get-load-balancing-route-prop
This command is supported only for the proxy.

Shows load balancing route properties. Suboptions are as follows:

--element-name *name*. The name of the load balancing workflow element.

--route-name *name*. The name of the load balancing route.

--property *property*. The name of a property to be displayed.

-E,--record. Modifies the display output to show one property value per line.

-z,--unit-size *unit*. Displays size data using the specified unit. The value for *unit* can be one of b, kb, mb, gb, or tb (bytes, kilobytes, megabytes, gigabytes, or terabytes).

-m,--unit-time *unit*. Displays time data using the specified unit. The value for *unit* can be one of ms, s, m, h, d, or w (milliseconds, seconds, minutes, hours, days, or weeks).

`list-load-balancing-algorithm`
This command is supported only for the proxy.

Lists existing load balancing algorithm. Suboptions are as follows:

`--element-name` *name*. The name of the load balancing workflow element.

`--property` *property*. The name of a property to be displayed.

`-z,--unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m,--unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`list-load-balancing-routes`
This command is supported only for the proxy.

Lists existing load balancing routes. Suboptions are as follows:

`--element-name` *name*. The name of the load balancing workflow element.

`--property` *property*. The name of a property to be displayed.

`-z,--unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m,--unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`list-workflow-elements`
Lists existing Workflow Elements. Suboptions are as follows:

`--property` *property*. The name of a property to be displayed.

`-z,--unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m,--unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`set-load-balancing-algorithm-prop`
This command is supported only for the proxy.

Modifies load-balancing algorithm properties. Suboptions are as follows:

`--element-name` *name*. The name of the load balancing workflow element.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*`:`*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*`:`*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

`set-load-balancing-route-prop`
This command is supported only for the proxy.

Modifies load balancing route properties. Suboptions are as follows:

`--element-name` *name*. The name of the load balancing workflow element.

`--route-name` *name*. The name of the load balancing route.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*`:`*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*`:`*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

`set-workflow-element-prop`
Modifies Workflow Element properties. Suboptions are as follows:

`--element-name` *name*. The name of the workflow element.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*`:`*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*`:`*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

### A.2.4.8 Local Storage Subcommands

`create-account-status-notification-handler`
Creates account status notification handlers. Suboptions are as follows:

`--handler-name` *name*. The name of the new account status notification handler.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t, --type` *type*. The type of Account Status Notification Handler that should be created. The value for *type* can be one of `custom`, `error-log`, or `smtp`.

`create-workflow-element --`*type* `backup-local-backend`
Creates Workflow Elements. Suboptions are as follows:

`--element-name` *name*. The name of the new Workflow Element.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t,--type` *type*. The type of Workflow Element that should be created. The value for *type* can be one of `ad-paging, backup-local-backend, db-local-backend, distribution, dn-renaming, eus, eus-context, fa, global-index-local-backend, global-index-replication-changes-local-backend, kerberos-auth-provider, ldif-local-backend, load-balancing, memory-local-backend, monitor-local-backend, null-local-backend, pass-through-authentication, plugin, proxy-ldap, rdn-changing, transformations, trust-store-local-backend.`

`create-workflow-element --`*`type`* `db-local-backend`
Creates Workflow Elements. Suboptions are as follows:

`--element-name` *name*. The name of the new Workflow Element.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t,--type` *type*. The type of Workflow Element that should be created. The value for *type* can be one of `ad-paging, backup-local-backend, db-local-backend, distribution, dn-renaming, eus, eus-context, fa, global-index-local-backend, global-index-replication-changes-local-backend, kerberos-auth-provider, ldif-local-backend, load-balancing, memory-local-backend, monitor-local-backend, null-local-backend, pass-through-authentication, plugin, proxy-ldap, rdn-changing, transformations, trust-store-local-backend.`

`create-entry-cache`
Creates entry caches. Suboptions are as follows:

`--cache-name` *name*. The name of the new Entry Cache.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t,--type` *type*. The type of Entry Cache that should be created. The value for *type* can be one of `custom`, `fifo`, `file-system`, or `soft-reference`.

`create-group-implementation`
This command is not supported for the proxy.

Creates group implementations. Suboptions are as follows:

`--implementation-name` *name*. The name of the new group implementation.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t, --type` *type*. The type of Group Implementation that should be created. The value for *type* can be one of `dynamic`, `static`, or `virtual-static`.

`create-workflow-element --`*`type`* `ldif-local-backend`
Creates Workflow Elements. Suboptions are as follows:

`--element-name` *name*. The name of the new Workflow Element.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t,--type` *type*. The type of Workflow Element that should be created. The value for *type* can be one of `ad-paging, backup-local-backend, db-local-backend, distribution, dn-renaming, eus, eus-context, fa, global-index-local-backend, global-index-replication-changes-local-backend, kerberos-auth-provider, ldif-local-backend, load-balancing, memory-local-backend, monitor-local-backend, null-local-backend, pass-through-authentication, plugin, proxy-ldap, rdn-changing, transformations, trust-store-local-backend.`

`create-local-db-index`
Creates local DB indexes. Suboptions are as follows:

`--element-name` *name*. The name of the local DB back end workflow element.

`--index-name` *name*. The name of the new local DB index, which is also used as the value for the `attribute` property. This specifies the name of the attribute for which the index is to be maintained.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`create-local-db-vlv-index`
Creates local DB VLV indexes. Suboptions are as follows:

`--element-name` *name*. The name of the local DB back end workflow element.

`--index-name` *name*. The name of the new local DB VLV index, which is also used as the value of the `name` property. This property specifies a unique name for this VLV index.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`create-workflow-element --`*type* `memory-local-backend`
Creates Workflow Elements. Suboptions are as follows:

`--element-name` *name*. The name of the new Workflow Element.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t,--type` *type*. The type of Workflow Element that should be created. The value for *type* can be one of `ad-paging, backup-local-backend, db-local-backend, distribution, dn-renaming, eus, eus-context, fa, global-index-local-backend, global-index-replication-changes-local-backend, kerberos-auth-provider, ldif-local-backend, load-balancing, memory-local-backend, monitor-local-backend, null-local-backend, pass-through-authentication, plugin, proxy-ldap, rdn-changing, transformations, trust-store-local-backend.`

`create-workflow-element --`*type* `null-local-backend`
Creates Workflow Elements. Suboptions are as follows:

`--element-name` *name*. The name of the new Workflow Element.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t,--type` *type*. The type of Workflow Element that should be created. The value for *type* can be one of `ad-paging, backup-local-backend, db-local-backend, distribution, dn-renaming, eus, eus-context, fa, global-index-local-backend, global-index-replication-changes-local-backend, kerberos-auth-provider, ldif-local-backend, load-balancing, memory-local-backend, monitor-local-backend, null-local-backend, pass-through-authentication, plugin, proxy-ldap, rdn-changing, transformations, trust-store-local-backend.`

`create-password-generator`
Creates password generators. Suboptions are as follows:

`--generator-name` *name*. The name of the new password generator.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t,--type` *type*. The type of Password Generator that should be created. The value for *type* can be one of `custom` or `random`.

`create-password-policy`
Creates Password Policies. Suboptions are as follows:

`--policy-name` *name*. The name of the new Password Policy.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`create-plugin --`*type* `password-policy-import`
Creates Plugins. Suboptions are as follows:

`--plugin-name` *name*. The name of the new Plugin.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t,--type` *type*. The type of Plugin that should be created. The value for *type* can be one of `dsee-gateway, password-policy-import, referential-integrity, seven-bit-clean, unique-attribute.`

`create-password-storage-scheme`
Creates password storage schemes. Suboptions are as follows:

`--scheme-name name`. The name of the new password storage scheme.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t,--type` *type*. The type of Password Storage scheme that should be created. The value for *type* can be one of `aes, base64, blowfish, clear, crypt, custom,`

`md5`, `rc4`, `salted-md5`, `salted-sha1`, `salted-sha256`, `salted-sha384`, `salted-sha512`, `sha1`, or `triple-des`.

`create-password-validator`
Creates password validators. Suboptions are as follows:

`--validator-name` *name*. The name of the new password validator.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t,--type` *type*. The type of Password Validator that should be created. The value for *type* can be one of `attribute-value`, `character-set`, `custom`, `dictionary`, `length-based`, `repeated-characters`, `similarity-based`, or `unique-characters`.

`create-plugin --`*type* `referential-integrity`
Creates Plugins. Suboptions are as follows:

`--plugin-name` *name*. The name of the new Plugin.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t,--type` *type*. The type of Plugin that should be created. The value for *type* can be one of `dsee-gateway`, `password-policy-import`, `referential-integrity`, `seven-bit-clean`, `unique-attribute`.

`create-plugin --`*type* `seven-bit-clean`
Creates Plugins. Suboptions are as follows:

`--plugin-name` *name*. The name of the new Plugin.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t,--type` *type*. The type of Plugin that should be created. The value for *type* can be one of `dsee-gateway`, `password-policy-import`, `referential-integrity`, `seven-bit-clean`, `unique-attribute`.

`create-plugin --`*type* `unique-attribute`
Creates Plugins. Suboptions are as follows:

`--plugin-name` *name*. The name of the new Plugin.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t,--type` *type*. The type of Plugin that should be created. The value for *type* can be one of `dsee-gateway`, `password-policy-import`, `referential-integrity`, `seven-bit-clean`, `unique-attribute`.

`create-virtual-attribute`
This command is not supported for the proxy.

Creates virtual attributes. Suboptions are as follows:

`--name` *name*. The name of the new virtual attribute.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t,--type` *type*. The type of Virtual Attribute that should be created. The value for *type* can be one of `collective-attribute-subentries, custom, entry-dn,entry-uuid, governing-structure-rule, has-subordinates, is-member-of, member, nsuniqueid, num-subordinates, orclguid, password-policy-subentry, proximity, structural-object-class, subschema-subentry, user-defined.`

`delete-account-status-notification-handler`
Deletes account status notification handlers. Suboptions are as follows:

`--handler-name` *name*. The name of the account status notification handler.

`-f,--force`. Ignore nonexistent account status notification handlers.

`delete-entry-cache`
Deletes entry caches. Suboptions are as follows:

`--cache-name` *name*. The name of the Entry Cache.

`-f,--force`. Ignore nonexistent entry cache.

`delete-group-implementation`
This command is not supported for the proxy.

Deletes group implementations. Suboptions are as follows:

`--implementation-name` *name*. The name of the group implementation.

`-f,--force`. Ignore nonexistent group implementations.

`delete-local-db-index`
Deletes local DB indexes. Suboptions are as follows:

`--element-name` *name*. The name of the local DB back end workflow element.

`--index-name` *name*. The name of the local DB index.

`-f,--force`. Ignore nonexistent local DB indexes.

`delete-local-db-vlv-index`
Deletes local DB VLV indexes. Suboptions are as follows:

`--element-name` *name*. The name of the local DB back end workflow element.

`--index-name` *name*. The name of the local DB VLV index.

`-f,--force`. Ignore nonexistent local DB VLV indexes.

`delete-password-generator`
Deletes password generators. Suboptions are as follows:

`--generator-name` *name*. The name of the password generator.

`-f,--force`. Ignore nonexistent password generators.

`delete-password-policy`
Deletes password policies. Suboptions are as follows:

`--policy-name` *name*. The name of the password policy.

`-f,--force`. Ignore nonexistent password policies.

`delete-password-storage-scheme`
Deletes password storage schemes. Suboptions are as follows:

`--scheme-name` *name*. The name of the password storage scheme.

`-f,--force`. Ignore nonexistent password storage schemes.

`delete-password-validator`
Deletes password validators. Suboptions are as follows:

`--validator-name` *name*. The name of the password validator.

`-f,--force`. Ignore nonexistent password validators.

`delete-plugin`
Deletes Plugins. Suboptions are as follows:

`--plugin-name` *name*. The name of the Plugin.

`-f,--force`. Ignore nonexistent Plugins.

`delete-virtual-attribute`
This command is not supported for the proxy.

Deletes virtual attributes. Suboptions are as follows:

`--name name`. The name of the virtual attribute.

`-f,--force`. Ignore nonexistent virtual attributes.

`delete-workflow-element`
Deletes Workflow Elements. Suboptions are as follows:

`--element-name` *name*. The name of the Workflow Element.

`-f,--force`. Ignore nonexistent Workflow Elements.

`get-account-status-notification-handler-prop`
Shows account status notification handler properties. Suboptions are as follows:

`--handler-name` *name*. The name of the account status notification handler.

`--property` *property*. The name of a property to be displayed.

`-E,--record`. Modifies the display output to show one property value per line.

`-z,--unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m,--unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`get-entry-cache-prop`
Shows entry cache properties. Suboptions are as follows:

`--cache-name` *name*. The name of the entry cache.

`--property` *property*. The name of a property to be displayed.

`-E,--record`. Modifies the display output to show one property value per line.

`-z,--unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`get-group-implementation-prop`
This command is not supported for the proxy.

Shows group implementation properties. Suboptions are as follows:

`--implementation-name` *name*. The name of the group implementation.

`--property` *property*. The name of a property to be displayed.

`-E, --record`. Modifies the display output to show one property value per line.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`get-local-db-index-prop`
Shows local DB index properties. Suboptions are as follows:

`--element-name` *name*. The name of the local DB back end workflow element.

`--index-name` *name*. The name of the local DB index.

`--property` *property*. The name of a property to be displayed.

`-E, --record`. Modifies the display output to show one property value per line.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`get-local-db-vlv-index-prop`
Shows the local DB VLV index properties. Suboptions are as follows:

`--element-name` *name*. The name of the local DB back end.

`--index-name` *name*. The name of the local DB VLV index.

`--property` *property*. The name of a property to be displayed.

`-E, --record`. Modifies the display output to show one property value per line.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`get-password-generator-prop`
Shows password generator properties. Suboptions are as follows:

`--generator-name` *name*. The name of the password generator.

`--property` *property*. The name of a property to be displayed.

`-E, --record`. Modifies the display output to show one property value per line.

`-z,--unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m,--unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`get-password-policy-prop`
Shows password policy properties. Suboptions are as follows:

`--policy-name` *name*. The name of the password policy.

`--property` *property*. The name of a property to be displayed.

`-E,--record`. Modifies the display output to show one property value per line.

`-z,--unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m,--unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`get-password-storage-scheme-prop`
Shows password storage scheme properties. Suboptions are as follows:

`--scheme-name` *name*. The name of the password storage scheme.

`--property` *property*. The name of a property to be displayed.

`-E,--record`. Modifies the display output to show one property value per line.

`-z,--unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m,--unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`get-password-validator-prop`
Shows password validator properties. Suboptions are as follows:

`--validator-name` *name*. The name of the password validator.

`--property` *property*. The name of a property to be displayed.

`-E,--record`. Modifies the display output to show one property value per line.

`-z,--unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m,--unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`get-virtual-attribute-prop`
This command is not supported for the proxy.

Shows virtual attribute properties. Suboptions are as follows:

`--name` *name*. The name of the virtual attribute.

`--property` *property*. The name of a property to be displayed.

`-E,--record`. Modifies the display output to show one property value per line.

`-z,--unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m,--unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`list-account-status-notification-handlers`
Lists existing account status notification handlers. Suboptions are as follows:

`--property` *property*. The name of a property to be displayed.

`-z,--unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m,--unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`list-entry-caches`
Lists existing entry caches. Suboptions are as follows:

`--property` *property*. The name of a property to be displayed.

`-z,--unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m,--unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`list-group-implementations`
This command is not supported for the proxy.

Lists existing group implementations. Suboptions are as follows:

`--property` *property*. The name of a property to be displayed.

`-z,--unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m,--unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`list-local-db-indexes`
Lists existing local DB indexes. Suboptions are as follows:

`--element-name` *name*. The name of the DB Local Backend Workflow Element.

`--property` *property*. The name of a property to be displayed.

`-z,--unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m,--unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`list-local-db-vlv-indexes`
Lists existing local DB VLV indexes. Suboptions are as follows:

`--element-name` *name*. The name of the DB Local Backend Workflow Element.

`--property` *property*. The name of a property to be displayed.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`list-password-generators`
Lists existing password generators. Suboptions are as follows:

`--property` *property*. The name of a property to be displayed.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`list-password-policies`
Lists existing password policies. Suboptions are as follows:

`--property` *property*. The name of a property to be displayed.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`list-password-storage-schemes`
Lists existing password storage schemes. Suboptions are as follows:

`--property` *property*. The name of a property to be displayed.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`list-password-validators`
Lists existing password validators. Suboptions are as follows:

`--property` *property*. The name of a property to be displayed.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`list-plugins`
Lists existing Plugins. Suboptions are as follows:

`--property` *property*. The name of a property to be displayed.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`list-virtual-attributes`
This command is not supported for the proxy.

Lists existing virtual attributes. Suboptions are as follows:

`--property` *property*. The name of a property to be displayed.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`list-workflow-elements`
Lists existing Workflow Elements. Suboptions are as follows:

`--property` *property*. The name of a property to be displayed.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`set-account-status-notification-handler-prop`
Modifies account status notification handler properties. Suboptions are as follows:

`--handler-name` *name*. The name of the account status notification handler.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*`:`*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*`:`*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

`set-entry-cache-prop`
Modifies Entry Cache properties. Suboptions are as follows:

`--cache-name` *name*. The name of the Entry Cache.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*`:`*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*`:`*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

`set-group-implementation-prop`
This command is not supported for the proxy.

Modifies group implementation properties. Suboptions are as follows:

`--implementation-name` *name.* The name of the group implementation.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*`:`*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*`:`*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

`set-local-db-index-prop`
Modifies local DB Index properties. Suboptions are as follows:

`--element-name` *name*. The name of the local DB back end workflow element.

`--index-name` *name*. The name of the local DB Index.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*`:`*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*`:`*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

`set-local-db-vlv-index-prop`
Modifies local DB VLV Index properties. Suboptions are as follows:

`--element-name` *name*. The name of the local DB back end workflow element.

`--index-name` *name*. The name of the local DB VLV Index.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*`:`*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*`:`*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

`set-password-generator-prop`
Modifies password generator properties. Suboptions are as follows:

`--generator-name` *name*. The name of the password generator.

`--set` *property*:*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*:*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*:*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

`set-password-policy-prop`
Modifies password policy properties. Suboptions are as follows:

`--policy-name` *name*. The name of the password policy.

`--set` *property*:*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*:*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*:*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

`set-password-storage-scheme-prop`
Modifies password storage scheme properties. Suboptions are as follows:

`--scheme-name` *name*. The name of the password storage scheme.

`--set` *property*:*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*:*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*:*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

`set-password-validator-prop`
Modifies password validator properties. Suboptions are as follows:

`--validator-name` *name*. The name of the password validator.

`--set` *property*:*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*:*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*`:`*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

`set-plugin-prop`
Modifies Plugin properties. Suboptions are as follows:

`--plugin-name` *name*. The name of the Plugin.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*`:`*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*`:`*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

`set-virtual-attribute-prop`
This command is not supported for the proxy.

Modifies virtual attribute properties. Suboptions are as follows:

`--name` *name.* The name of the virtual attribute.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*`:`*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*`:`*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

`set-workflow-element-prop`
Modifies Workflow Element properties. Suboptions are as follows:

`--element-name` *name.* The name of the Workflow Element.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*`:`*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*`:`*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

### A.2.4.9 Miscellaneous Workflow Elements Subcommands

This section describes the subcommands for various workflow operations.

`create-workflow-element --type ad-paging`
This command creates Ad Paging Workflow Elements. Suboptions are as follows:

`--element-name` *name.* The name of the new Workflow Element.

`--set` *property*:*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t, --type` *type*. The type of Workflow Element that should be created. The value for *type* can be one of `ad-paging, backup-local-backend, db-local-backend,distribution, dn-renaming, eus, eus-context, fa, global-index-local-backend, global-index-replication-changes-local-backend, kerberos-auth-provider, ldif-local-backend, load-balancing, memory-local-backend, monitor-local-backend, null-local-backend, pass-through-authentication, plugin, proxy-ldap, rdn-changing, transformations, trust-store-local-backend.`

`create-workflow-element --type eus-context`
This command creates Eus Context Workflow Elements. Suboptions are as follows:

`--element-name` *name.* The name of the new Workflow Element.

`--set` *property*:*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t, --type` *type*. The type of Workflow Element that should be created. The value for *type* can be one of `ad-paging,backup-local-backend,db-local-backend,distribution, dn-renaming,eus,eus-context,fa,global-index-local-backend,global-index-replication-changes-local-backend, kerberos-auth-provider, ldif-local-backend, load-balancing, memory-local-backend, monitor-local-backend, null-local-backend, pass-through-authentication, plugin, proxy-ldap, rdn-changing, transformations, trust-store-local-backend.`

`create-workflow-element --type eus`
This command creates Eus Workflow Elements. Suboptions are as follows:

`--element-name` *name.* The name of the new Workflow Element.

`--set` *property*:*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t, --type` *type*. The type of Workflow Element that should be created. The value for *type* can be one of `ad-paging, backup-local-backend, db-local-backend, distribution, dn-renaming, eus, eus-context, fa, global-index-local-backend, global-index-replication-changes-local-backend, kerberos-auth-provider, ldif-local-backend, load-balancing, memory-local-backend, monitor-local-backend, null-local-backend, pass-through-authentication, plugin, proxy-ldap, rdn-changing, transformations, trust-store-local-backend.`

`create-workflow-element --type fa`
This command creates Fa Workflow Elements. Suboptions are as follows:

`--element-name` *name.* The name of the new Workflow Element.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t, --type` *type*. The type of Workflow Element that should be created. The value for *type* can be one of `ad-paging, backup-local-backend, db-local-backend,distribution, dn-renaming, eus, eus-context, fa, global-index-local-backend, global-index-replication-changes-local-backend, kerberos-auth-provider, ldif-local-backend, load-balancing, memory-local-backend, monitor-local-backend, null-local-backend, pass-through-authentication, plugin, proxy-ldap, rdn-changing, transformations, trust-store-local-backend.`

`create-workflow-element --type kerberos-auth-provider`
This command creates Kerberos Auth Provider Workflow Elements. Suboptions are as follows:

`--element-name` *name.* The name of the new Workflow Element.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t, --type` *type*. The type of Workflow Element that should be created. The value for *type* can be one of `ad-paging, backup-local-backend, db-local-backend, distribution, dn-renaming, eus, eus-context, fa, global-index-local-backend, global-index-replication-changes-local-backend, kerberos-auth-provider, ldif-local-backend, load-balancing, memory-local-backend, monitor-local-backend, null-local-backend, pass-through-authentication, plugin, proxy-ldap, rdn-changing, transformations, trust-store-local-backend.`

`create-workflow-element --type pass-through-authentication`
This command creates Pass Through Authentication Workflow Elements. Suboptions are as follows:

`--element-name` *name.* The name of the new Workflow Element.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t, --type` *type*. The type of Workflow Element that should be created. The value for *type* can be one of `ad-paging, backup-local-backend, db-local-backend, distribution, dn-renaming, eus, eus-context, fa, global-index-local-backend, global-index-replication-changes-local-backend, kerberos-auth-provider, ldif-local-backend, load-balancing, memory-local-backend, monitor-local-backend, null-local-backend, pass-through-authentication, plugin, proxy-ldap, rdn-changing, transformations, trust-store-local-backend.`

`create-workflow-element --type plugin`
This command creates Plugin Workflow Elements. Suboptions are as follows:

`--element-name` *name.* The name of the new Workflow Element.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t, --type` *type*. The type of Workflow Element that should be created. The value for *type* can be one of `ad-paging, backup-local-backend, db-local-backend, distribution, dn-renaming, eus, eus-context, fa, global-index-local-backend, global-index-replication-changes-local-backend, kerberos-auth-provider, ldif-local-backend, load-balancing, memory-local-backend, monitor-local-backend, null-local-backend, pass-through-authentication, plugin, proxy-ldap, rdn-changing, transformations, trust-store-local-backend.`

`delete-workflow-element`
This command deletes Workflow Elements. Suboptions are as follows:

`--element-name` *name.* The name of the Workflow Element.

`-f, --force`. Ignore nonexistent Workflow Elements.

`list-workflow-elements`
Lists existing workflow elements. Suboptions are as follows:

`--property` *property*. The name of a property to be displayed.

`-z,--unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m,--unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`set-workflow-element-prop`
Modifies workflow element properties. Suboptions are as follows:

`--element-name` *name*. The name of the workflow element.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*`:`*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*`:`*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

### A.2.4.10 Remote Storage Subcommands

This section describes subcommands for various remote storage operations.

`create-extension --type ldap-server`
This command creates LDAP Server Extensions. Suboptions are as follows:

`--extension-name` *name.* The name of the new extension.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t, --type` *type*. The type of Extension that should be created. The value for *type* can be one of `global-index-catalog, global-index-catalogs-shared-cache,ldap-server`.

`create-workflow-element --type proxy-ldap`
This command creates Proxy LDAP Workflow Elements. Suboptions are as follows:

`--element-name` *name.* The name of the new workflow element.

`--set` *property*:*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t, --type` *type*. The type of Workflow Element that should be created. The value for *type* can be one of `ad-paging, backup-local-backend, db-local-backend, distribution, dn-renaming, eus, eus-context, fa, global-index-local-backend, global-index-replication-changes-local-backend, kerberos-auth-provider, ldif-local-backend, load-balancing, memory-local-backend, monitor-local-backend, null-local-backend, pass-through-authentication, plugin, proxy-ldap, rdn-changing, transformations, trust-store-local-backend`.

`delete-extension`
Deletes extension. Suboptions are as follows:

`--extension-name` *name*. The name of the extension.

`-f,--force`. Ignore nonexistent extensions.

`delete-workflow-element`
Deletes workflow elements. Suboptions are as follows:

`--element-name` *name*. The name of the workflow element.

`-f,--force`. Ignore nonexistent workflow elements.

`list-extensions`
Lists existing extensions. Suboptions are as follows:

`--property` *property*. The name of a property to be displayed.

`-z,--unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m,--unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`list-workflow-elements`
Lists existing workflow elements. Suboptions are as follows:

`--property` *property*. The name of a property to be displayed.

`-z,--unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m,--unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`set-extension-prop`
This command modifies Extension properties. Suboptions are as follows:

`--extension-name` *name*. The name of the Extension.

`--set` *property*:*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*:*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*:*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

`set-workflow-element-prop`
This command modifies Workflow Element properties. Suboptions are as follows:

`--element-name` *name.* The name of the Workflow Element.

`--set` *property*:*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*:*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*:*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

#### A.2.4.11 Replication Subcommands

This section describes subcommands for various replication operations.

`create-plugin --`*type* `dsee-gateway`
Creates Plugins. Suboptions are as follows:

`--plugin-name` *name*. The name of the Plugin.

`--set` *property*:*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t, --type` *type*. The type of Plugin that should be created. The value for *type* can be one of `dsee-gateway,password-policy-import,referential-integrity, seven-bit-clean,unique-attribute.`

`create-gateway-domain`
Creates gateway domains. Suboptions are as follows:

`--plugin-name` *name*. The name of the DSEE gateway plugin.

`--domain-name` *name*. The name of the gateway domain.

`--set` *property*:*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`create-replication-domain`
Creates replication domains. Suboptions are as follows:

`--provider-name` *name*. The name of the multi-master synchronization provider.

`--domain-name` *name*. The name of the new replication domain.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`create-replication-server`
Creates replication servers. Suboptions are as follows:

`--provider-name` *name*. The name of the multi-master synchronization provider.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`create-synchronization-provider`
Creates synchronization providers. Suboptions are as follows:

`--provider-name` *name*. The name of the new synchronization provider.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t,--type` *type*. The type of Synchronization Provider that should be created. The value for *type* can be one of `custom,replication`.

`delete-gateway-domain`
Deletes gateway domains. Suboptions are as follows:

`--plugin-name` *name*. The name of the DSEE gateway plugin.

`--domain-name` *name*. The name of the gateway domain.

-f, --force. Ignore nonexistent Gateway Domains.

`delete-plugin`
Deletes Plugins. Suboptions are as follows:

`--plugin-name` *name*. The name of the Plugin.

`-f,--force`. Ignore nonexistent Plugin.

`delete-replication-domain`
Deletes replication domains. Suboptions are as follows:

`--provider-name` *name*. The name of the synchronization provider.

`--domain-name` *name*. The name of the replication domain.

`-f,--force`. Ignore nonexistent replication domains.

`delete-replication-server`
Deletes replication servers. Suboptions are as follows:

`--provider-name` *name*. The name of the synchronization provider.

`-f,--force`. Ignore nonexistent replication servers.

`delete-synchronization-provider`
Deletes synchronization providers. Suboptions are as follows:

`--provider-name` *name*. The name of the synchronization provider.

`-f,--force`. Ignore nonexistent synchronization providers.

`get-external-changelog-domain-prop`
Shows External Changelog Domain properties. Suboptions are as follows:

`--provider-name` *name*. The name of the Replication Synchronization Provider.

`--domain-name` *name*. The name of the Replication Domain.

`--property` *property*. The name of a property to be displayed.

`-E,--record`. Modifies the display output to show one property value per line.

`-z,--unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m,--unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`get-gateway-domain-prop`
Shows gateway domain properties.

`--plugin-name` *name*. The name of the DSEE gateway plugin.

`--domain-name` *name*. The name of the gateway domain.

`--property` *property*. The name of a property to be displayed.

`-E,--record`. Modifies the display output to show one property value per line.

`-z,--unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m,--unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`get-replication-domain-prop`
Shows replication domain properties. Suboptions are as follows:

`--provider-name` *name*. The name of the multi-master synchronization provider.

`--domain-name` *name*. The name of the replication domain.

`--property` *property*. The name of a property to be displayed.

`-E,--record`. Modifies the display output to show one property value per line.

`-z,--unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m,--unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`get-replication-server-prop`
Shows replication server properties. Suboptions are as follows:

`--provider-name name`. The name of the multi-master synchronization provider.

`--property` *property*. The name of a property to be displayed.

`-E, --record`. Modifies the display output to show one property value per line.

`-z,--unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`get-synchronization-provider-prop`
Shows synchronization provider properties. Suboptions are as follows:

`--provider-name` *name*. The name of the synchronization provider.

`--property` *property*. The name of a property to be displayed.

`-E, --record`. Modifies the display output to show one property value per line.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`list-plugins`
Lists existing Plugins. Suboptions are as follows:

`--property` *property*. The name of a property to be displayed.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`list-gateway-domains`
Lists existing gateway domains. Suboptions are as follows.

`--plugin-name` *name*. The name of the DSEE Gateway Plugin.

`--property` *property*. The name of a property to be displayed.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`list-replication-domains`
Lists existing replication domains. Suboptions are as follows:

`--provider-name` *name*. The name of the replication synchronization provider.

`--property` *property*. The name of a property to be displayed.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`list-replication-server`
Lists existing replication server. Suboptions are as follows:

`--provider-name` *name*. The name of the replication synchronization provider.

`--property` *property*. The name of a property to be displayed.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`list-synchronization-providers`
Lists existing synchronization providers. Suboptions are as follows:

`--property` *property*. The name of a property to be displayed.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`set-external-changelog-domain-prop`
Modifies External Changelog Domain properties. Suboptions are as follows:

`--provider-name` *name*. The name of the Replication Synchronization Provider.

`--domain-name` *name*. The name of the Replication Domain.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*`:`*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*`:`*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

`set-gateway-domain-prop`
Modifies gateway domain properties. Suboptions are as follows:

`--plugin-name` *name*. The name of the DSEE Gateway Plugin.

`--domain-name` *name*. The name of the gateway domain.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*`:`*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*`:`*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

`set-plugin-prop`
Modifies Plugin properties. Suboptions are as follows:

`--plugin-name` *name*. The name of the Plugin.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*`:`*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*`:`*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

`set-replication-domain-prop`
Modifies replication domain properties. Suboptions are as follows:

`--provider-name` *name*. The name of the replication synchronization provider.

`--domain-name` *name*. The name of the replication domain.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*`:`*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*`:`*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

`set-replication-server-prop`
Modifies replication server properties. Suboptions are as follows:

`--provider-name` *name*. The name of the replication synchronization provider.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*`:`*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*`:`*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

`set-synchronization-provider-prop`
Modifies synchronization provider properties. Suboptions are as follows:

`--provider-name` *name*. The name of the synchronization provider.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*`:`*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*`:`*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

### A.2.4.12 Schema Subcommands

This section describes subcommands for various schema operations.

`create-attribute-syntax`
This command is not supported for the proxy.

Creates attribute syntaxes. Suboptions are as follows:

`--syntax-name` *name*. The name of the new attribute syntax.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t,--type` *type*. The type of Attribute Syntax that should be created. The value for *type* can be one of `attribute-type-description`, `directory-string`, `generic`, or `telephone-number`.

`create-matching-rule`
This command is not supported for the proxy.

Creates matching rules. Suboptions are as follows:

`--rule-name` *name*. The name of the new matching rule.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t,--type` *type*. The type of Matching Rule that should be created. The value for *type* can be one of `collation` or `generic`.

`delete-attribute-syntax`
This command is not supported for the proxy.

Deletes attribute syntaxes. Suboptions are as follows:

`--syntax-name` *name*. The name of the attribute syntax.

`-f,--force`. Ignore nonexistent attribute syntaxes.

`delete-matching-rule`
This command is not supported for the proxy.

Deletes matching rules. Suboptions are as follows:

`--rule-name` *name*. The name of the matching rule.

`-f,--force`. Ignore nonexistent matching rules.

`get-attribute-syntax-prop`
This command is not supported for the proxy.

Shows attribute syntax properties. Suboptions are as follows:

`--syntax-name` *name*. The name of the attribute syntax.

`--property` *property*. The name of a property to be displayed.

`-E,--record`. Modifies the display output to show one property value per line.

`-z,--unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m,--unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`get-matching-rule-prop`
This command is not supported for the proxy.

Shows matching rule properties. Suboptions are as follows:

`--rule-name` *name*. The name of the matching rule.

`--property` *property*. The name of a property to be displayed.

`-E,--record`. Modifies the display output to show one property value per line.

`-z,--unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m,--unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`list-attribute-syntaxes`
This command is not supported for the proxy.

Lists existing attribute syntaxes. Suboptions are as follows:

`--property` *property*. The name of a property to be displayed.

`-z,--unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m,--unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`list-matching-rules`
This command is not supported for the proxy.

Lists existing matching rules. Suboptions are as follows:

`--property` *property*. The name of a property to be displayed.

`-z,--unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m,--unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`set-attribute-syntax-prop`
This command is not supported for the proxy.

Modifies attribute syntax properties. Suboptions are as follows:

`--syntax-name` *name*. The name of the attribute syntax.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*:*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*:*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

`set-matching-rule-prop`
This command is not supported for the proxy.

Modifies matching rule properties. Suboptions are as follows:

`--rule-name` *name*. The name of the matching rule.

`--set` *property*:*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*:*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*:*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

### A.2.4.13 Security Subcommands

`create-key-manager-provider`
Creates key manager providers. Suboptions are as follows:

`--provider-name` *name*. The name of the new key manager provider.

`--set` *property*:*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t,--type type`. The type of Key Manager Provider that should be created. The value for *type* can be one of `file-based`, `custom`, or `pkcs11`.

PKCS#11 is not supported for a proxy server instance.

`create-sasl-mechanism-handler`
This command is not supported for the proxy.

Creates SASL mechanism handlers. Suboptions are as follows:

`--handler-name` *name*. The name of the new SASL mechanism handler.

`--set` *property*:*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t,--type` *type*. The type of SASL Mechanism Handler that should be created. The value for *type* can be one of `anonymous`, `cram-md5`, `digest-md5`, `external`, `custom`, `gssapi`, or `plain`.

`create-trust-manager-provider`
Creates trust manager providers. Suboptions are as follows:

`--provider-name` *name*. The name of the new trust manager provider.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t,--type` *type*. The type of Trust Manager Provider that should be created. The value for *type* can be one of `blind`, `file-based`, or `custom`.

`create-workflow-element --`*type* `trust-store-local-backend`
Creates Workflow Elements. Suboptions are as follows:

`--element-name` *name*. The name of the new Workflow Element.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t,--type` *type*. The type of Workflow Element that should be created. The value for *type* can be one of `ad-paging, backup-local-backend, db-local-backend,distribution, dn-renaming, eus, eus-context, fa, global-index-local-backend, global-index-replication-changes-local-backend, kerberos-auth-provider, ldif-local-backend, load-balancing, memory-local-backend, monitor-local-backend, null-local-backend, pass-through-authentication, plugin, proxy-ldap, rdn-changing, transformations, trust-store-local-backend`.

`delete-key-manager-provider`
Deletes key manager providers. Suboptions are as follows:

`--provider-name` *name.* The name of the Key Manager provider.

`-f,--force`. Ignore nonexistent Key Manager providers.

`delete-sasl-mechanism-handler`
This command is not supported for the proxy.

Deletes SASL mechanism handlers. Suboptions are as follows:

--handler-name *name*. The name of the SASL mechanism handler.

`-f,--force`. Ignore nonexistent SASL mechanism handlers.

`delete-trust-manager-provider`
Deletes trust manager providers. Suboptions are as follows:

`--provider-name` *name*. The name of the trust manager provider.

`-f,--force`. Ignore nonexistent trust manager providers.

`delete-workflow-element`
Deletes Workflow Elements. Suboptions are as follows:

`--element-name` *name*. The name of the Workflow Element.

`-f,--force`. Ignore nonexistent Workflow Elements.

`get-access-control-handler-prop`
Shows access control handler properties. Suboptions are as follows:

`--property` *property*. The name of a property to be displayed.

`-E,--record`. Modifies the display output to show one property value per line.

`-z,--unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m,--unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`get-crypto-manager-prop`
Show crypto manager properties. Suboptions are as follows:

`--property` *property*. The name of a property to be displayed.

`-E,--record`. Modifies the display output to show one property value per line.

`-z,--unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m,--unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`get-key-manager-provider-prop`
Shows key manager provider properties. Suboptions are as follows:

`--provider-name` *name*. The name of the key manager provider.

`--property` *property*. The name of a property to be displayed.

`-E,--record`. Modifies the display output to show one property value per line.

`-z,--unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m,--unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`get-root-dn-prop`
Shows Root DN properties. Suboptions are as follows:

`--property` *property*. The name of a property to be displayed.

`-E,--record`. Modifies the display output to show one property value per line.

`-z,--unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m,--unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`get-sasl-mechanism-handler-prop`
Shows SASL mechanism handler properties. Suboptions are as follows:

`--handler-name` *name*. The name of the SASL mechanism handler.

`--property` *property*. The name of a property to be displayed.

`-E,--record`. Modifies the display output to show one property value per line.

`-z,--unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`get-trust-manager-provider-prop`
Shows trust manager provider properties. Suboptions are as follows:

`--provider-name` *name*. The name of the trust manager provider.

`--property` *property*. The name of a property to be displayed.

`-E, --record`. Modifies the display output to show one property value per line.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`list-key-manager-providers`
Lists existing key manager providers. Suboptions are as follows:

`--property` *property*. The name of a property to be displayed.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`list-sasl-mechanism-handlers`
This command is not supported for the proxy.

Lists existing SASL mechanism handlers. Suboptions are as follows:

`--property` *property*. The name of a property to be displayed.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`list-trust-manager-providers`
Lists existing trust manager providers. Suboptions are as follows:

`--property` *property*. The name of a property to be displayed.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`list-workflow-elements`
Lists existing Workflow Elements. Suboptions are as follows:

`--property` *property*. The name of a property to be displayed.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`set-access-control-handler-prop`
Modifies access control handler properties. Suboptions are as follows:

`--set` *property*:*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*:*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*:*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

`set-crypto-manager-prop`
Modifies crypto manager properties. Suboptions are as follows:

`--set` *property*:*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*:*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*:*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

`set-key-manager-provider-prop`
Modifies key manager provider properties. Suboptions are as follows:

`--provider-name` *name*. The name of the key manager provider.

`--set` *property*:*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*:*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*:*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

`set-root-dn-prop`
Modifies root DN properties. Suboptions are as follows:

`--set` *property*:*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*`:`*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*`:`*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

`set-sasl-mechanism-handler-prop`
This command is not supported for the proxy.

Modifies SASL mechanism handler properties. Suboptions are as follows:

`--handler-name` *name*. The name of the SASL mechanism handler.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*`:`*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*`:`*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

`set-trust-manager-provider-prop`
Modifies trust manager provider properties. Suboptions are as follows:

`--provider-name` *name*. The name of the trust manager provider.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*`:`*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*`:`*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

`set-workflow-element-prop`
Modifies Workflow Element properties. Suboptions are as follows:

`--element-name` *name*. The name of the Workflow Element.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*`:`*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*`:`*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

### A.2.4.14 Virtualization Subcommands

This section describes subcommands for virtualization.

`create-transformation --type add-inbound-attribute`
Creates Add Inbound Attribute Transformations. Suboptions are as follows:

`--transformation-name` *name*. The name of the new transformation.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t,--type` *type*. The type of Transformation that should be created. The value for *type* can be one of `add-inbound-attribute,add-outbound-attribute, filter-inbound-attribute,filter-outbound-attribute, map-attribute`. For more information about each transformation, see Section 11.6.3.2, "Example of Configuring Transformation Using CLI."

`create-transformation --type add-outbound-attribute`
Creates Add Outbound Attribute Transformations. Suboptions are as follows:

`--transformation-name` *name*. The name of the new transformation.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t,--type` *type*. The type of Transformation that should be created. The value for *type* can be one of `add-inbound-attribute,add-outbound-attribute, filter-inbound-attribute,filter-outbound-attribute, map-attribute`. For more information about each transformation, see Section 11.6.3.2, "Example of Configuring Transformation Using CLI."

`create-workflow-element --type dn-renaming`
Creates DN Renaming Workflow Elements. Suboptions are as follows:

`--element-name` *name*. The name of the new workflow element.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t,--type` *type*. The type of Workflow Element that should be created. The value for *type* can be one of `ad-paging, backup-local-backend, db-local-backend,distribution, dn-renaming, eus, eus-context, fa, global-index-local-backend, global-index-replication-changes-local-backend, kerberos-auth-provider, ldif-local-backend, load-balancing, memory-local-backend, monitor-local-backend, null-local-backend, pass-through-authentication, plugin, proxy-ldap, rdn-changing, transformations, trust-store-local-backend.`

`create-transformation --type filter-inbound-attribute`
Creates Filter Inbound Attribute Transformations. Suboptions are as follows:

`--transformation-name` *name*. The name of the new transformation.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t,--type` *type*. The type of Transformation that should be created. The value for *type* can be one of `add-inbound-attribute,add-outbound-attribute, filter-inbound-attribute,filter-outbound-attribute,`

`map-attribute`. For more information about each transformation, see Section 11.6.3.2, "Example of Configuring Transformation Using CLI."

`create-transformation --type filter-outbound-attribute`
Creates Filter Outbound Attribute Transformations. Suboptions are as follows:

`--transformation-name` *name*. The name of the new transformation.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t,--type` *type*. The type of Transformation that should be created. The value for type can be one of `add-inbound-attribute,add-outbound-attribute,filter-inbound-attribute,filter-outbound-attribute, map-attribute`. For more information about each transformation, see Section 11.6.3.2, "Example of Configuring Transformation Using CLI."

`create-transformation --type map-attribute`
Creates Map Attribute Transformations. Suboptions are as follows:

`--transformation-name` *name*. The name of the new transformation.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t,--type` *type*. The type of Transformation that should be created. The value for *type* can be one of `add-inbound-attribute,add-outbound-attribute, filter-inbound-attribute,filter-outbound-attribute, map-attribute`. For more information about each transformation, see Section 11.6.3.2, "Example of Configuring Transformation Using CLI."

`create-workflow-element --type rdn-changing`
Creates RDN Changing Workflow Elements. Suboptions are as follows:

`--element-name` *name*. The name of the workflow element.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t,--type` *type*. The type of Workflow Element that should be created. The value for *type* can be one of `ad-paging, backup-local-backend, db-local-backend,distribution, dn-renaming, eus, eus-context, fa, global-index-local-backend, global-index-replication-changes-local-backend, kerberos-auth-provider, ldif-local-backend, load-balancing, memory-local-backend, monitor-local-backend, null-local-backend, pass-through-authentication, plugin, proxy-ldap, rdn-changing, transformations, trust-store-local-backend`.

`create-transformation`
Creates Transformations. Suboptions are as follows:

`--transformation-name` *name*. The name of the new transformation.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t,--type` *type*. The type of Transformation that should be created. The value for *type* can be one of `add-inbound-attribute,add-outbound-attribute, filter-inbound-attribute,filter-outbound-attribute, map-attribute.`

`create-workflow-element --type transformations`
Creates Transformations Workflow Elements. Suboptions are as follows:

`--element-name` *name*. The name of the new workflow element.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`-t,--type` *type*. The type of Workflow Element that should be created. The value for *type* can be one of `ad-paging, backup-local-backend, db-local-backend,distribution, dn-renaming, eus, eus-context, fa, global-index-local-backend, global-index-replication-changes-local-backend, kerberos-auth-provider, ldif-local-backend, load-balancing, memory-local-backend, monitor-local-backend, null-local-backend, pass-through-authentication, plugin, proxy-ldap, rdn-changing, transformations, trust-store-local-backend.`

`delete-transformation`
Deletes Transformations. Suboptions are as follows:

`--transformation-name` *name*. The name of the transformation.

`-f, --force`. Ignore nonexistent transformation.

`delete-workflow-element`
Deletes Workflow Elements. Suboptions are as follows:

`--element-name` *name*. The name of the workflow element.

`-f, --force`. Ignore nonexistent workflow elements.

`get-transformation-prop`
Shows Transformation properties. Suboptions are as follows:

`--transformation-name` *name*. The name of the transformation element.

`--property` *property*. The name of a property to be displayed.

`-E,--record`. Modifies the display output to show one property value per line.

`-z,--unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m,--unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`list-transformations`
Lists existing Transformations. Suboptions are as follows:

`--property` *property*. The name of a property to be displayed.

`-z,--unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`list-workflow-elements`
Lists existing Workflow Elements. Suboptions are as follows:

`--property` *property*. The name of a property to be displayed.

`-z, --unit-size` *unit*. Displays size data using the specified unit. The value for *unit* can be one of `b`, `kb`, `mb`, `gb`, or `tb` (bytes, kilobytes, megabytes, gigabytes, or terabytes).

`-m, --unit-time` *unit*. Displays time data using the specified unit. The value for *unit* can be one of `ms`, `s`, `m`, `h`, `d`, or `w` (milliseconds, seconds, minutes, hours, days, or weeks).

`set-transformation-prop`
Modifies Transformation properties. Suboptions are as follows:

`--transformation-name` *name*. The name of the transformation element.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*`:`*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*`:`*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

`set-workflow-element-prop`
Modifies Workflow Element properties. Suboptions are as follows:

`--element-name` *name*. The name of the workflow element.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`--reset` *property*. Resets a property back to its default values, where *property* is the name of the property to be reset.

`--add` *property*`:`*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

`--remove` *property*`:`*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed.

### A.2.4.15 Options

The `dsconfig` command accepts an option in either its short form (for example, `-h hostname`) or its long form equivalent (for example, `--hostname hostname`).

`--advanced`
Allows the configuration of advanced components and properties.

### A.2.4.16 LDAP Connection Options

The `dsconfig` command contacts the directory server over SSL through the administration connector (described in Section 14.3, "Managing Administration Traffic to the Server"). These connection options are used to contact the directory server.

`-D, --bindDN bindDN`
Use the bind DN to bind the server. This option is used when performing simple authentication and is not required if SASL authentication is to be used. The default value for this option is `cn=Directory Manager`.

SASL is not supported for a proxy server instance.

`-h, --hostname hostname`
Contact the server on the specified hostname or IP address. If this option is not provided, a default of `localhost` is used.

`-j, --bindPasswordFile filename`
Use the bind password in the specified file when authenticating to the server.

`-K, --keyStorePath path`
Use the client keystore certificate in the specified path.

`-N, --certNickname nickname`
Use the nickname of certificate for SSL client authentication.

`-o, --saslOption name=value`
Use the specified options for SASL authentication.

SASL is not supported for a proxy server instance.

`-p, --port port`
Contact the server at the specified administration port. If this option is not provided, the administration port of the local configuration is used.

`-P, --trustStorePath path`
Use the client trust store certificate in the specified path. This option is not needed if `--trustAll` is used, although a trust store should be used when working in a production environment.

`-u, --keyStorePasswordFile filename`
Use the password in the specified file to access the certificates in the client keystore. This option is only required if `--keyStorePath` is used.

`-U, --trustStorePasswordFile filename`
Use the password in the specified file to access the certificates in the client trust store. This option is only required if `--trustStorePath` is used and the specified trust store requires a password in order to access its contents (most trust stores do not require this).

`-X, --trustAll`
Trust all server SSL certificates that the server presents. This option can be used for convenience and testing purposes, but for security reasons a trust store should be used to determine whether the client should accept the server certificate. If the client and the server are running in the same instance, there is no certificate interaction.

`--connectTimeout {timeout}`
This is used to specify the maximum length of time (in milliseconds) that can be taken to establish a connection. Use `0` to specify no time out. The default value is `30000`.

### A.2.4.17 Command Input/Output Options

`--commandFilePath path`
Specify the full path to the file, where the equivalent non-interactive commands will be written when this command is run in interactive mode.

`--displayCommand`
Display the equivalent non-interactive option in the standard output when this command is run in interactive mode.

`-F, --batchFilePath batchFilePath`
Specifies the path to a file that contains a set of `dsconfig` commands to be executed. This option supports line splitting, backslash ('\'), quotes (") escaped quotes (\") inside a quoted string, and hash for comments ('#').

`-n, --no-prompt`
Use non-interactive mode. If some data in the command is missing, you are not prompted and the command will fail.

`--noPropertiesFile`
Indicate that the command will not use a properties file to get the default command-line options.

`--sortMenuItems`
Allows to sort the menu items if the interactive mode is used. The order is the user locale alphabetic order.

`--propertiesFilePath path`
Specify the path to the properties file that contains the default command-line options.

`-Q, --quiet`
Run in quiet mode. No output will be generated unless a significant error occurs during the process.

`-s, --script-friendly`
Run in "script friendly" mode. Display the output in a format that can be easily parsed by a script.

`-v, --verbose`
Run in verbose mode, displaying diagnostics on standard output.

### A.2.4.18 General Options

`-?, -H, --help`
Display command-line usage information for the command and exit without making any attempt to stop or restart the server.

`-V, --version`
Display the version information for the server and exit rather than attempting to run this command.

### A.2.4.19 Examples

The following examples show how to use the `dsconfig` command. For additional `dsconfig` examples, see Section 14.1, "Managing the Server Configuration With `dsconfig`."

#### *Example A–12 Viewing the Global Help Subcommands and Global Options*

The following command displays the available global help subcommands and global options for the server:

```
$ dsconfig --help
```

#### *Example A–13 Viewing a Component's Subcommand Help Information*

The following command displays subcommands relating to authentication and authorization:

```
$ dsconfig --help-security
```

#### *Example A–14 Viewing Help on an Individual Subcommand*

The following command displays the help information for the `set-distribution-partition-prop` subcommand:

```
$ dsconfig set-distribution-partition-prop --help
```

#### *Example A–15 Displaying a Component's Properties*

The following command displays the properties for `local-db-index`. If `-t` is not specified, the command displays the properties for all components.

```
$ dsconfig list-properties -c local-db-index
Option Types:

 r -- Property value(s) are readable
 w -- Property value(s) are writable
 m -- The property is mandatory
 s -- The property is single-valued
 a -- Administrative action is required for changes to take effect

Component       Type     Property                        Options  Syntax
-------------------------------------------------------------------------------
local-db-index  generic  attribute                       r-ms-    OID
local-db-index  generic  index-entry-limit               rw-sa    INTEGER
local-db-index  generic  index-extensible-matching-rule  rw--a    LOCALE | OID
local-db-index  generic  index-type                      rwm-a    TYPE
```

The following command displays the properties for `crypto-manager`.

```
$ dsconfig list-properties -c crypto-manager
Option Types:

 r -- Property value(s) are readable
 w -- Property value(s) are writable
 m -- The property is mandatory
 s -- The property is single-valued
 a -- Administrative action is required for changes to take effect

Component       Type     Property                     Options  Syntax
-----------------------------------------------------------------------
crypto-manager  generic  key-wrapping-transformation  rw-s-    STRING
crypto-manager  generic  ssl-cert-nickname            rw-sa    STRING
crypto-manager  generic  ssl-cipher-suite             rw---    STRING
crypto-manager  generic  ssl-encryption               rw-s-    BOOLEAN
crypto-manager  generic  ssl-protocol                 rw---    STRING
```

***Example A–16 Parameters Supported by the `-F, --batchFilePath` subcommand***

This example describes the various parameters supported by the `-F, --batchFilePath` subcommand.

Executing the `-F, --batchFilePath` subcommand using the line splitting approach. The file `/tmp/batch` contains the following set of commands:

```
create-workflow-element \
--type db-local-backend \
--set base-dn:cn=myexample,cn=com \
--set enabled:true \
--element-name myBackend
```

Running the `-F, --batchFilePath` subcommand.

```
dsconfig -X -j /path/pwd-file -F /tmp/batch -n
```

Executing the `-F, --batchFilePath` subcommand using quotes (") and escaped quotes (\") inside a quoted string. The file /tmp/batch contains the following set of commands:

```
set-access-control-handler-prop \
--add global-aci:"(targetattr != \"description || mail\") \
(version 3.0; acl \"Allow self entry modification except for \
description and mail attributes\"; allow (write)userdn =\"ldap:///self\";) "
```

Running the `-F, --batchFilePath` subcommand.

```
dsconfig -X  -j /path/pwd-file -F /tmp/batch -n
```

***Example A–17 Using the sortMenuItem Option to Display Information as per Locale***

This example describes how to display information as per the user locale using `--sortMenuItems` subcommand. In this example, the dsconfig command is run with and without the --sortMenuItems subcommand to highlight the difference in the way information is displayed. Step 2 displays the menu in US order, whereas Step 3 displays the menu in French order.

1. Set the desired locale using the following command:

   ```
   export LC_ALL=fr_FR.UTF-8
   ```

2. Run the `dsconfig` command without the `--sortMenuItems` subcommand.

   ```
   dsconfig -j /path/pwd-file

   >>>> SpÃ©cifiez les paramÃ¨tres de connexion LDAP Oracle Unified Directory

   Nom d'hÃ´te ou adresse IP du serveur d'annuaire [nixes] :
   NumÃ©ro de port d'administration du serveur d'annuaire [11444] :
   DN de liaison de l'administrateur [cn=Directory Manager] :

   >>>> Menu principal de la console de configuration Oracle Unified Directory

   Que voulez-vous configurer ?

       1)   Gestionnaire de contrÃ´le      26)  Index VLV de base de donnÃ©es
            d'accÃ¨s                            locale
       2)   Gestionnaire de notification   27)  Editeur de journal
   ```

```
     de statuts de comptes
3)   Connecteur d'administration    28)  StratÃ©gie de conservation de
                                        journal
4)   Gestionnaire d'alertes         29)  StratÃ©gie de rotation des
                                        journaux
5)   Syntaxe d'attribut             30)  RÃ¨gle de correspondance
6)   Mappeur de certificats         31)  Fournisseur de surveillance
7)   Gestionnaire de connexions     32)  Groupe de rÃ©seaux
8)   Gestionnaire de cryptage       33)  StratÃ©gie de qualitÃ© de service
                                        (QoS) du groupe de rÃ©seaux
9)   Cible de dÃ©bogage            34)  GÃ©nÃ©rateur de mot de passe
10)  Algorithme de distribution    35)  StratÃ©gie de mot de passe
11)  Partition de distribution     36)  SchÃ©ma de stockage de mot de
                                        passe
12)  Cache d'entrÃ©e                37)  Valideur de mot de passe
13)  Gestionnaire d'opÃ©rations     38)  Plug-in
     d'extension
14)  Extension                     39)  Racine de plug-in
15)  Domaine de journal des        40)  Domaine de rÃ©plication
     modifications externe
16)  Domaine de passerelle         41)  Serveur de rÃ©plication
17)  Configuration globale         42)  Nom distinctif (DN) racine
18)  Index global                  43)  Back-end de la DSE racine
19)  Domaine de rÃ©plication du     44)  Gestionnaire de mÃ©canisme SASL
     catalogue d'index globaux
20)  ImplÃ©mentation de groupe      45)  Fournisseur de synchronisation
21)  Mappeur d'identitÃ©s           46)  Fournisseur de gestionnaire de
                                        sÃ©curisation
22)  Fournisseur de gestionnaire   47)  Attribut virtuel
     de clÃ©s
23)  Algorithme d'Ã©quilibrage de   48)  File d'attente de travaux
     charge
24)  Route d'Ã©quilibrage de charge  49)  Workflow
25)  Index de base de donnÃ©es       50)  ElÃ©ment de workflow

q)   quit

Entrez votre choix :
```

**3.** Run the `dsconfig` command with the `--sortMenuItems` subcommand.

```
dsconfig -j /path/pwd-file --sortMenuItems

>>>> SpÃ©cifiez les paramÃ¨tres de connexion LDAP Oracle Unified Directory

Nom d'hÃ´te ou adresse IP du serveur d'annuaire [nixes] :
NumÃ©ro de port d'administration du serveur d'annuaire [11444] :
DN de liaison de l'administrateur [cn=Directory Manager] :

>>>> Menu principal de la console de configuration Oracle Unified Directory

Que voulez-vous configurer ?

    1)   Algorithme d'Ã©quilibrage de   26)  Gestionnaire de mÃ©canisme SASL
         charge
    2)   Algorithme de distribution    27)  Gestionnaire de notification de
                                            statuts de comptes
    3)   Attribut virtuel              28)  Groupe de rÃ©seaux
    4)   Back-end de la DSE racine     29)  GÃ©nÃ©rateur de mot de passe
    5)   Cache d'entrÃ©e                30)  ImplÃ©mentation de groupe
    6)   Cible de dÃ©bogage             31)  Index de base de donnÃ©es locale
```

```
     7)   Configuration globale              32)  Index global
     8)   Connecteur d'administration        33)  Index VLV de base de donnÃ©es
                                                  locale
     9)   Domaine de journal des             34)  Mappeur d'identitÃ©s
          modifications externe
     10)  Domaine de passerelle              35)  Mappeur de certificats
     11)  Domaine de rÃ©plication             36)  Nom distinctif (DN) racine
     12)  Domaine de rÃ©plication du          37)  Partition de distribution
          catalogue d'index globaux
     13)  Editeur de journal                 38)  Plug-in
     14)  ElÃ©ment de workflow                39)  Racine de plug-in
     15)  Extension                          40)  Route d'Ã©quilibrage de charge
     16)  File d'attente de travaux          41)  RÃ¨gle de correspondance
     17)  Fournisseur de gestionnaire        42)  SchÃ©ma de stockage de mot de
          de clÃ©s                                  passe
     18)  Fournisseur de gestionnaire        43)  Serveur de rÃ©plication
          de sÃ©curisation
     19)  Fournisseur de surveillance        44)  StratÃ©gie de conservation de
                                                  journal
     20)  Fournisseur de                     45)  StratÃ©gie de mot de passe
          synchronisation
     21)  Gestionnaire d'alertes             46)  StratÃ©gie de qualitÃ© de service
                                                  (QoS) du groupe de rÃ©seaux
     22)  Gestionnaire d'opÃ©rations          47)  StratÃ©gie de rotation des
          d'extension                             journaux
     23)  Gestionnaire de connexions         48)  Syntaxe d'attribut
     24)  Gestionnaire de contrÃ´le           49)  Valideur de mot de passe
          d'accÃ¨s
     25)  Gestionnaire de cryptage           50)  Workflow

q)   quit

Entrez votre choix :
```

### A.2.4.20 Exit Codes

An exit code of 0 indicates that the operation completed successfully. An exit code of 1 or greater indicates that an error occurred during processing.

### A.2.4.21 Using a Properties File

The server supports the use of a *properties file* that passes in any default option values used with the `dsconfig` command. The properties file is convenient when working in different configuration environments, especially in scripted or embedded applications. For more information, see Section A.1.2, "Using a Properties File With Server Commands."

The following options can be stored in a properties file:

- `bindDN`
- `bindPasswordFile`
- `certNickname`
- `hostname`
- `keyStorePasswordFile`
- `keyStorePath`
- `port`

- saslOption

  SASL is not supported for a proxy server instance.
- trustAll
- trustStorePasswordFile
- trustStorePath
- useSSL
- useStartTLS

Entries in the properties file have the following format:

```
toolname.propertyname=propertyvalue
```

For example:

```
dsconfig.trustAll=Yes
```

#### A.2.4.22 Location

- UNIX and Linux: *INSTANCE_DIR*/OUD/bin/dsconfig
- Windows: *INSTANCE_DIR*\OUD\bat\dsconfig.bat

#### A.2.4.23 Related Commands

Section A.2.7, "gicadm"

Section A.2.6, "dsreplication"

### A.2.5 dsjavaproperties

The dsjavaproperties command specifies the JVM version and Java arguments that are used by each server command.

#### A.2.5.1 Synopsis

dsjavaproperties [*options*]

#### A.2.5.2 Description

The dsjavaproperties command can be used to specify the JVM version and Java arguments that are used by each server command. The JVM and Java arguments for each command are specified in a properties file, located at *INSTANCE_DIR*/OUD/config/java.properties. The properties file is not used unless you run the dsjavaproperties command. If you edit the properties file, you must run dsjavaproperties again for the new settings to be taken into account.

dsjavaproperties can be used to specify (among other arguments) whether a command runs using the JVM in -server mode or -client mode. By default, all client applications run in -client mode, and all of the server utilities run in -server mode. Generally, -server mode provides higher throughput than -client mode, at the expense of slightly longer startup times.

For certain commands (import-ldif, export-ldif, backup, and restore) you can also specify different Java arguments (and a different JVM) depending on whether the command is run in online or offline mode.

If the value of the `overwrite-env-java-home` property is set to `false` in the `java.properties` file, the *OPENDS_JAVA_HOME* environment variable takes precedence over the arguments specified in the properties file. Similarly, if the value of the `overwrite-env-java-args` property is set to `false` in the `java.properties` file, the *OPENDS_JAVA_ARGS* environment variable takes precedence over the arguments specified in the properties file.

#### A.2.5.3 Options

The `dsjavaproperties` command accepts an option in either its short form (for example, `-Q`) or their long form equivalent (for example, `--quiet`).

`-Q, --quiet`
Run in quiet mode. Quiet mode does not output progress information to standard output.

`-?, -H, --help`
Display command-line usage information for the command and exit without making any attempt to stop or restart the server.

`-V, --version`
Display the version information for the server and exit rather than attempting to run this command.

#### A.2.5.4 Example

The following example shows how to use the `export–ldif` command.

***Example A–18 Modifying a Script***

This example shows how to change the `export-ldif` script to use a maximum JVM heap size of 256 Mbytes when the command is run with the server online.

1. Edit the *INSTANCE_DIR*`/OUD/config/java.properties` file and set the `export-ldif.online` arguments as follows:

   ```
   export-ldif.online.java-args=-client -Xms8m -Xmx256m
   ```

2. Run the `dsjavaproperties` command for the change to take effect.

   ```
   $ dsjavaproperties
    The script files were successfully updated.  The Oracle Unified Directory
    command-line utilities will use the java properties specified in the
    properties file INSTANCE_DIR/OUD/config/java.properties
   ```

#### A.2.5.5 Exit Codes

An exit code of 0 indicates that the operation completed successfully. A nonzero exit code indicates that an error occurred during processing.

#### A.2.5.6 Location

- UNIX and Linux: *INSTANCE_DIR*`/OUD/bin/dsjavaproperties`
- Windows: *INSTANCE_DIR*`\OUD\bat\dsjavaproperties.bat`

### A.2.6 dsreplication

The `dsreplication` command configures replication between directory servers so that the data of the servers is synchronized.

This command is not supported for the proxy.

### A.2.6.1 Synopsis

`dsreplication` [*subcommands*] [*options*]

### A.2.6.2 Description

The `dsreplication` command can be used to configure replication between directory servers so that the data of the servers is synchronized. First enable replication by using the `enable` subcommand and then initialize the contents of one directory server with the contents of another server by using the `initialize` subcommand.

The `dsreplication` command contacts the server over SSL using the administration connector (see Section 14.3, "Managing Administration Traffic to the Server").

Like the `dsconfig` command, `dsreplication` can be run in interactive mode, which walks you through the replication setup process. To run `dsreplication` in interactive mode, type the command name with no parameters, as shown in the following example:

```
$ dsreplication
What do you want to do?

1)  Enable Replication
2)  Disable Replication
3)  Initialize Replication on one Server
4)  Initialize All Servers
5)  Pre External Initialization
6)  Post External Initialization
7)  Display Replication Status
8)  Purge Historical
9)  Set the trust flag of the Directory Server
10) Enable External Changelog
11) Disable External Changelog

c)  cancel

Enter choice: 1
...
```

To display the equivalent non-interactive command, use the `--displayCommand` or `--commandFilePath` option.

### A.2.6.3 Server Subcommands

The following subcommands are used with the `dsreplication` command.

`disable`
Disable replication on the specified directory server for the specified base DN. This subcommand removes references to the specified server in the configuration of the servers with which this server is replicating data. Suboptions are as follows:

`-D, --bindDN` *bindDN*. The DN used to bind to the server on which replication will be disabled. This option must be used if no global administrator has been defined on the server or if you do not want to remove references in the other replicated servers. The password provided for the global administrator is used when this option is specified.

`-a, --disableAll`. Disable the replication configuration on the specified server. The contents of the server are no longer replicated and the replication server (change log and replication port) is disabled, if it is configured.

`--disableReplicationServer`. Disable the replication server. The replication port and change log are disabled on the specified server.

`-h, --hostname` *host*. Directory server host name or IP address.

`-p, --port` *port*. Directory server administration port number.

`disable-changelog`
Disables the external change log for a set of base DNs. If there is no data to replicate, then all the associated replication configuration is removed. For more information about external change log, see Section 26.5, "Using the External Change Log." Suboptions are as follows:

`-h, --hostname` *host*

Directory server host name or IP address.

`-p, --port` *port*

The Directory Server administration port number.

`-D, --bindDN` *bindDN*

The DN to bind with the server where you want to configure the external change log. The default value is `cn=Directory Manager`.

`enable-changelog`
Creates an external change log for a set of base DNs. The external change log feature allows you to retrieve the modifications performed under a specific base DN. For more information about external change log, see Section 26.5, "Using the External Change Log." Suboptions are as follows:

`-h, --hostname` *host*

Directory server host name or IP address.

`-p, --port` *port*

The Directory Server administration port number.

`-D, --bindDN` *bindDN*

The DN to bind with the server where you want to configure the external change log. The default value is `cn=Directory Manager`.

`-r, --replicationPort` *port*

The port required to configure the change log. You have to specify this option only if the changelog (or replication) is not previously configured in the server. The default value is `8989`.

`enable`
Update the configuration of the directory servers to replicate data under the specified base DN. If one of the specified servers is already replicating the data under the base DN to other servers, executing this subcommand updates the configuration of all the servers. It is therefore sufficient to execute the subcommand once for each server that is added to the replication topology. Suboptions are as follows:

`--bindDN2` *bindDN*. The DN used to bind to the second server whose contents will be replicated. If no bind DN is specified, the global administrator is used to bind.

`--bindPasswordFile1` *filename*. The file containing the password used to bind to the first server whose contents will be replicated. If no bind DN was specified for the first server, the password of the global administrator is used to bind.

`-D, --bindDN1` *bindDN*. The DN used to bind to the first server whose contents will be replicated. If no bind DN is specified, the global administrator is used to bind.

`-F, --bindPasswordFile2` *filename*. The file containing the password used to bind to the second server whose contents will be replicated. If no bind DN was specified for the second server, the password of the global administrator is used to bind.

`-h, --host1` *host*. Host name or IP address of the first server whose contents will be replicated.

`--noReplicationServer1`. Do not configure a replication port or change log on the first server. The first server will contain replicated data but will not contain a change log of modifications made to the replicated data. Note that each replicated topology must contain at least two servers with a change log to avoid a single point of failure.

`--noReplicationServer2`. Do not configure a replication port or change log on the second server. The second server will contain replicated data but will not contain a change log of modifications made to the replicated data. Note that each replicated topology must contain at least two servers with a change log to avoid a single point of failure.

`--noSchemaReplication`. Do not replicate the schema between the servers. Note that schema replication is enabled by default. Use this option if you do not want the schema to be synchronized between servers.

`--onlyReplicationServer1`. Configure only a change log and replication port on the first server. The first server will not contain replicated data, but will contain a change log of the modifications made to the replicated data on other servers.

`--onlyReplicationServer2`. Configure only a change log and replication port on the second server. The second server will not contain replicated data, but will contain a change log of the modifications made to the replicated data on other servers.

`-O, --host2` *host*. Hostname or IP address of the second server whose contents will be replicated.

`-p, --port1` *port*. Directory server administration port number of the first server whose contents will be replicated.

`--port2` *port*. Directory server administration port number of the second server whose contents will be replicated.

`-r, --replicationPort1` *port*. The port that will be used by the replication mechanism in the first directory server to communicate with other servers. Only specify this option if replication was not previously configured on the first directory server.

`-R, --replicationPort2` *port*. The port that will be used by the replication mechanism in the second directory server to communicate with other servers. Only specify this option if replication was not previously configured in the second server.

`-S, --skipPortCheck`. Skip the check to determine whether the specified replication ports are usable. If this argument is not specified, the server checks that the port is available only if you are configuring the local host.

`--secureReplication1`. Specifies whether communication through the replication port of the first server is encrypted. This option is only taken into account the first time replication is configured on the first server.

`--secureReplication2`. Specifies whether communication through the replication port of the second server is encrypted. This option is only taken into account the first time replication is configured on the second server.

`--useSecondServerAsSchemaSource`. Use the second server to initialize the schema of the first server. If neither this option nor the `--noSchemaReplication` option is specified, the schema of the first server is used to initialize the schema of the second server.

`initialize`
Initialize the contents of the data under the specified base DN on the destination directory server with the contents on the source server. This operation is required after enabling replication. Suboptions are as follows:

`-h, --hostSource` *host*. Directory server host name or IP address of the source server whose contents will be used to initialize the destination server.

`-O, --hostDestination` *host*. Directory server hostname or IP address of the destination server whose contents will be initialized.

`-p, --portSource` *port*. Directory server administration port number of the source server whose contents will be used to initialize the destination server.

`--portDestination` *port*. Directory server administration port number of the destination server whose contents will be initialized.

`initialize-all`
Initialize the data under the specified base DN, on all the directory servers in the topology, with the data on the specified server. This operation is required after enabling replication for replication to work. Alternatively, you can use the `initialize` subcommand on each individual server in the topology. Suboptions are as follows:

`-h, --hostname` *host*. Directory server host name or IP address of the source server.

`-p, --port` *port*. Directory server administration port number of the source server.

`post-external-initialization`
Enable replication to work after the entire topology has been reinitialized by using `import-ldif` or binary copy. This subcommand must be called after you initialize the contents of all directory servers in a topology by using `import-ldif` or binary copy. If you do not run this subcommand, replication will no longer work after the initialization. Suboptions are as follows:

`-h, --hostname` *host*. Directory server host name or IP address.

`-p, --port` *port*. Directory server administration port number.

`pre-external-initialization`
Prepare a replication topology for initialization by using `import-ldif` or binary copy. This subcommand must be called before you initialize the contents of all directory servers in a topology by using `import-ldif` or binary copy. If you do not run this subcommand, replication will no longer work after the initialization. After running this subcommand, initialize the contents of all the servers in the topology, then run the subcommand `post-external-initialization`. Suboptions are as follows:

`-h, --hostname` *host*. Directory server host name or IP address.

`-l, --local-only`. Use this option when the contents of only the specified directory server will be initialized with an external method.

`-p, --port` *port*. Directory server administration port number.

`purge-historical`
Launches a purge processing of the historical information stored in the user entries by replication. Since this processing may take a while, you must specify the maximum duration for this processing. Suboptions are as follows:

`-h, --hostname` *host*. Directory server host name or IP address.

`-p, --port` *port*. Directory server administration port number.

`--maximumDuration` *maximum duration*. Specifies the maximum duration the purge processing must last expressed in seconds. The default value is 3600.

`-t, --start` *startTime*. Specifies the date and time at which this operation will start when scheduled as a server task expressed in YYYYMMDDhhmmssZ format for UTC time or YYYYMMDDhhmmss for local time. Use `0` to schedule the task for immediate execution. When this option is specified the operation is scheduled to start at the specified time after which the utility exits immediately.

`--recurringTask` *schedulePattern*. Indicates the task is recurring and will be scheduled according to the value argument expressed in crontab(5) compatible time/date pattern.

`--completionNotify` *emailAddress*. Indicates the e-mail address of the recipient to be notified when the task completes. You can specify this option more than once.

`--errorNotify` *emailAddress*. Indicates the e-mail address of the recipient to be notified if an error occurs when this task executes. You can specify this option more than once.

`--dependency` *taskID*. Indicates the ID of a task upon which this task depends. A task will not start execution until all its dependent tasks have completed execution.

`--failedDependencyAction` *action*. Indicates the action that should take place if one if its dependent tasks fail. It must have one of the following values: `PROCESS`,`CANCEL`, or `DISABLE`. The default value is CANCEL.

`set-trust`
Set the trust flag of a directory server. Any change that is sent by an untrusted directory server will be discarded by the rest of the topology. Only trusted directory servers are allowed to send changes to be replayed by other directory servers. Suboptions are as follows:

`-h, --trustedHost` *host*. Specifies the fully qualified host name or IP address of the directory server that will perform the change.

`-p, --trustedPort` *port*. Specifies the administration port number of the directory server that will perform the change.

`-M, --modifiedHost` *host*. Specifies the fully qualified host name or IP address of the directory server whose trust flag is modified.

`-c, --modifiedPort` *port*. Specifies the administration port number of the directory server whose trust flag is modified.

`-t, --trustValue` *trusted*|*untrusted*. Specifies the new value of the trust flag for the directory server to be modified. The value can be `trusted` or `untrusted`. The default value is `trusted`.

`status`
List the replication configuration for the specified base DNs of all directory servers defined in the registration information. If no base DNs are specified, the information for all base DNs is displayed. Suboptions are as follows:

`-h, --hostname` *host*. Directory server host name or IP address.

`-p, --port` *port*. Directory server administration port number.

`-s, --script-friendly`. Display the status in a format that can be parsed by a script.

The `status` sub command can have the following values:

- **Normal.** The connection to a replication server is established with the right data set. Replication is working. If assured mode is used, acknowledgement signals from this directory server are sent.
- **Degraded.** The connection to a replication server is established with the right data set. Replication is working in degraded mode as the directory server has a lot of changes to be replayed pending in the replication server queue. If assured mode is used, acknowledgement signals from this directory server are not expected.
- **Full Update.** The connection to a replication server is established and a new data set is received from this connection (online import), to initialize the local back end.
- **Bad Data Set.** The connection to a replication server is established with a data set that is different from the rest of the topology. The replication is not working. Either the other directory servers of the topology should be initialized with a compatible data set, or this server should be initialized with another data set compatible with other servers'.
- **Not Connected.** The directory server is not connected to any replication server.

### A.2.6.4 Options

The `dsreplication` command accepts an option in either its short form (for example, `-H`) or its long form equivalent (for example, `--help`).

`-b, --baseDN` ***baseDN***
Specify the base DN of the data to be replicated or initialized, or for which replication should be disabled. Multiple base DNs can be specified by using this option multiple times.

### A.2.6.5 Configuration Options

`--advanced`
Use this option to access advanced settings when running this command in interactive mode.

### A.2.6.6 LDAP Connection Options

`-I, --adminUID` ***adminUID***
Specify the User ID of the global administrator to bind to the server. If no global administrator was defined previously for any of the servers, this option creates a global administrator by using the data provided.

`-j, --adminPasswordFile` ***bindPasswordFile***
Use the global administrator password in the specified file when authenticating to the directory server.

`-o, --saslOption` ***name=value***
Use the specified options for SASL authentication.

SASL is not supported for a proxy server instance.

`-X, --trustAll`
Trust any certificate that the server might present during SSL or StartTLS negotiation. This option can be used for convenience and testing purposes, but for security reasons a trust store should be used to determine whether the client should accept the server certificate.

`-P, --trustStorePath` ***trustStorePath***
Use the client trust store certificate in the specified path. This option is not needed if `--trustAll` is used, although a trust store should be used when working in a production environment.

`-U, --TrustStorePasswordFile` ***path***
Use the password in the specified file to access the certificates in the client trust store. This option is only required if `--trustStorePath` is used and the specified trust store requires a password in order to access its contents (most trust stores do not require this).

`-K, --keyStorePath` ***keyStorePath***
Use the client keystore certificate in the specified path.

`-u, --keyStorePasswordFile` ***keyStorePasswordFile***
Use the password in the specified file to access the certificates in the client keystore. This option is only required if `--keyStorePath` is used.

`-N, --certNickname` ***nickname***
Use the specified certificate for authentication.

`--connectTimeout` ***timeout***
Specifies the maximum length of time (in milliseconds) that can be taken to establish a connection. Use `0`to specify no time out. The default value is 30000.

### A.2.6.7 Command Input/Output Options

`--commandFilePath` ***path***
Specify the full path to the file in which the equivalent non-interactive commands are written when the command is run in interactive mode.

`--displayCommand`
Display the equivalent non-interactive command in the standard output when the command is run in interactive mode.

`-n, --no-prompt`
Run in non-interactive mode. If some data in the command is missing, the user will not be prompted and the command will fail.

`--noPropertiesFile`
Indicate that the command will not use a properties file to get the default command-line options.

`--propertiesFilePath` ***propertiesFilePath***
Specify the path to the properties file that contains the default command-line options.

`-Q, --quiet`
Run in quiet mode. No output will be generated unless a significant error occurs during the process.

### A.2.6.8 General Options

`-?, -H, --help`
Display command-line usage information for the command and exit without making any attempt to stop or restart the server.

`-V, --version`
Display the version information for the server and exit rather than attempting to run this command.

### A.2.6.9 Examples

The following examples assume that two directory servers are installed: `host1` and `host2`. Both servers are configured with the default administration port (4444). The base DN `dc=example,dc=com` is populated with data on `host1`. The base DN exists on `host2`, but is empty. The examples configure replication between the two servers and initialize `host2` with data.

> **Note:** The easiest way to use `dsreplication` is in interactive mode, in which case you are prompted for all of the relevant arguments. However, to illustrate which arguments are configured, these examples do not use the interactive mode.

#### *Example A–19 Enabling Directory Server Replication*

The following command enables replication for the base DN `dc=example,dc=com` on `host1` and `host2`. The command runs in non-interactive mode (`-n`) and specifies that all server certificates should be accepted (`-X`).

```
$ dsreplication enable \
  --host1 host1 --port1 4444 --bindDN1 "cn=Directory Manager" \
  --bindPasswordFile1 /tmp/pwd-file --replicationPort1 8989 \
  --host2 host2 --port2 4444 --bindDN2 "cn=Directory Manager" \
  --bindPasswordFile2 /tmp/pwd-file --replicationPort2 8989 \
  --adminUID admin --adminPasswordFile /tmp/pwd-file --baseDN "dc=example,dc=com"
-X -n
```

#### *Example A–20 Initializing Directory Server Replication*

To initialize one replica from another, use the `initialize` subcommand. The following command initializes the base DN `dc=example,dc=com` on `host2` with the data contained on `host1`. The command runs in non-interactive mode (`-n`) and specifies that all server certificates should be accepted (`-X`).

```
$ dsreplication initialize --baseDN "dc=example,dc=com" \
  --adminUID admin --adminPasswordFile /tmp/pwd-file \
  --hostSource host1 --portSource 4444 \
  --hostDestination host2 --portDestination 4444 -X -n
```

To initialize an entire topology, use the `initialize-all` subcommand. This subcommand takes the details of the source directory server as options and initializes all other replicas for which replication has been enabled.

***Example A–21 Obtaining the Directory Server Replication Status***

The following command obtains the replication status of the directory servers in the topology.

```
$ dsreplication status --adminUID admin --adminPasswordFile /tmp/pwd-file -X --hostname host1 --port 4444

dc=example,dc=com - Replication Enabled
=======================================
Server     : Entries:M.C.[1]:A.O.M.C.[2]:Port[3]:Encryption[4]:Trust[5]:U.C.[6]:Status[7]:ChangeLog[8]:Group
ID[9]:Connected to[10]
-----------:--------:-------:-----------:-------:-------------:--------:-------:---------:------------:---------
--:-------------------------
host1:4444 : 200000 : 0     : N/A       : 1898  : Disabled    :Trusted : N/A   : Normal  : Enabled    : 1
:host1/203.0.113.24:1898
host2:5444 : 200000 : 0     : N/A       : 2898  : Disabled    :Trusted : N/A   : Normal  : Enabled    : 1
:host2/203.0.113.24:2898

[1] The number of changes that are still missing on this server (and that have been applied to at least one
    other server).
[2] Age of oldest missing change: the age (in seconds) of the oldest change that has not yet arrived on this
    server.
[3] The port used to communicate between the servers whose contents are being replicated.
[4] Whether the replication communication through the replication port is encrypted or not.
[5] Whether this directory server is trusted or not. Updates coming from an untrusted server are discarded and
    not propagated.
[6] The number of untrusted changes. These are changes generated on this server while it is untrusted.
    Those changes are not propagated to the rest of the topology but are effective on the untrusted server.
[7] The status of the replication domain on this directory server.
[8] Whether the external change log is enabled or not for the base DN on this server.
[9] The ID of the replication group to which the server belongs.
[10] The replication this server is connected to.
```

***Example A–22 Disabling Directory Server Replication***

The following command disables replication for the base DN `dc=example,dc=com` on `host2`. Disabling replication on one directory server removes all references to that server from the other directory servers in the replication topology.

```
$ dsreplication disable --baseDN "dc=example,dc=com" \
--hostname host2 --port 4444 --adminUID admin --adminPasswordFile /tmp/pwd-file \
-X -n
Establishing connections ..... Done.
Disabling replication on base DN cn=admin data of server host2:4444 ..... Done.
Disabling replication on base DN dc=example,dc=com of server host2:4444 .....
Done.
Disabling replication on base DN cn=schema of server host2:4444 ..... Done.
Removing references on base DN cn=admin data of server host1:4444 ..... Done.
Removing references on base DN dc=example,dc=com of server host1:4444 ..... Done.
Removing references on base DN cn=schema of server host1:4444 ..... Done.
Disabling replication port 8990 of server host2:4444 ..... Done.
```

***Example A–23 Configuring the External Change Log on a Non-replicated Server***

The following command enables the external change log on a non-replicated sever.

```
$ dsreplication status -h localhost -p 4444 -D "cn=directory manager" --adminPasswordFile /tmp/pwd-file -X -n

dc=example,dc=com - Replication Disabled
========================================
Server         : Entries : ChangeLog [8]
---------------:---------:--------------
localhost:4444 : 0       : Disabled
```

```
dsreplication enable-changelog -h localhost -p 4444 -D "cn=directory manager" --adminPasswordFile
/tmp/pwd-file -X -b dc=example,dc=com -n

Establishing connections ..... Done.
Enabling Changelog on base DN 'dc=example,dc=com' ..... Done.
Configuring Replication port on server localhost:4444 ..... Done.

dsreplication status -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -x -n

dc=example,dc=com - Replication Enabled
======================================
Server     : Entries:M.C.[1]:A.O.M.C.[2]:Port[3]:Encryption[4]:Trust[5]:U.C.[6]:Status[7]:ChangeLog[8]:Group
ID[9]:Connected to[10]
-----------:--------:-------:-----------:-------:-------------:--------:-------:---------:------------:---------
-:-------------------------
host1:4444 : 200000 : N/A   : N/A       : 1898  : Disabled    :Trusted : N/A   : Normal  : Enabled    : 1
: host1/203.0.113.24:1898

[1] The number of changes that are still missing on this server (and that have been applied to at least one
    other server).
[2] Age of oldest missing change: the age (in seconds) of the oldest change that has not yet arrived on this
    server.
[3] The port used to communicate between the servers whose contents are being replicated.
[4] Whether the replication communication through the replication port is encrypted or not.
[5] Whether this directory server is trusted or not. Updates coming from an untrusted server are discarded and
    not propagated.
[6] The number of untrusted changes. These are changes generated on this server while it is untrusted.
    Those changes are not propagated to the rest of the topology but are effective on the untrusted server.
[7] The status of the replication domain on this directory server.
[8] Whether the external change log is enabled or not for the base DN on this server.
[9] The ID of the replication group to which the server belongs.
[10] The replication this server is connected to.
```

#### A.2.6.10 Exit Codes

0
Successful.

1
Unable to initialize arguments.

2
Cannot parse arguments because the provided arguments are not valid or there was an error checking the user data.

3
The user canceled the operation in interactive mode.

4
Conflicting arguments.

5
The specified base DNs cannot be used to enable replication.

6
The specified base DNs cannot be used to disable replication.

7
The specified base DNs cannot be used to initialize the contents of the replicas.

8
Error connecting with the credentials provided.

9
Could not find the replication ID of the domain to be used to initialize the replica.

10
The maximum number of attempts to start the initialization has been exceeded. A systematic "peer not found error" was received.

11
Error enabling replication on base DN.

12
Error initializing base DN.

13
Error reading configuration.

14
Error updating ADS.

15
Error reading ADS.

16
Error reading Topology Cache.

17
Error configuring the replication server.

18
Unsupported ADS scenario.

19
Error disabling replication on base DN.

20
Error removing replication port reference on base DN.

21
Error initializing Administration Framework.

22
Error seeding trust store.

23
Error launching pre-external initialization.

24
Error launching post-external initialization.

25
Error disabling replication server.

26
Error executing purge historical.

`27`
The specified base DN cannot be purged.

`28`
Error launching purge historical.

`29`
Error loading configuration class in local purge historical.

`30`
Error starting server in local purge historical.

`31`
Timeout error in local purge historical.

`32`
Generic error executing local purge historical.

`33`
The trusted host was not found in the ADS.

`34`
The modified host was not found in the ADS.

`35`
The changelog cannot be enabled on this base DN.

`36`
The changelog cannot be disabled on this base DN.

`37`
An error occurred configuring the changelog.

`38`
The specified host was not found in the configuration.

#### A.2.6.11 Using a Properties File

The directory server supports the use of a *properties file* that passes in any default option values used with the `dsreplication` command. The properties file is convenient when working in different configuration environments, especially in scripted or embedded applications. For more information, see Section A.1.2, "Using a Properties File With Server Commands."

The following options can be stored in a properties file:

- `adminUID`
- `baseDN`
- `certNickname`
- `keyStorePasswordFile`
- `keyStorePath`
- `saslOption`

  SASL is not supported for a proxy server instance.
- `trustAll`

- `trustStorePasswordFile`
- `trustStorePath`

Entries in the properties file have the following format:

```
toolname.propertyname=propertyvalue
```

For example:

```
dsreplication.baseDN=dc=example,dc=com
```

#### A.2.6.12 Location

- UNIX and Linux: *INSTANCE_DIR*`/OUD/bin/dsreplication`
- Windows: *INSTANCE_DIR*`\OUD\bat\dsreplication.bat`

#### A.2.6.13 Related Commands

Section A.2.4, "dsconfig"

### A.2.7 gicadm

The `gicadm` command manages global indexes and global index catalogs.

This command is supported only for the proxy.

#### A.2.7.1 Synopsis

`gicadm` [*subcommand*] [*options*]

#### A.2.7.2 Description

The `gicadm` command enables you to create and delete a global index catalog, as well as add, modify, and delete global indexes in a global index catalog, and manage replication of global index catalogs. It also allows you to associate a global index to a distribution.

The `gicadm` command accesses the server over SSL through the administration connector.

#### A.2.7.3 Options

The `gicadm` command accepts the following options.

`add-index`
Adds a new global index to a global index catalog. Suboptions are as follows:

`-c, --catalogName` *name*. A unique identifier for the global index catalog. This is a required argument.

`--attributeName attribute-name`. The identifier for the global index attribute. This identifier should be unique in the context of the global index catalog and it is used to identify the global index.

`--set` *property*`:`*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it.

`associate`
Associates a global index catalog to a distribution workflow element. Suboptions are as follows:

`-c, --catalogName` *name*. A unique identifier for the global index catalog. This is a required argument.

`-d, --distributionWorkflowElement` *distribution-workflow-element*. The name of the distribution workflow element object using this global index catalog, from which the global index catalog is to be disassociated.

`create-catalog`
Creates a new global index catalog. Suboptions are as follows:

`-c, --catalogName` *name*. A unique identifier for the global index catalog. This is a required argument.

`delete-catalog`
Deletes a global index catalog. Suboptions are as follows:

`-c, --catalogName` *name*. A unique identifier for the global index catalog. This is a required argument.

`disable-replication`
Disables replication on the specified server for the specified global index catalog and removes any references to this server from the other servers in the replication topology. Suboptions are as follows:

`-c, --catalogName` *name*. A unique identifier for the global index catalog. This is a required argument.

`--adminUID` *adminUID*. User ID of the global administrator used to bind to the server. For the `enable-replication` subcommand if no global administrator was defined previously the global administrator will be created using the provided data.

`disassociate`
Disassociates a global index catalog from a distribution workflow element. Suboptions are as follows:

`-d, --distributionWorkflowElement` *distribution-workflow-element*. The name of the distribution workflow element object using this global index catalog, from which the global index catalog is to be disassociated.

`enable-replication`
Updates the server configuration to replicate the global index catalog and all its global indexes. If one of the specified servers already replicates the global index catalog for a given global index, executing this subcommand will update the configuration of all servers in the topology. Therefore, it is sufficient to execute this command once for each server added to the replication topology. Suboptions are as follows:

`-c, --catalogName` *name*. A unique identifier for the global index catalog. This is a required argument.

`--adminUID` *adminUID*. User ID of the global administrator used to bind to the server. For the `enable-replication` subcommand, if no global administrator was defined previously, the global administrator will be created using the provided data.

`--adminPasswordFile` *bindPasswordFile*. The file containing the password of the global administrator.

`--localReplicationPort` *port*. Replication port number of the first server whose content will be replicated.

`--localSecureReplication`. Specifies whether or not the communication through the replication port of the first server is encrypted or not. This option will only be taken into account the first time replication is configured on the first server.

`--remoteAdminPort` *port*. Directory server administration port number of the second server whose contents will be replicated.

`--remoteHost` *host*. Fully qualified directory server host name or IP address of the second server whose contents will be replicated.

`--remoteBindDN` *bindDN*. DN to use to bind to the second server whose content will be replicated. If not specified the global administrator will be used to bind.

`--remoteBindPasswordFile` *bindPasswordFile*. File containing the password to use to bind to the second server whose content will be replicated. If no bind DN was specified for the second server the password of the global administrator will be used to bind.

`--remoteReplicationPort` *port*. Replication port number of the second server whose content will be replicated.

`--remoteSecureReplication`. Specifies whether or not the communication through the replication port of the second server is encrypted or not. This option will only be taken into account the first time.

`export`
Exports a global index catalog to file. Suboptions are as follows:

`-c, --catalogName` *name*. A unique identifier for the global index catalog. This is a required argument.

`--exportDirectory` *directory*. Path to the directory to be used to export the global index catalog. This is a required argument.

`-a, --attributeName attribute-name`. The name of the global index attribute. This option can be used multiple times to specify multiple indexed attributes. If this option is provided, any indexed attribute in the import source that does not match is skipped.

`get-catalog-prop`
Shows global index catalog properties. Suboptions are as follows:

`-c, --catalogName` *name*. A unique identifier for the global index catalog. This is a required argument.

`--property` *property*. The name of a property to be displayed.

`-E,--record`. Modifies the display output to show one property value per line.

`get-index-prop`
Shows index properties. Suboptions are as follows:

`-c, --catalogName` *name*. A unique identifier for the global index catalog. This is a required argument.

`-a, --attributeName` *attribute-name*. The identifier for the global index attribute. This identifier should be unique in the context of the global index catalog and it is used to identify the global index.

`--property` *property*. The name of a property to be displayed. Valid property names are:`all`, `global-index-deleted-entry-retention-timeout`, `db-cleaner-min-utilization`, `db-log-file-max`, `db-checkpointer-bytes-interval`, `db-checkpointer-wakeup-interval`,

db-num-lock-tables, db-num-cleaner-threads, db-txn-no-sync, db-txn-write-no-sync, je-property, db-directory, db-directory-permissions, global-index-catalogs-shared-cache, and global-index-attribute.

import
Imports content of a file into a specified global index catalog. Suboptions are as follows:

-c, --catalogName *name*. A unique identifier for the global index catalog. This is a required argument.

--importDirectory *directory*. Path to the file to be used to import the global index catalog. This is a required argument.

--attributeName *attribute-name*. The identifier for the global index attribute. This identifier should be unique in the context of the global index catalog and it is used to identify the global index.

--append. Append to an existing global index rather than overwriting it.

initialize-replication
Initializes the replication of a global index catalog. All the replicated global index catalogs (part of the replication topology) can be initialized at once or the local global index catalog is initialized from a given global index catalog (also part of the replication topology). Suboptions are as follows:

-c, --catalogName *name*. A unique identifier for the global index catalog. This is a required argument.

--adminUID *adminUID*. User ID of the global administrator used to bind to the server. For the initialize-replication subcommand, if no global administrator was defined previously, the global administrator will be created using the provided data.

--fromServerPort *port*. Directory server port number of the source server whose contents will be used to initialize the destination server.

--fromServerHost *host*. Directory server hostname or IP address of the source server whose contents will be used to initialize the destination server.

--all. Initializes the contents of the global index attribute on all the servers whose contents is being replicated with the contents on the specified server.

list-catalogs
Lists the global index catalogs that have been defined. Suboptions are as follows:

--property *property*. The name of a property to be displayed. Valid property names are:all, replication-server, server-id, window-size, heartbeat-interval and group-id.

list-indexes
Lists the global indexes that have been defined in the global index catalog. Suboptions are as follows:

-c, --catalogName *name*. A unique identifier for the global index catalog. This is a required argument.

--property *property*. The name of a property to be displayed. Valid property names are:all, global-index-deleted-entry-retention-timeout, db-cleaner-min-utilization, db-log-file-max, db-checkpointer-bytes-interval, db-checkpointer-wakeup-interval,

db-num-lock-tables, db-num-cleaner-threads, db-txn-no-sync, db-txn-write-no-sync, je-property, db-directory, db-directory-permissions, global-index-catalogs-shared-cache, and global-index-attribute.

post-external-initialization
This subcommand must be called after initializing the contents of all the replicated global indexes using the import subcommand of this tool. It will use the generation id of the targeted instance as the valid one. Suboptions are as follows:

-c, --catalogName *name*. A unique identifier for the global index catalog. This is a required argument.

-a, --attributeName *attribute-name*. The identifier for the global index attribute. This option can be used multiple times to specify multiple indexed attributes. If this option is provided, any indexed attribute in the import source that does not match is skipped.

pre-external-initialization
This subcommand can be called before initializing the contents of all the replicated servers using the import subcommand of this tool. It will erase the replication change logs stored in the replication servers. Suboptions are as follows:

-c, --catalogName *name*. A unique identifier for the global index catalog. This is a required argument.

-a, --attributeName *attribute-name*. The identifier for the global index attribute. This option can be used multiple times to specify multiple indexed attributes. If this option is provided, any indexed attribute in the import source that does not match is skipped.

remove-index
Removes a global index from a global index catalog. Suboptions are as follows:

-c, --catalogName *name*. A unique identifier for the global index catalog. This is a required argument.

--attributeName *attribute-name*. The identifier for the global index attribute. This identifier should be unique in the context of the global index catalog and it is used to identify the global index.

set-catalog-prop
Modifies the properties of the global index catalog. Suboptions are as follows:

-c, --catalogName *name*. A unique identifier for the global index catalog. This is a required argument.

--set *property*:*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it. Valid property names are: all, global-index-deleted-entry-retention-timeout, db-cleaner-min-utilization, db-log-file-max, db-checkpointer-bytes-interval, db-checkpointer-wakeup-interval, db-num-lock-tables, db-num-cleaner-threads, db-txn-no-sync, db-txn-write-no-sync, je-property, db-directory, db-directory-permissions, global-index-catalogs-shared-cache, and global-index-attribute.

--reset *property*. Resets a property back to its default values, where *property* is the name of the property to be reset. Valid property names are: all,

global-index-deleted-entry-retention-timeout, db-cleaner-min-utilization, db-log-file-max, db-checkpointer-bytes-interval, db-checkpointer-wakeup-interval, db-num-lock-tables, db-num-cleaner-threads, db-txn-no-sync, db-txn-write-no-sync, je-property, db-directory, db-directory-permissions, global-index-catalogs-shared-cache, and global-index-attribute.

--add *property*:*value*. Adds a single value to a property, where *property* is the name of the property and *value* is the single value to be added.

--remove *property*:*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed. Valid property names are: all, global-index-deleted-entry-retention-timeout, db-cleaner-min-utilization, db-log-file-max, db-checkpointer-bytes-interval, db-checkpointer-wakeup-interval, db-num-lock-tables, db-num-cleaner-threads, db-txn-no-sync, db-txn-write-no-sync, je-property, db-directory, db-directory-permissions, global-index-catalogs-shared-cache, and global-index-attribute.

set-index-prop
Modifies the properties of an index. Suboptions are as follows:

-c, --catalogName *name*. A unique identifier for the global index catalog. This is a required argument.

--attributeName attribute-name. The identifier for the global index attribute. This identifier should be unique in the context of the global index catalog and it is used to identify the global index.

--set *property*:*value*. Assigns a value to a property, where *property* is the name of the property and *value* is the single value to be assigned. Specify the same property multiple times to assign more than one value to it. Valid property names are: all, global-index-deleted-entry-retention-timeout, db-cleaner-min-utilization, db-log-file-max, db-checkpointer-bytes-interval, db-checkpointer-wakeup-interval, db-num-lock-tables, db-num-cleaner-threads, db-txn-no-sync, db-txn-write-no-sync, je-property, db-directory, db-directory-permissions, global-index-catalogs-shared-cache, and global-index-attribute.

--reset *property*. Resets a property back to its default values, where *property* is the name of the property to be reset. Valid property names are: all, global-index-deleted-entry-retention-timeout, db-cleaner-min-utilization, db-log-file-max, db-checkpointer-bytes-interval, db-checkpointer-wakeup-interval, db-num-lock-tables, db-num-cleaner-threads, db-txn-no-sync, db-txn-write-no-sync, je-property, db-directory, db-directory-permissions, global-index-catalogs-shared-cache, and global-index-attribute.

--remove *property*:*value*. Removes a single value from a property, where *property* is the name of the property and *value* is the single value to be removed. Valid property names are: all, global-index-deleted-entry-retention-timeout, db-cleaner-min-utilization, db-log-file-max, db-checkpointer-bytes-interval, db-checkpointer-wakeup-interval, db-num-lock-tables, db-num-cleaner-threads, db-txn-no-sync, db-txn-write-no-sync, je-property, db-directory,

`db-directory-permissions`, `global-index-catalogs-shared-cache`, and `global-index-attribute`.

`status-replication`
Displays a list with the basic replication configuration of the global index catalog. If no global index catalog is specified, the information for all replicated global index catalogs is displayed. Suboptions are as follows:

`-c, --catalogName` *name*. A unique identifier for the global index catalog. This is a required argument.

`--adminUID` *adminUID*. User ID of the global administrator used to bind to the server. For the `status-replication` subcommand, if no global administrator was defined previously, the global administrator will be created using the provided data.

`-s, --scriptFriendly`. Use the script-friendly mode.

### A.2.7.4 LDAP Connection Options

The `gicadm` command contacts the directory server over SSL through the administration connector (described in Section 14.3, "Managing Administration Traffic to the Server"). These connection options are used to contact the directory server.

`-h, --hostname` ***host***
Directory server hostname or IP address.

`-D, --bindDN` ***bindDN***
DN to use to bind to the server.

`-j, --bindPasswordFile` ***filename***
The full path to the file containing the bind password.

`-K, --keyStorePath` ***path***
Use the client keystore certificate in the specified path.

`-N, --certNickname` ***nickname***
Use the certificate for SSL client authentication.

`-o, --saslOption`***name=value***
SASL bind option.

`-p, --port` ***port***
Directory server administration port number.

`-P, --trustStorePath` ***path***
Use the client trust store certificate in the specified path. This option is not needed if `--trustAll` is used, although a trust store should be used when working in a production environment.

`-u, --keyStorePasswordFile` ***filename***
Use the password in the specified file to access the certificates in the client keystore. This option is only required if `--keyStorePath` is used.

`-U, --trustStorePasswordFile` ***filename***
Use the password in the specified file to access the certificates in the client trust store. This option is only required if `--trustStorePath` is used and the specified trust store requires a password to access its contents (most trust stores do not require this).

`-X, --trustAll`
Trust any certificate that the server presents. This option can be used for testing purposes, but for security reasons, a trust store should be used to determine whether the client should accept the server certificate.

`--connectTimeout` ***timeout***
Specifies the maximum duration of time (in milliseconds) that can be taken to establish a connection. Use `0`to indicate no time out. The default value is 30000 milliseconds.

### A.2.7.5 Command Input/Output Options

`--noPropertiesFile`
Indicate that the command will not use a properties file to get the default command-line options.

`--propertiesFilePath` ***propertiesFilePath***
Specify the path to the properties file that contains the default command-line options.

`-v, --verbose`
Run in verbose mode, displaying diagnostics on standard output.

### A.2.7.6 General Options

`-?, -H, --help`
Displays command-line usage information for the command and exit without making any attempt to stop or restart the directory server.

`-V, --version`
Displays the version information for the directory server.

### A.2.7.7 Examples

The following examples show how to use the `gicadm` command.

> **Note:** The following examples for creating a global index catalog, adding a global index, and associating a global index catalog to a distribution are the three steps required to use a global index catalog in a distribution deployment.

***Example A–24 Viewing the Global Help Subcommands and Global Options***

The following command displays the available global Help subcommands and global options for managing the global index catalog:

```
$ gicadm --help
```

***Example A–25 Viewing Help on an Individual Subcommand***

The following command displays the help information for the `create-catalog` subcommand:

```
$ gicadm create-catalog --help
```

***Example A–26 Using `gicadm` to Create a Global Index Catalog***

You must have deployed the proxy with distribution before running this command.

```
$ gicadm -h localhost -p 4444 -D "cn=Directory Manager" -j /path/pwd-file -X \
create-catalog --catalogName myCatalog
```

***Example A–27 Using `gicadm` to Add a Global Index to a Global Index Catalog***

You must have deployed the proxy with distribution before running this command. Moreover, you must already have created the global index catalog before running this command.

```
$ gicadm -h localhost -p 4444 -D "cn=Directory Manager" -j /tmp-pwd-file -X \
add-index --catalogName myCatalog --attributeName telephoneNumber
```

***Example A–28 Using `gicadm` to Associate a Global Index Catalog to a Distribution***

You must have deployed the proxy with distribution before running this command. Moreover, you must already have created the global index catalog before running this command.

```
$ gicadm -h localhost -p 4444 -D "cn=Directory Manager" -j /tmp-pwd-file -X \
associate --catalogName myCatalog --distributionWorkflowElement myDistributionName
```

#### A.2.7.8 Exit Codes

An exit code of 0 indicates that the operation completed successfully. A nonzero exit code indicates that an error occurred during processing.

#### A.2.7.9 Location

- UNIX and Linux: *INSTANCE_DIR*`/OUD/bin/gicadm`
- Windows: *INSTANCE_DIR*`\OUD\bat\gicadm.bat`

#### A.2.7.10 Related Commands

- `dsconfig`
- `split-ldif`

### A.2.8 manage-tasks

The `manage-tasks` command manages and monitors tasks that have been scheduled to run on the directory server.

This command is not supported for the proxy.

#### A.2.8.1 Synopsis

`manage-tasks` [*options*]

#### A.2.8.2 Description

The `manage-tasks` command can be used to manage and monitor tasks that have been scheduled to run on the directory server. Tasks are scheduled by providing the appropriate scheduling information when the task is invoked (see Section 14.4, "Configuring Commands As Tasks"). The `manage-tasks` command can be used to list tasks that are currently scheduled or that have already been executed. In addition, you can get more detailed information about a task's scheduled and execution time, its log messages, and its options.

The `manage-tasks` command can only be run on an online server instance, and accesses the task back end over SSL through the administration connector (described in Section 14.3, "Managing Administration Traffic to the Server".)

### A.2.8.3 Options

The `manage-tasks` command accepts an option in either its short form (for example, `-c` *taskID*) or its long form equivalent (for example, `--cancel` *taskID*).

`-c, --cancel` ***taskID***
Specify a particular task to cancel.

`-i, --info` ***taskID***
Display information for a particular task.

`-s, --summary`
Print a summary of tasks.

### A.2.8.4 LDAP Connection Options

`-D, --bindDN` ***bindDN***
Use the bind DN to authenticate to the directory server. This option is used when performing simple authentication and is not required if SASL authentication is used. The default value for this option is `cn=Directory Manager`.

`-h, --hostname` ***hostname***
Contact the directory server on the specified hostname or IP address. If this option is not provided, a default of `localhost` is used.

`-j, --bindPasswordFile` ***filename***
Use the bind password in the specified file when authenticating to the directory server.

`-K, --keyStorePath` ***path***
Use the client keystore certificate in the specified path.

`-N, --certNickname` ***nickname***
Use the specified certificate for client authentication.

`-o, --saslOption` ***name*=*value***
Use the specified options for SASL authentication.

`-p, --port` ***port***
Contact the directory server at the specified administration port. If this option is not provided, a default administration port of `4444` is used.

`-P, --trustStorePath` ***path***
Use the client trust store certificate in the specified path. This option is not needed if `--trustAll` is used, although a trust store should be used when working in a production environment.

`-u, --keyStorePasswordFile` ***filename***
Use the password in the specified file to access the certificates in the client keystore. This option is only required if `--keyStorePath` is used.

`-U, --trustStorePasswordFile` ***filename***
Use the password in the specified file to access the certificates in the client trust store. This option is only required if `--trustStorePath` is used and the specified trust store requires a password in order to access its contents (most trust stores do not require this).

`-X, --trustAll`
Trust all server SSL certificates that the directory server presents. This option can be used for convenience and testing purposes, but for security reasons a trust store should be used to determine whether the client should accept the server certificate.

### A.2.8.5 Command Input/Output Options

`-n,--no-prompt`
Use non-interactive mode. If required option values are missing, you are not prompted and the command will fail.

`--noPropertiesFile`
Indicates that a properties file is not used to obtain the default command-line options.

`--propertiesFilePath` ***path***
Specify the path to the properties file that contains the default command-line options.

### A.2.8.6 General Options

`-?, -H, --help`
Display command-line usage information for the command and exit without making any attempt to manage tasks.

`-V, --version`
Display the version information for the directory server and exit rather than attempting to run this command.

### A.2.8.7 Examples

The following examples show how to use the `manage-tasks` command.

***Example A–29 Displaying a Summary of Scheduled Tasks***

The following command displays a list of scheduled tasks:

```
$ manage-tasks -h localhost -p 4444 -D "cn=directory manager" -j /path/pwd-file \
  -X -s

  ID               Type     Status
  ------------------------------------------------
  2008101610361710  Backup   Completed successfully
  2008101610403710  Restore  Completed successfully
  2008101610442610  Restore  Waiting on start time
```

***Example A–30 Obtaining Task Information***

The following command returns information about a specific task:

```
$ manage-tasks -h localhost -p 4444 -D "cn=directory manager" -j /path/pwd-file \
  -X -i 2008101610442610
  Task Details
  ------------------------------------------------------
  ID                       2008101610442610
```

```
  Type                      Restore
  Status                    Waiting on start time
  Scheduled Start Time      Jan 25, 2009 12:15:00 PM SAST
  Actual Start Time
  Completion Time
  Dependencies              None
  Failed Dependency Action  None
  Email Upon Completion     admin@example.com
  Email Upon Error          admin@example.com

  Restore Options
  ---------------------------------
  Backup Directory  /backup/userRoot
```

***Example A–31 Canceling a Scheduled Task***

The following command cancels a scheduled task. The command uses the `--no-prompt` option to run in non-interactive mode.

```
$ manage-tasks -h localhost -p 4444 -D "cn=directory manager" -j /path/pwd-file \
  -X -c 2008101610442610
Task 2008101610442610 canceled
```

#### A.2.8.8 Exit Codes

An exit code of 0 indicates that the operation completed successfully. An exit code of 1 indicates that an error occurred during processing.

#### A.2.8.9 Using a Properties File

The directory server supports the use of a *properties file* that passes in any default option values used with the `manage-tasks` command. The properties file is convenient when working in different configuration environments, especially in scripted or embedded applications. For more information, see Section A.1.2, "Using a Properties File With Server Commands."

#### A.2.8.10 Location

- UNIX and Linux: *OUD_ORACLE_HOME*`/bin/manage-tasks`
- Windows: *OUD_ORACLE_HOME*`\bat\manage-tasks.bat`

#### A.2.8.11 Related Commands

- Section A.3.6, "import-ldif"
- Section A.3.5, "export-ldif"
- Section A.3.1, "backup"
- Section A.3.14, "restore"
- Section A.2.17, "stop-ds"

### A.2.9 oudCopyConfig

The `oudCopyConfig` command is used to obtain a copy of an existing configuration, from the source environment.

For more information about moving from a test to production environment, see Chapter 28, "Moving From a Test to a Production Environment."

### A.2.9.1 Synopsis

`oudCopyConfig` [*options*]

### A.2.9.2 Description

To obtain a copy of an existing configuration, run the `oudCopyConfig` command in the source environment.

The `oudCopyConfig` command performs the following actions:

- It creates an archive (`archivePath`) that contains the required configuration data to move the test instance (`instHomePath`) to a production environment. The `-archiveLoc` option specifies the full path to the archive.
- It creates a move plan in the archive.
- Logs any messages to *log_directory.* If not specified, the default location of logged messages is the system temporary directory.

### A.2.9.3 Options

The `oudCopyConfig` command accepts an option in the form :

`-javaHome,` ***javaHomePath***
Absolute path of JDK.

`-al, -archiveLoc` ***archivePath***
Absolute path of archive location. It contains the required configuration data to move the test instance (`instHomePath`) to a production environment.

`-sih, -sourceInstanceHomeLoc` ***instHomePath***
Absolute path of an existing instance that you want to copy to a production environment.

`-h, -help`
Show this help message and exit. This parameter is **optional.**

`-ldl, -logDirLoc` ***logPath***
Existing log directory location. Default location is system temporary location. This parameter is **optional.**

### A.2.9.4 Examples

The following examples show how to use the `oudCopyConfig` command.

***Example A–32 Obtaining a Copy of an Existing Configuration***

The following command obtains a copy of an existing configuration.

```
$ OUD_ORACLE_HOME/bin/oudCopyConfig -javaHome /usr/jdk \
-sourceInstanceHomeLoc /local/asinst_1 -archiveLoc /tmp/oud.jar \
-logDirLoc /tmp/logs
```

***Example A–33 Running the Help Command Option***

```
$ OUD_ORACLE_HOME/bin/oudCopyConfig -javaHome /usr/jdk -help
```

#### A.2.9.5 Location

- UNIX and Linux: *OUD_ORACLE_HOME*`/bin/oudCopyConfig`
- Windows: *OUD_ORACLE_HOME*`\bat\oudCopyConfig.bat`

#### A.2.9.6 Related Commands

- Section A.2.10, "oudExtractMovePlan"
- Section A.2.11, "oudPasteConfig"

### A.2.10 oudExtractMovePlan

The `oudExtractMovePlan` command is used to create an editable version of the configuration in a file named *moveplan.xml*, in the location specifed by the `-planDirLoc` argument. This directory must exist, and be writable.

For more information about moving from a test to production environment, see Chapter 28, "Moving From a Test to a Production Environment."

#### A.2.10.1 Synopsis

`oudExtractMovePlan` [*options*]

#### A.2.10.2 Description

You can modify certain configuration parameters by editing the move plan. A move plan is an XML file that exposes customizable parameters during the move across environments.

The move plan is generated when you run the `oudCopyConfig` command and is used by the `oudPasteConfig` command to duplicate the configuration.

#### A.2.10.3 Options

The `oudExtractMovePlan` command accepts an option in the form:

`-javaHome,` ***javaHomePath***
Absolute path of JDK.

`-al, -archiveLoc` ***archivePath***
Absolute path of archive location.

`-pdl, -planDirLoc` ***planPath***
Absolute path to directory where moveplan is to be extracted. The name of move plan file is *moveplan.xml*.

`-h, -help`
Show this help message and exit. This parameter is **optional.**

`-ldl, -logDirLoc` ***logPath***
Existing log directory location. Default location is system temporary location. This parameter is **optional.**

#### A.2.10.4 Examples

The following examples show how to use the `oudExtractMovePlan` command.

***Example A–34 Editing the Configuration***

The following command allows you to edit the configuration.

```
$ OUD_ORACLE_HOME/bin/oudExtractMovePlan -javaHome /usr/jdk \
-al /tmp/oud.jar -pdl /tmp -logDirLoc /tmp/logs
```

***Example A–35 Running the Help Command Option***

```
$ OUD_ORACLE_HOME/bin/oudExtractMovePlan -javaHome /usr/jdk -help
```

#### A.2.10.5 Location

- UNIX and Linux: *OUD_ORACLE_HOME*`/bin/oudExtractMovePlan`
- Windows: *OUD_ORACLE_HOME*`\bat\oudExtractMovePlan.bat`

#### A.2.10.6 Related Commands

- Section A.2.9, "oudCopyConfig"
- Section A.2.11, "oudPasteConfig"

### A.2.11 oudPasteConfig

The `oudPasteConfig` command is used to paste the configuration in the target environment.

For more information about moving from a test to production environment, see Chapter 28, "Moving From a Test to a Production Environment."

#### A.2.11.1 Synopsis

`oudPasteConfig` [*options*]

#### A.2.11.2 Description

To obtain the configuration in the target environment, run the `oudPasteConfig` command.

The `oudPasteConfig` command creates a new server instance with the configuration obtained from the archive and the amended move plan.

#### A.2.11.3 Options

The `oudPasteConfig` command accepts an option in the form:

`-javaHome`, ***javaHomePath***
Absolute path of JDK.

`-al, -archiveLoc` ***archivePath***
Absolute path of archive location.

`-mpl, -movePlanLoc` ***planPath***
Absolute path to the moveplan extracted during extract plan operation.

`-tih, -targetInstanceHomeLoc` ***instHomePath***
Absolute path of instance home under which Oracle Unified Directory configuration will be restored.

`-toh, -targetOracleHomeLoc` ***oracleHomePath***
Absolute path of the Oracle home associated with the instance home.

`-tin, -targetInstanceName` ***instanceName***
Target instance name. If specified, must be consistent with target instance path. This parameter is **optional.**

`-h, -help`
Show this help message and exit. This parameter is **optional.**

`-ldl, -logDirLoc` ***logPath***
Existing log directory location. Default location is system temporary location. This parameter is **optional.**

#### A.2.11.4 Examples

The following examples show how to use the `oudPasteConfig` command.

***Example A–36 Pasting the Configuration***

The following command allows you to paste the configuration.

```
$ OUD_ORACLE_HOME/bin/oudPasteConfig -javaHome /usr/jdk -al /tmp/oud.jar \
-tih /tmp/asinst_2 -toh /tmp/Oracle_OUD1 \
-mpl /tmp/moveplan.xml -tin asinst_2
```

***Example A–37 Running the Help Command Option***

```
$ OUD_ORACLE_HOME/bin/oudPasteConfig -javaHome /usr/jdk -help
```

#### A.2.11.5 Location

- UNIX and Linux: *OUD_ORACLE_HOME*`/bin/oudPasteConfig`
- Windows: *OUD_ORACLE_HOME**bat**oudPasteConfig.bat*

#### A.2.11.6 Related Commands

- Section A.2.9, "oudCopyConfig"
- Section A.2.10, "oudExtractMovePlan"

### A.2.12 oud-replication-gateway-setup

The `oud-replication-gateway-setup` command is used to setup the replication gateway instance.

#### A.2.12.1 Synopsis

`oud-replication-gateway-setup` [*options*]

#### A.2.12.2 Description

The `oud-replication-gateway-setup` command installs and configures a replication gateway instance, including specifying the ports on which it will listen, the DN and password for the initial root user, and the base DN for the replication gateway data. The replication gateway allows replication to work between a set of Oracle Directory Server Enterprise Edition servers and a set of Oracle Unified Directory servers.

The utility can be run in one of the following modes:

- **Graphical-user interface (GUI) mode**. GUI mode is the default and recommended installation option. The `oud-replication-gateway-setup` GUI provides an

easy interface for installing and configuring replication servers in replicated multi-network environments. GUI mode also allows for easy server setup using SSL or StartTLS if desired.

The utility launches the graphical installer and creates the Oracle Unified Directory instance in *OUD_BASE_LOCATION/INSTANCE_DIR*. The default instance directory name is `asinst_1`, with subsequent instances on the same server named `asinst_2`, `asinst_3`, and so on.

- **Command-line interface (CLI) mode**. The command-line mode is either interactive or non-interactive. The interactive CLI mode prompts you for any required information before the configuration begins, and is used with the `--cli` option, or if no GUI is available.

  The utility launches the command-line installer and creates the Oracle Unified Directory instance in *OUD_BASE_LOCATION/INSTANCE_DIR*. The default instance directory name is `asinst_1`, with subsequent instances on the same server named `asinst_2`, `asinst_3`, and so on.

  The non-interactive CLI mode enables you to set up the server without user intervention. Use the `--no-prompt` and the `--quiet` options to suppress interactivity and output information, respectively.

When the `oud-replication-gateway-setup` command is run without any options, it starts in GUI mode but falls back to interactive command-line mode if no GUI is available. To run the setup in interactive command-line mode use the `--cli` option. Note that no options are allowed if the command is run in GUI mode.

### A.2.12.3 Options

The `oud-replication-gateway-setup` command accepts an option in either its short form (for example, `-i`) or its long form equivalent (for example, `--cli`).

`-i, --cli`
Use the command line install. If not specified the graphical interface will be launched. The rest of the options (excluding help and version) will only be taken into account if this option is specified.

### A.2.12.4 Replication Gateway Configuration Options

`-h, --hostname` ***hostname***
The fully-qualified name of the host where the replication gateway will be installed. The Oracle Directory Server Enterprise Edition and Oracle Unified Directory servers in the replication topology must be able to access this hostname. If this option is not provided, a default of `localhost` is used.

`--adminConnectorPort` ***port***
Specifies the port on which the administration connector should listen for administration traffic. For information about the administration connector, see Section 14.3, "Managing Administration Traffic to the Server." The configuration and administration tools use this port to connect to the replication gateway. The default value is `4444`.

`--replicationPortForLegacy` ***port***
Specifies the port that is used by the Oracle Directory Server Enterprise Edition server to communicate with the replication gateway to replicate contents.

`-S, --skipPortCheck`
Do not make any attempt to determine whether the specified port is available. Normally, when this option is not present, the `oud-replication-gateway-setup` command verifies if that port is in use or not, and if not in use then the user running the command can bind to that port. With the `--skipPortCheck` option, the `oud-replication-gateway-setup` command skips the port check.

`-D, --rootUserDN` ***rootUserDN***
DN for the initial root user for the replication gateway.

`-j, --rootUserPasswordFile` ***rootUserPasswordFile***
Path to a file containing the password for the initial root user for the replication gateway.

`-O, --doNotStart`
Do not start the replication gateway when the configuration is completed.

`-b, --baseDN` ***baseDN***
Specify the base DN of the data to be replicated between the Oracle Unified Directory and the Oracle Directory Server Enterprise Edition server. Multiple base DN's can be provided by using this option multiple times.

### A.2.12.5 Oracle Directory Server Enterprise Edition Server Options

`--hostNameLegacy` ***hostname***
The fully-qualified name of the host or IP address of the Oracle Directory Server Enterprise Edition server whose contents will be replicated.

`--portLegacy` ***port***
Specifies the port number of the Oracle Directory Server Enterprise Edition server whose contents will be replicated. This port is used by the replication mechanism to replicate contents.

`--bindDNLegacy` ***bindDN***
Specifies the DN that is used to bind the Oracle Directory Server Enterprise Edition server whose contents will be replicated.

`--bindPasswordFileLegacy` ***bindPasswordFile***
Specifies the file that stores the password that is used to bind the Oracle Directory Server Enterprise Edition server whose contents will be replicated.

`--secureReplicationLegacy`
Specifies if the replication updates between the Oracle Directory Server Enterprise Edition server and the replication gateway are sent encrypted or not. If you enable this option, then you must specify the certificate to be used by the server using the options in `Replication Gateway Security Options` and the port specified using argument `--portLegacy` must be an LDAP port.

`--clientAuthenticationToLegacy`
Uses client authentication to send replication updates from the replication gateway to the Oracle Directory Server Enterprise Edition server. You can use this argument only if attribute `--secureReplicationLegacy` is used.

`--certFileForClientAuthenticationToLegacy` ***certificateFile***
Specifies the file that contains the certificate to be used in client authentication mode when the replication gateway connects to the Oracle Directory Server Enterprise

Edition server to send replication updates. The file must contain the certificate in X.509 format.

`--doNotSendUpdatesToLegacyServer`
Do not propagate the updates made in the Oracle Unified Directory servers to the Oracle Directory Server Enterprise Edition server. If you use this option the changes made directly in the Oracle Unified Directory servers will not be propagated to the Oracle Directory Server Enterprise Edition servers replication topology.

`--doNotUpdateTrustStoreWithLegacyCertsArg`
If you specify this argument and the replication gateway sends replication updates to the Oracle Directory Server Enterprise Edition server using an encrypted communication (specified using the `--secureReplicationLegacy` argument), then you will have to update the trust store used by the replication gateway with the server certificate of the Oracle Directory Server Enterprise Edition server for replication to work.

`--clientAuthenticationFromLegacy`
Uses client authentication to send replication updates from the Oracle Directory Server Enterprise Edition server to the replication gateway. You can use this argument only if attribute `--secureReplicationLegacy` is used.

### A.2.12.6 Replication Gateway Security Options

`--generateSelfSignedCertificate`
Generates a self-signed certificate that the replication gateway will use as server certificate when accepting encrypted connections from the Oracle Directory Server Enterprise Edition server.

`--usePkcs11Keystore`
Use a certificate in a PKCS#11 token that the replication gateway will use as server certificate when accepting encrypted connections from the Oracle Directory Server Enterprise Edition server.

`--useJavaKeystore` ***keyStorePath***
Specifies the path of a Java Key Store (JKS) that contains a certificate that the replication gateway will use as server certificate when accepting encrypted connections from the Oracle Directory Server Enterprise Edition server.

`--useJCEKS` ***keyStorePath***
Specifies the path of a JCEKS that contains a certificate that the replication gateway will use as server certificate when accepting encrypted connections from the Oracle Directory Server Enterprise Edition server.

`--usePkcs12keyStore` ***keyStorePath***
Path of a PKCS#12 key store that contains the certificate that the replication gateway will use as server certificate when accepting encrypted connections from the Oracle Directory Server Enterprise Edition server.

`--gatewayKeyStorePasswordFile` ***keyStorePasswordFile***
Specifies the file containing the certificate key store PIN. It is required to access the key store that contains the certificate (JKS, JCEKS, PKCS#12, or PKCS#11) that the replication gateway will use as server certificate. This is required when the replication gateway is configured for encrypted replication communication with the Oracle Directory Server Enterprise Edition server.

`--gatewayCertNickname` ***nickname***
Specifies the nickname of the certificate that the replication gateway will use when accepting encrypted connections from the Oracle Directory Server Enterprise Edition server.

### A.2.12.7 Oracle Unified Directory Server Options

`--hostNameNg` ***hostname***
The fully-qualified name of the host or IP address of the Oracle Unified Directory server whose contents will be replicated.

`--portNg` ***port***
Specifies the port number of the Oracle Unified Directory server whose contents will be replicated.

`--bindDNNg` ***bindDN***
Specifies the DN that is used to bind the Oracle Unified Directory server whose contents will be replicated. If this attribute is not specified the global administrator is used to bind.

`--bindPasswordFileNg` ***bindPasswordFile***
Specifies the file that stores the password that is used to bind the Oracle Unified Directory server whose contents will be replicated. If no bind DN is specified for this server the password of the global administrator is used to bind.

`--replicationPortNg` ***port***
Specifies the port used by the replication mechanism in the Oracle Unified Directory server to communicate with other Oracle Unified Directory servers. You have to specify this option only if you have not configured replication for the provided Oracle Unified Directory server.

`--secureReplicationNg`
Specifies whether or not the communication through the replication port of the Oracle Unified Directory server is encrypted or not. This option is only taken into account if replication is not configured on the Oracle Unified Directory server.

`-I, --adminUID` ***adminUID***
Specifies the user ID of the Global Administrator to use to bind to the Oracle Unified Directory server. If you have not defined a Global Administrator in the Oracle Unified Directory, then the Global Administrator is created using the provided data. The default value is admin.

`--adminPasswordFile` ***bindPasswordFile***
The file that contains the password of the global administrator.

### A.2.12.8 Secure Connection Options

`-o, --saslOption` ***name*=*value***
These are SASL bind options.

SASL is not supported for a proxy instance.

`-X, --trustAll`
Trust all server SSL certificates that the server presents. This option can be used for convenience and testing purposes, but for security reasons a trust store should be used to determine whether the client should accept the server certificate.

`-P, --trustStorePath` ***path***
Use the client trust store certificate in the specified path. This option is not needed if `--trustAll` is used, although a trust store should be used when working in a production environment.

`-U, --trustStorePasswordFile` ***path***
Use the password in the specified file to access the certificates in the client trust store. This option is only required if `--trustStorePath` is used and the specified trust store requires a password in order to access its contents (most trust stores do not require this).

`-K, --keyStorePath` ***path***
Use the client keystore certificate in the specified path.

`-u, --keyStorePasswordFile` ***filename***
Use the password in the specified file to access the certificates in the client keystore. This option is only required if `--keyStorePath` is used.

`-N, --certNickname` ***nickname***
Use the specified certificate for SSL client authentication.

`--connectTimeout` ***timeout***
Specifies the maximum length of time (in milliseconds) that can be taken to establish a connection. Use `0`to specify no time out. The default value is 30000.

### A.2.12.9 Command Input/Output Options

`-n, --no-prompt`
Run `setup` in non-interactive mode. If some data in the command is missing, the user will not be prompted and the command will fail.

`-Q, --quiet`
Run in quiet mode. No output will be generated unless a significant error occurs during the process.

`-v, --verbose`
Run in verbose mode, displaying diagnostics on standard output.

`--noPropertiesFile`
Indicate that the command will not use a properties file to get the default command-line options.

`--propertiesFilePath` ***path***
Specify the path to the properties file that contains the default command-line options.

### A.2.12.10 General Options

`-?, -H, --help`
Display command-line usage information for the command and exit without making any attempt to stop or restart the server.

`--version`
Display the version information for the directory server and exit rather than attempting to run this command.

### A.2.12.11 Examples

The following examples show how to use the replication server commands.

***Example A–38 Running `oud-replication-gateway-setup` in GUI Mode***

The following command runs an installation in GUI mode:

```
$ oud-replication-gateway-setup
```

The utility launches the graphical installer and creates the Oracle Unified Directory instance in *OUD_BASE_LOCATION/INSTANCE_DIR*. The default instance directory name is `asinst_1`, with subsequent instances on the same server named `asinst_2`, `asinst_3`, and so on. To specify a different instance name, set the *INSTANCE_NAME* environment variable before you run the setup, for example:

```
$ export INSTANCE_NAME=my-oud-instance
```

The GUI is launched and provides several screens that walk you through setting up your replication server in standalone or replicated environments. You also have the option to set up SSL or StartTLS certificates.

***Example A–39 Running `oud-replication-gateway-setup` in Interactive Mode From the Command Line***

The `oud-replication-gateway-setup` command can be run in interactive mode, where you are prompted for installation options. To run `oud-replication-gateway-setup` in interactive mode, type the following command:

```
$ oud-replication-gateway-setup --cli
```

The command prompts you for the required setup values. Press Enter or Return to accept the default, or enter a value at the prompt.

The utility launches the command-line installer and creates the Oracle Unified Directory instance in *OUD_BASE_LOCATION/INSTANCE_DIR*. The default instance directory name is `asinst_1`, with subsequent instances on the same server named `asinst_2`, `asinst_3`, and so on. To specify a different instance name, set the *INSTANCE_NAME* environment variable before you run the setup, for example:

```
$ export INSTANCE_NAME=my-oud-instance
```

### A.2.12.12 Exit Codes

`0`
Successful completion or successful no-op.

`1`
Error unexpected. Potential bug.

`2`
Error user data. Cannot parse options, or data provided by user is not valid.

`4`
Error initializing server.

#### A.2.12.13 Using a Properties File

The directory server supports the use of a *properties file* that passes in any default option values used with the `oud-replication-gateway-setup` command. The properties file is convenient when working in different configuration environments, especially in scripted or embedded applications. For more information, see Section A.1.2, "Using a Properties File With Server Commands."

All the `oud-replication-gateway-setup` options can be stored in a properties file. Entries in the properties file have the following format:

```
toolname.propertyname=propertyvalue
```

For example:

```
oud-replication-gateway-setup.hostname=grevalon:1444
```

#### A.2.12.14 Log Files

The `oud-replication-gateway-setup` command writes a log file named `oud-setup-`*IDnumber* where *IDnumber* is a decimal number. The log files are located at these paths:

- UNIX (Solaris): `/var/tmp/`
- Linux: `/tmp/`
- Windows: `%TEMP%`

  By default, this folder is `C:\Documents and Settings\User\Local Settings\Temp`.

#### A.2.12.15 Location

The `oud-replication-gateway-setup` command is located at these paths:

- UNIX and Linux: *OUD_BASE_LOCATION*`/`*OUD_ORACLE_HOME*`/oud-replication-gateway-setup`
- Windows: *OUD_BASE_LOCATION*`\`*OUD_ORACLE_HOME*`\oud-replication-gateway-setup.bat`

#### A.2.12.16 Related Commands

- Section A.2.13, "oud-setup"
- Section A.2.14, "oud-proxy-setup"

### A.2.13 oud-setup

The `oud-setup` command installs and minimally configures a directory server instance.

This command sets up a *directory server* instance. For information about setting up a proxy server instance, see Section A.2.14, "oud-proxy-setup."

#### A.2.13.1 Synopsis

`oud-setup` [*options*]

### A.2.13.2 Description

The `oud-setup` command installs and configure a directory server instance, including specifying the ports on which it will listen, the DN and password for the initial root user, the base DN for the directory data, and the manner in which the database should be populated. It can be run in one of the following modes:

- **Graphical-user interface (GUI) mode**. GUI mode is the default and recommended installation option. The `oud-setup` GUI provides an easy interface for installing and configuring standalone directory servers or replication servers in replicated multi-network environments. GUI mode also allows for easy server setup using SSL or StartTLS if desired.

  The utility launches the graphical installer and creates the Oracle Unified Directory instance in *OUD_BASE_LOCATION/INSTANCE_DIR*. The default instance directory name is `asinst_1`, with subsequent instances on the same server named `asinst_2`, `asinst_3`, and so on.

- **Command-line interface (CLI) mode**. The command-line mode is either interactive or non-interactive. The interactive CLI mode prompts you for any required information before the configuration begins, and is used with the `--cli` option, or if no GUI is available.

  The utility launches the command-line installer and creates the Oracle Unified Directory instance in *OUD_BASE_LOCATION/INSTANCE_DIR*. The default instance directory name is `asinst_1`, with subsequent instances on the same server named `asinst_2`, `asinst_3`, and so on.

  The non-interactive CLI mode enables you to set up the server without user intervention. Use the `--no-prompt` and the `--quiet` options to suppress interactivity and output information, respectively.

When the `oud-setup` command is run without any options, it starts in GUI mode but falls back to interactive command-line mode if no GUI is available. To run `oud-setup` in command-line mode, use the `--cli` option. The options that can be provided are listed below. Note that no options are allowed if the command is run in GUI mode.

### A.2.13.3 Options

The `oud-setup` command accepts an option in either its short form (for example, `-a`) or its long form equivalent (for example, `--addBaseEntry`).

`-a, --addBaseEntry`
Indicates whether to create the base entry in the directory server database.

`-i, --cli`
Run the `setup` command in command-line interactive mode rather than in GUI mode. If `setup` is run without the `--cli` option, it cannot accept other options.

`-b, --baseDN` ***baseDN***
Use the base DN for user information in the Directory Server. The default value for this option is `dc=example,dc=com`. Multiple base DNs can be specified by providing this option multiple times.

`-l, --ldifFile` ***filename***
Use the specified LDIF file to populate the database. Data can be imported from multiple files by providing this option multiple times, in which case the files are processed in the order they are provided in the option list. This option must not be used in conjunction with either the `--addBaseEntry` or `--sampleData` option. If this option is not provided, then the database is left empty.

`-R, --rejectFile` ***filename***
Write rejected entries to the specified file. Rejected entries occur if they do not comply with the default schema during an import using the `-l` or `--ldifFile` option.

`--skipFile` ***filename***
Write skipped entries to the specified file. Skipped entries occur if entries cannot be placed under any specified base DN during an import using the `-l` or `--ldifFile` option.

`-d, --sampleData` ***number-of-entries***
Populate the database with the specified number of sample user entries. The entries are generated by using the MakeLDIF facility of the `import` command and are based on the default `example.template` template. This option must not be used in conjunction with either `--addBaseEntry` or `--ldifFile`. If this option is not provided, then the database is left empty.

`--eus`
Configure the server for Oracle's Enterprise User Security (EUS).

`-p,--ldapPort` ***port***
Contact the directory server at the specified port. If it is not provided, then the default port of `1389` as non-root and `389` as root is used.

`--adminConnectorPort` ***port***
Specifies the port on which the administration connector should listen for administration traffic. For information about the administration connector, see Section 14.3, "Managing Administration Traffic to the Server." The default value is `4444`.

`-x, --jmxPort` ***port***
Specify the port for a JMX MBeans server connection. The default value for this option is `1689`.

`-S, --skipPortCheck`
Do not make any attempt to determine whether the specified port is available. Normally, when this option is not present, the `oud-setup` command verifies that the port is not in use and that the user running the setup command can bind to that port. With the `--skipPortCheck` option, the `oud-setup` command skips the port check.

`-D, --rootUserDN` ***rootUserDN***
Use the specified root user DN to authenticate the directory server. This option is used when performing simple authentication and is not required if SASL authentication is used. The default value for this option is `cn=Directory Manager`.

`-j, --rootUserPasswordFile` ***filename***
Specifies the file containing the password for the initial root user while authenticating the directory server.

`-O, --doNotStart`
Do not start the directory server when the configuration is completed.

`-q, --enableStartTLS`
Enable StartTLS to allow secure communication with the directory server by using the LDAP port.

`-Z, --ldapsPort` ***port***
Contact the directory server at the specified port for LDAP SSL (LDAPS) communication. The LDAPS port will be configured and SSL will be enabled only if this option is explicitly specified. The default value is `1636`.

`--generateSelfSignedCertificate`
Generate a self-signed certificate that the directory server should use when accepting SSL-based connection or performing StartTLS negotiation.

`-h, --hostname` ***host***
The name of the directory server host or IP address that is used to generate the self-signed certificate. This argument is considered only if the self-signed certificate argument, `--generateSelfSignedCertificate` is specified

`--usePkcs11Keystore`
Use a certificate in a PKCS#11 format that the server should use when accepting SSL-based connections or performing StartTLS negotiation

`--useJavaKeystore` ***path***
Specify the path to the Java Keystore (JKS) that contains the server certificate.

`--useJCEKS` ***path***
Specify the path to the Java Cryptography Extension Keystore (JCEKS) that contains the server certificate.

`--usePkcs12Keystore` ***path***
Specify the path to the PKCS#12 keystore that contains the server certificate.

`-u, --keyStorePasswordFile` ***filename***
Use the password in the specified file to access the certificate keystore. A password is required when you specify an existing certificate (JKS, JCEKS, PKCS#11, or PKCS#12) as a server certificate.

`-N, --certNickname` ***nickname***
Use the specified certificate for SSL or StartTLS client authentication.

`-e, --enableWindowsService`
Enable the directory server as a Windows service. For Windows-platforms only.

### A.2.13.4 Command Input/Output Options

`-n, --no-prompt`
Run `setup` in non-interactive mode. If some data in the command is missing, the user will not be prompted and the command will fail.

`--noPropertiesFile`
Indicate that the command will not use a properties file to get the default command-line options.

`--propertiesFilePath` ***path***
Specify the path to the properties file that contains the default command-line options.

`-Q, --quiet`
Run in quiet mode. No output will be generated unless a significant error occurs during the process.

`-v, --verbose`
Run in verbose mode, displaying diagnostics on standard output.

### A.2.13.5 General Options

`-?, -H, --help`
Display command-line usage information for the command and exit without making any attempt to stop or restart the server.

`-V, --version`
Display the version information for the directory server and exit rather than attempting to run this command.

### A.2.13.6 Examples

The following examples show how to use the directory server commands.

***Example A–40 Running `oud-setup` in GUI Mode***

The following command runs an installation in GUI mode:

```
$ oud-setup
```

The GUI is launched and provides several screens that walk you through setting up your directory server in standalone or replicated environments. You also have the option to set up SSL or StartTLS certificates.

The utility creates the Oracle Unified Directory instance in *OUD_BASE_LOCATION /INSTANCE_DIR*. The default instance directory name is `asinst_1`, with subsequent instances on the same server named `asinst_2`, `asinst_3`, and so on. To specify a different instance name, set the INSTANCE_NAME environment variable before you run the setup, for example:

```
$ export INSTANCE_NAME=my-oud-instance
```

***Example A–41 Running `oud-setup` in Interactive Mode From the Command Line***

The `oud-setup` command can be run in interactive mode, where you are prompted for installation options. To run `oud-setup` in interactive mode, type the following command:

```
$ oud-setup --cli
```

The command prompts you for the required setup values. Press Enter or Return to accept the default, or enter a value at the prompt.

The utility launches the command-line installer and creates the Oracle Unified Directory instance in *OUD_BASE_LOCATION/INSTANCE_DIR*. The default instance directory name is `asinst_1`, with subsequent instances on the same server named `asinst_2`, `asinst_3`, and so on. To specify a different instance name, set the *INSTANCE_NAME* environment variable before you run the setup, for example:

```
$ export INSTANCE_NAME=my-oud-instance
```

***Example A–42 Running `oud-setup` in Non-Interactive CLI Mode***

The non-interactive CLI mode enables you to create installation scripts with the `oud-setup` command when many directory server instances must be configured for large replicated environments. This mode requires the `--no-prompt` and `--quiet`

options to be provided. If no option is present, the `oud-setup` command defaults to interactive mode.

The following command runs the installation in non-interactive (`--no-prompt`) and quiet (`-Q`) modes. It sets the LDAP port (`-p`), the administration connector port (`--adminConnectorPort`), the root DN (`-D`), the file containing the root DN password (`-j`), and adds a base entry (`-a`) with the specified base DN (`-b`),

```
$ oud-setup --cli --no-prompt -Q -p 1389 --adminConnectorPort 4444 \
  -D "cn=Directory Manager" -j /path/pwd-file -a -b dc=example,dc=com
```

***Example A–43 Running `oud-setup` in Non-Interactive CLI Mode With LDIF Import***

The following command runs the installation in non-interactive (`--no-prompt`) and quiet (`-Q`) modes. It sets the LDAP port (`-p`), the administration connector port (`--adminConnectorPort`), the root DN (`-D`), the file containing the root DN password (`-j`), and adds the baseDN (`-b`) with data imported from an LDIF file (`-l`).

```
$ oud-setup --cli --no-prompt -Q -p 1389 --adminConnectorPort 4444 \
  -D "cn=Directory Manager" -j /path/pwd-file -b dc=example,dc=com \
  -l "/home/ldif/company.ldif"
```

***Example A–44 Running `oud-setup` in Non-Interactive Mode With Sample Entry Generation***

The following command runs the installation in non-interactive (`--no-prompt`) and quiet (`-Q`) modes. It sets the LDAP port (`-p`), the administration connector port (`--adminConnectorPort`), the root DN (`-D`), the file containing the root DN password (`-j`), the baseDN (`-b`) and generates 2000 sample entries (`-d`).

```
$ oud-setup --cli --no-prompt -Q -p 1389 --adminConnectorPort 4444 \
  -D "cn=Directory Manager" -j /path/pwd-file -b dc=example,dc=com -d 2000
```

***Example A–45 Running `oud-setup` on Windows***

The following command enables the directory server to run as a Windows service (`-e`). It sets the LDAP port (`-p`), the administration connector port (`--adminConnectorPort`), the JMX port (`-x`), the rootDN (`-D`), the file containing the root DN password (`-j`), and the baseDN (`-b`), and generates 10000 sample entries.

```
C:\> oud-setup.bat --cli -e -p 1389 --adminConnectorPort 4444 -x 1689 \
  -D "cn=Directory Manager" -j /path/pwd-file -b dc=example,dc=com -d 10000
```

The utility launches the graphical installer and creates the Oracle Unified Directory instance in *OUD_BASE_LOCATION/INSTANCE_DIR*. The default instance directory name is `asinst_1`, with subsequent instances on the same server named `asinst_2`, `asinst_3`, and so on. To specify a different instance name, set the `INSTANCE_NAME` environment variable before you run the setup, for example:

```
$ export INSTANCE_NAME=my-oud-instance
```

### A.2.13.7 Exit Codes

`0`
Successful completion or successful no-op.

1
Error unexpected. Potential bug.

2
Error user data. Cannot parse options, or data provided by user is not valid.

4
Error initializing server.

### A.2.13.8 Using a Properties File

The directory server supports the use of a *properties file* that passes in any default option values used with the `oud-setup` command. The properties file is convenient when working in different configuration environments, especially in scripted or embedded applications. For more information, see Section A.1.2, "Using a Properties File With Server Commands."

The following options can be stored in a properties file:

- `certNickname`
- `hostname`
- `keyStorePasswordFile`

All the preceding `oud-setup` options can be stored in a properties file. Entries in the properties file have the following format:

```
toolname.propertyname=propertyvalue
```

For example:

```
oud-setup.hostname=grevalon:1444
```

### A.2.13.9 Log Files

The `oud-setup` command writes a log file named `oud-setup-`*IDnumber* where *IDnumber* is a decimal number. The log files are located at these paths:

- UNIX (Solaris): `/var/tmp/`
- Linux: `/tmp/`
- Windows: `%TEMP%`

  By default, this folder is `C:\Documents and Settings\User\Local Settings\Temp`.

### A.2.13.10 Location

The `oud-setup` command is located at these paths:

- UNIX and Linux: *OUD_BASE_LOCATION/OUD_ORACLE_HOME*`/oud-setup`
- Windows: *OUD_BASE_LOCATION*`\`*OUD_ORACLE_HOME*`\oud-setup.bat`

### A.2.13.11 Related Commands

- Section A.2.12, "oud-replication-gateway-setup"
- Section A.2.14, "oud-proxy-setup"

## A.2.14 oud-proxy-setup

The `oud-proxy-setup` command manages the setup and configuration of a proxy server instance.

### A.2.14.1 Synopsis

`oud-proxy-setup` [*options*]

### A.2.14.2 Description

The `oud-proxy-setup` command installs and configures a proxy server instance, including specifying the ports on which it will listen, the DN and password for the initial root user, the base DN for the directory data, authentication methods, as well load balancing, distribution, and a global index catalog, depending on the deployment chosen.

The `oud-proxy-setup` can only be launched once. It can be run in one of the following modes:

- **Graphical-user interface (GUI) mode**. GUI mode is the default and recommended installation option. The setup GUI provides an easy interface for defining and deploying the proxy instance.

  The utility launches the graphical installer and creates the proxy instance in *OUD_BASE_LOCATION/INSTANCE_DIR*. The default instance directory name is `asinst_1`, with subsequent instances on the same server named `asinst_2`, `asinst_3`, and so on.

- **Command-line interface (CLI) mode**. The command-line setup defines the proxy port, host name, and security configuration. If you specify the `--cli` option with `oud-proxy-setup` then you must provide the required values in the command line, else the default values are used. If you do not provide any value for a parameter that has no default value then the setup fails, and an error message is displayed.

  The utility launches the command-line installer and creates the proxy instance in *OUD_BASE_LOCATION/INSTANCE_DIR*. The default instance directory name is `asinst_1`, with subsequent instances on the same server named `asinst_2`, `asinst_3`, and so on.

  The proxy setup CLI mode prompts the user to accept the license. Use the `--no-prompt` option to automatically accept the license.

### A.2.14.3 Options

The `oud-proxy-setup` command accepts an option in either its short form (for example, `-i`) or its long form equivalent (for example, `--cli`).

`-i, --cli`
Use the command line install. If not specified the graphical interface will be launched. The rest of the options (excluding help and version) will only be taken into account if this option is specified.

`-p, --ldapPort` ***port***
Port on which the Directory Server should listen for LDAP communication. The default value is `389`.

`--adminConnectorPort` ***port***
Port on which the Administration Connector should listen for communication. The default value is `4444`.

`-S, --skipPortCheck`
Skip the check to determine whether the specified ports are usable.

`-D, --rootUserDN` ***rootUserDN***
DN for the initial root user for the proxy server.

`-j, --rootUserPasswordFile` ***rootUserPasswordFile***
Path to a file containing the password for the initial root user for the proxy server.

`-q, --enableStartTLS`
Enable StartTLS to allow secure communication with the server using the LDAP port.

`-Z, --ldapsPort` ***port***
Port on which the Directory Server should listen for LDAP SSL (LDAPS) communication. The LDAPS port will be configured and SSL will be enabled only if this argument is explicitly specified. The default value is `636`.

`--generateSelfSignedCertificate`
Generate a self-signed certificate that the server should use when accepting SSL-based connections or performing StartTLS negotiation.

`--usePkcs11keyStore` ***keyStorePath***
Path of a PKCS#11 key store containing the certificate that the server should use when accepting SSL-based connections or performing StartTLS negotiation.

`--useJavaKeystore` ***keyStorePath***
Path of a Java Key Store (JKS) containing a certificate to be used as the server certificate.

`--useJCEKS` ***keyStorePath***
Path of a JCEKS containing a certificate to be used as the server certificate.

`--usePkcs12keyStore` ***keyStorePath***
Path of a PKCS#12 key store containing the certificate that the server should use when accepting SSL-based connections or performing StartTLS negotiation.

`-u, --keyStorePasswordFile` ***keyStorePasswordFile***
Certificate key store PIN file. A PIN is required when you specify to use an existing certificate (JKS, JCEKS, PKCS#12, or PKCS#11) as server certificate.

`-N, --certNickname` ***nickname***
Nickname of the certificate that the server should use when accepting SSL-based connections or performing StartTLS negotiation.

`-O, --doNotStart`
Do not start the server when the configuration is completed.

### A.2.14.4 Command Input/Output Options

`-Q, --quiet`
Run in quiet mode. No output will be generated unless a significant error occurs during the process.

`-v, --verbose`
Use verbose mode

`--propertiesFilePath` ***path***
Specify the path to the properties file that contains the default command-line options.

`--noPropertiesFile`
Indicate that a properties file will not be used to get the default command-line options.

`-n, --no-prompt`
Perform an installation in non-interactive mode, for license acceptance only. If some data in the command is missing the user will not be prompted and the command will fail.

### A.2.14.5 General Options

`-?, -H, --help`
Display command-line usage information for the command and exit without making any attempt to stop or restart the server.

`-V, --version`
Display the version information for the directory server and exit rather than attempting to run this command.

### A.2.14.6 Examples

The following examples show how to use the `oud-proxy-setup` command.

***Example A–46 Running*** `oud-proxy-setup` ***in GUI Mode***

The following command runs an installation in GUI mode:

```
$ oud-proxy-setup
```

The utility launches the graphical installer and creates the proxy instance in *OUD_BASE_LOCATION/INSTANCE_DIR*. The default instance directory name is `asinst_1`, with subsequent instances on the same server named `asinst_2`, `asinst_3`, and so on. To specify a different instance name, set the `INSTANCE_NAME` environment variable before you run the setup, for example:

```
$ export INSTANCE_NAME=my-oud-proxy-instance
```

***Example A–47 Running*** `oud-proxy-setup` ***in Non-Interactive CLI Mode***

The non-interactive CLI mode enables you to create installation scripts with the setup command when many proxy server instances must be configured for large replicated environments. This mode requires the `--no-prompt` and `--quiet` options to be provided. If no option is present, the setup command defaults to interactive mode.

The following command runs the installation in non-interactive (`--no-prompt`) and quiet (`-Q`) modes. It sets the LDAP port (`-p`), the administration connector port (`--adminConnectorPort`), the root DN (`-D`), and the file containing the root DN password (`-j`).

```
$ oud-proxy-setup --cli --no-prompt -Q -p 1389 --adminConnectorPort 4444 \
  -D "cn=Directory Manager" -j /path/pwd-file
```

The utility launches the command-line installer and creates the proxy instance in *OUD_BASE_LOCATION/INSTANCE_DIR*. The default instance directory name is `asinst_1`, with subsequent instances on the same server named `asinst_2`,

asinst_3, and so on. To specify a different instance name, set the INSTANCE_NAME environment variable before you run the setup, for example:

```
$ export INSTANCE_NAME=my-oud-proxy-instance
```

#### A.2.14.7 Exit Codes

An exit code of 0 indicates that the operation completed successfully. A nonzero exit code indicates that an error occurred during processing.

#### A.2.14.8 Log Files

The oud-proxy-setup command writes a log file named oud-proxy-setup.log, once the setup in complete. The log file is located at these paths:

- UNIX (Solaris): /var/tmp/
- Linux: /tmp/
- Windows: The %TEMP% folder. By default, this folder is C:\Documents and Settings*user*\Local Settings\Temp

#### A.2.14.9 Location

- UNIX and Linux: *OUD_BASE_LOCATION*/*OUD_ORACLE_HOME*/oud-proxy-setup
- Windows: *OUD_BASE_LOCATION**OUD_ORACLE_HOME*\oud-proxy-setup.bat

#### A.2.14.10 Related Commands

Section A.2.12, "oud-replication-gateway-setup"

Section A.2.17, "stop-ds"

### A.2.15 start-ds

The start-ds command starts an installed server instance.

#### A.2.15.1 Synopsis

start-ds [*options*]

#### A.2.15.2 Description

The start-ds command is used to start the server and to provide general server information.

You can run start-ds without any options, which starts the server as a background process. In this case, the script will not exit until the server has either started successfully or has encountered an error that prevents it from starting.

On UNIX systems, the server will not start if it cannot log the process ID at *INSTANCE_DIR*/logs/server.pid. Ensure that the file is writable by the user account that the server uses.

#### A.2.15.3 Options

The start-ds command accepts an option in either its short form (for example, -N) or its long form equivalent (for example, --nodetach).

`-L, --useLastKnownGoodConfig`
Attempt to start using the configuration that was in place at the last successful startup (if it is available) rather than using the current active configuration.

`-N, --nodetach`
Start the server as a foreground process that does not detach from the terminal. When the server is running in this mode, it can be stopped by using the `stop-ds` command from another window, or by pressing `Control+C` in the terminal window in which the server is running.

`-s, --systemInfo`
Display general information about the system on which the server is installed, including the instance and installation paths, and then exit rather than attempting to start the server.

`-t, --timeout` ***seconds***
Wait no longer than the maximum time (in seconds) before the command returns. (The server continues the startup process, regardless). A value of `0` indicates an infinite timeout, which means that the command returns only when the server startup is completed. The default value is 60 seconds. This option cannot be used with the `-N, --nodetach` option.

### A.2.15.4 Command Input/Output Options

`-Q, --quiet`
Run in quiet mode. No output is generated unless a significant error occurs during the process.

### A.2.15.5 General Options

`-?, -H, --help`
Display command-line usage information for the command and exit without making any attempt to stop or restart the server.

`-V, --version`
Display the version information for the server and exit rather than attempting to run this command.

### A.2.15.6 Examples

The following examples show how to use the `start—ds` command.

***Example A–48 Starting the Server***

The following command starts the server:

```
$ start-ds
```

***Example A–49 Starting the Server as a Foreground Process***

The following command starts the server as a foreground process. You can stop the server by running the `stop-ds` command from another window or by pressing Control+C in the terminal window in which the server is running.

```
$ start-ds -N

[25/Jul/2007:10:39:17 -0500] category=CORE severity=NOTICE msgID=458887
```

```
msg=The Directory Server has started successfully
```

#### A.2.15.7 Exit Codes

| Exit Code | Description |
|---|---|
| 0 | Server started successfully. |
| 1 | Check error. Generated from incompatible options. |
| 98 | Server already started. |
| 99 | Server must start as a detached process. |
| 100 | Server must start as a non-detached process. |
| 101 | Server must start as a Windows service. |
| 102 | Server must start as a detached process and it is being called from a Windows service. |

#### A.2.15.8 Location

- UNIX and Linux: *INSTANCE_DIR*`/OUD/bin/start-ds`
- Windows: *INSTANCE_DIR*`\OUD\bat\start-ds.bat`

#### A.2.15.9 Related Commands

- Section A.2.17, "stop-ds"

### A.2.16 status

The `status` command displays basic server status information.

#### A.2.16.1 Synopsis

`status` [*options*]

#### A.2.16.2 Description

The `status` command can be used to display basic server information, such as the status of the server (started or stopped), the configured connection handlers, or the list of defined back ends and suffixes.

If the server is started, the `status` command connects to the server over SSL, through the administration connector.

For more information, see Section 14.3, "Managing Administration Traffic to the Server."

If the server is stopped, you must run this command as a user with file system access rights to read the configuration files (particularly the `config.ldif` file).

> **Note:** Certain monitoring data can only be displayed when the server is running (for example, the number of entries in a back end).

### A.2.16.3 LDAP Connection Options

The `status` command contacts the server over SSL through the administration connector (described in Section 14.3, "Managing Administration Traffic to the Server"). These connection options are used to contact the server.

`-D, --bindDN` ***bindDN***
Use the bind DN to authenticate to the server. This option is used when performing simple authentication and is not required if SASL authentication is to be used. The default value for this option is `cn=Directory Manager`.

`-j, --bindPasswordFile` ***filename***
Use the bind password in the specified file when authenticating to the server.

`-K, --keyStorePath` ***path***
Use the client keystore certificate in the specified path.

`-N, --certNickname` ***nickname***
Use the specified certificate for client authentication.

`-o, --saslOption` ***name*=*value***
Use the specified options for SASL authentication.

SASL is not supported for a proxy server instance.

`-P, --trustStorePath` ***path***
Use the client trust store certificate in the specified path. This option is not needed if `--trustAll` is used, although a trust store should be used when working in a production environment.

`-u, --keyStorePasswordFile` ***filename***
Use the password in the specified file to access the certificates in the client keystore. This option is only required if `--keyStorePath` is used.

`-U, --trustStorePasswordFile` ***filename***
Use the password in the specified file to access the certificates in the client trust store. This option is only required if `--trustStorePath` is used and the specified trust store requires a password in order to access its contents (most trust stores do not require this).

`-X, --trustAll`
Trust all server SSL certificates that the server presents. This option can be used for convenience and testing purposes, but for security reasons a trust store should be used to determine whether the client should accept the server certificate.

### A.2.16.4 Command Input/Output Options

`-n, --no-prompt`
Use non-interactive mode. If some data in the command is missing, you are not prompted and the command will fail.

`--noPropertiesFile`
Indicate that the command should not use a properties file to get the default command-line options.

`--propertiesFilePath` ***path***
Specify the path to the properties file that contains the default command-line options.

`-r, --refresh` ***period***
When this argument is specified, the status command will display its contents periodically. Used to specify the period (in seconds) between two displays of the status.

`-s, --script-friendly`
Run in "script friendly" mode. Display the output in a format that can be easily parsed by a script.

### A.2.16.5 General Options

`-?, -H, --help`
Display command-line usage information for the command and exit without making any attempt to stop or restart the server.

`-V, --version`
Display the version information for the server and exit rather than attempting to run this command.

### A.2.16.6 Examples

The following examples show how to use the `status` command.

***Example A–50 Displaying the Server Status***

The following example displays the current status of a standalone server that is currently online:

```
$ status -D "cn=directory manager" -j /path/pwd-file -X -n

          --- Server Status ---
Server Run Status:        Started
Open Connections:         1

          --- Server Details ---
Host Name:                hostname
Administrative Users:     cn=Directory Manager
Installation Path:        /path/OracleUnifiedDirectory
Instance Path:            /path/asinst_1/OUD
Version:                  Oracle Unified Directory 11.1.1.5.0
Java Version:             1.6.0_24
Administration Connector: Port 4444 (LDAPS)

          --- Connection Handlers ---
Address:Port : Protocol    : State
-------------:-------------:---------
--           : LDIF        : Disabled
8989         : Replication : Enabled
0.0.0.0:161  : SNMP        : Disabled
0.0.0.0:636  : LDAPS       : Disabled
0.0.0.0:1389 : LDAP        : Enabled
0.0.0.0:1689 : JMX         : Disabled

          --- Data Sources ---
Base DN:                     dc=example,dc=com
Backend ID:                  userRoot
Entries:                     7
Replication:                 Enabled
Missing Changes:             0
```

```
Age Of Oldest Missing Change: not available
```

### A.2.16.7 Exit Codes

An exit code of 0 indicates that the operation completed successfully. A nonzero exit code indicates that an error occurred during processing.

### A.2.16.8 Using a Properties File

The server supports the use of a *properties file* that passes in any default option values used with the `status` command. The properties file is convenient when working in different configuration environments, especially in scripted or embedded applications. For more information, see Section A.1.2, "Using a Properties File With Server Commands."

The following options can be stored in a properties file:

- `bindDN`
- `bindPasswordFile`
- `certNickname`
- `hostname`
- `keyStorePasswordFile`
- `keyStorePath`
- `port`
- `saslOption`

  SASL is not supported for a proxy server instance.
- `trustAll`
- `trustStorePasswordFile`
- `trustStorePath`

Entries in the properties file have the following format:

```
toolname.propertyname=propertyvalue
```

For example:

```
status.bindPasswordFile=/path/pwd-file
```

### A.2.16.9 Location

- UNIX and Linux: *INSTANCE_DIR*`/OUD/bin/status`
- Windows: *INSTANCE_DIR*`\OUD\bat\status.bat`

## A.2.17 stop-ds

The `stop-ds` command stops a server instance.

### A.2.17.1 Synopsis

`stop-ds` [*options*]

### A.2.17.2 Description

The `stop-ds` command is used to stop or restart the server. It can operate on either a local or remote server instance.

The ability to perform a local stop of the server is currently only available on UNIX based systems. When run locally, `stop-ds` sends a kill signal to the server process. This method of stopping the server is used if `stop-ds` is run without any options and if a PID file (*INSTANCE_DIR*`/OUD/logs/server.pid`) exists.

The remote shutdown mechanism issues an LDAP request to create a task entry in the server. The command can be run from any system that can communicate with the server (local or remote). It can also be used to restart the server. In this case, the server does an "in-core" restart, which reinitializes itself without shutting down the JVM.

When it is run remotely, `stop-ds` communicates with the server over SSL, through the administration connector. For more information, see Section 14.3, "Managing Administration Traffic to the Server."

### A.2.17.3 Options

The `stop-ds` command accepts an option in either its short form (for example, `-D` *bindDN*) or its long form equivalent (for example, `--bindDN` *bindDN*).

`-r,--stopReason` ***reason***
Provide a human-readable reason for the shutdown. If a reason is provided, it appears in the server's error log, and is provided to shut down plugins and shut down listeners.

`-R,--restart`
Restart the server rather than shutting it down. If the `--restart` option is used along with authentication options, the server will reinitialize itself without shutting down the JVM. Because the JVM is not stopped, any configuration changes that require a JVM restart will not take effect. If the `--restart` option is used without authenticating, the server will first stop, then start. A new process will replace the original server.

`-t,--stopTime` ***time***
Indicates the date and time at which the shutdown operation begins as a server task, expressed in the format `YYYYMMDDhhmmss`. A value of `0` causes the shutdown to be scheduled for immediate execution. When this option is used, the operation is scheduled to start at the specified time, after which this command exits immediately.

`-Y,--proxyAs` ***authzID***
Use authorization control during the shutdown request. The value provided for this option should be an authorization ID, which can be in the form `dn:` followed by a user DN or `u:` followed by a user name. Clients will use the proxy authorization v2 control as described in RFC 4370 (`http://www.ietf.org/rfc/rfc4370.txt`).

### A.2.17.4 LDAP Connection Options

The `stop-ds` command contacts the server over SSL through the administration connector (described in Section 14.3, "Managing Administration Traffic to the Server"). These connection options are used to contact the server.

`-D, --bindDN` ***bindDN***
Use the bind DN to authenticate to the server. This option is used when performing simple authentication and is not required if SASL authentication is to be used. The default value for this option is `cn=Directory Manager`.

`-h, --hostname` ***hostname***
Contact the server on the specified hostname or IP address. If this option is not provided, a default of `localhost` is used.

`-j, --bindPasswordFile filename`
Use the bind password in the specified file when authenticating to the server.

`-K, --keyStorePath path`
Use the client keystore certificate in the specified path.

`-N, --certNickname nickname`
Use the specified certificate for client authentication.

`-o, --saslOption` ***name*=*value***
Use the specified options for SASL authentication.

SASL is not supported for a proxy server instance.

`-p, --port` ***port***
Contact the server at the specified administration port. If this option is not provided, a default administration port of `4444` is used.

`-P, --trustStorePath` ***path***
Use the client trust store certificate in the specified path. This option is not needed if `--trustAll` is used, although a trust store should be used when working in a production environment.

`-u, --keyStorePasswordFile` ***filename***
Use the password in the specified file to access the certificates in the client keystore. This option is only required if `--keyStorePath` is used.

`-U, --trustStorePasswordFile` ***filename***
Use the password in the specified file to access the certificates in the client trust store. This option is only required if `--trustStorePath` is used and the specified trust store requires a password in order to access its contents (most trust stores do not require this).

`-X, --trustAll`
Trust all server SSL certificates that the server presents. This option can be used for convenience and testing purposes, but for security reasons a trust store should be used to determine whether the client should accept the server certificate.

### A.2.17.5 Command Input/Output Options

`--noPropertiesFile`
Indicate that a properties file will not be used to get the default command-line options.

`--propertiesFilePath` ***path***
Specify the path to the properties file that contains the default command-line options.

`-Q, --quiet`
Run in quiet mode. No output will be generated unless a significant error occurs during the process.

### A.2.17.6 General Options

`-?, -H, --help`
Display command-line usage information for the command and exit without making any attempt to stop or restart the server.

`--version`
Display the version information for the server and exit rather than attempting to run this command.

### A.2.17.7 Examples

The following examples show how to use the `stop-ds` command.

#### *Example A–51 Stopping a Server Locally*

The following command stops the server:

```
$ stop-ds
```

#### *Example A–52 Stopping a Server Remotely*

The following command stops a remote server instance.

```
$ stop-ds -h remotehost -p 4444 -D "cn=directory manager" -j /path/pwd-file -X
```

#### *Example A–53 Restarting a Server Remotely*

The following command restarts a remote server instance.

```
$ stop-ds -R -h remotehost -p 4444 -D "cn=directory manager" -j /path/pwd-file -X
```

### A.2.17.8 Exit Codes

| Exit Code | Description |
|---|---|
| 0 | Server stopped successfully. |
| 98 | Server already stopped. |
| 99 | Server must be started. |
| 100 | Server must be stopped using a system call. |
| 101 | Server must be restarted using a system call. |
| 102 | Server must be stopped using a protocol. |
| 103 | Server must be stopped as a Windows service. |
| 104 | Server must be restarted as a Windows service. |

### A.2.17.9 Using a Properties File

The server supports the use of a *properties file* that passes in any default option values used with the `stop-ds` command. The properties file is convenient when working in different configuration environments, especially in scripted or embedded applications.

For more information, see Section A.1.2, "Using a Properties File With Server Commands."

The following options can be stored in a properties file:

- `bindDN`
- `bindPasswordFile`
- `certNickname`
- `hostname`
- `keyStorePasswordFile`
- `keyStorePath`
- `saslOption`

  SASL is not supported for a proxy server instance.
- `trustAll`
- `trustStorePasswordFile`
- `trustStorePath`

Entries in the properties file have the following format:

```
toolname.propertyname=propertyvalue
```

For example:

```
stop-ds.trustAll=yes
```

#### A.2.17.10 Location

- UNIX and Linux: *INSTANCE_DIR*`/OUD/bin/stop-ds`
- Windows: *INSTANCE_DIR*`\OUD\bat\stop-ds.bat`

#### A.2.17.11 Related Commands

Section A.2.15, "start-ds"

### A.2.18 uninstall

The `uninstall` command is used to uninstall the server instance. It is applicable for directory servers, proxy servers, and replication gateway servers. The command removes the server instance, and not the software.

#### A.2.18.1 Synopsis

`uninstall` [*options*]

#### A.2.18.2 Description

The `uninstall` command is used to uninstall a server instance. It can be run in one of the following modes:

- **Graphical-user interface (GUI) mode**. GUI mode is the default and recommended uninstallation option. The `uninstall` GUI provides an easy interface for removing instance files.
- **Command-line interface (CLI) mode**. The command-line mode is either interactive or non-interactive. The interactive CLI mode prompts you for any required information before the uninstallation begins, and is used with the `--cli` option, or if no GUI is available.

The non-interactive CLI mode enables you to uninstall the instance files without user intervention. Use the `--no-prompt` and the `--quiet` options to suppress interactivity and output information, respectively.

Whether running in GUI mode or in command-line mode, uninstall lists the components that you can remove. If uninstall cannot remove all of the instance files, it displays a message that lists any directories that are still present.

Depending on the type of server installed, you are presented with different uninstall options. These are broadly categorized into the following:

- Section A.2.18.3, "Removing a Directory Server"
- Section A.2.18.4, "Removing a Proxy Server"
- Section A.2.18.5, "Removing a Replication Gateway Server"

**Note:** For any instance (directory server, proxy, or replication gateway) type that you decide to remove, the uninstall procedure also stops the server. In addition, for a server instance that is part of a replication topology, the uninstall procedure removes the server that is under deletion from that topology. On a Windows platform, if the instance was installed as a windows service, the windows service is unregistered.

### A.2.18.3 Removing a Directory Server

This section describes the options to remove a directory server instance.

#### A.2.18.3.1 Options

The `uninstall` command accepts an option in either its short form (for example, `-i`) or its long form equivalent (for example, `--cli`).

`-i, --cli`
Use the command line install. If not specified the graphical interface will be launched. The rest of the options (excluding help and version) will only be taken into account if this option is specified.

`-a, --remove-all`
Remove all components of the server (this option is not compatible with the rest of the remove options).

`-l, --server-libraries`
Remove server libraries and administrative tools.

`-d, --databases`
Remove all database content.

`-L, --log-files`
Remove all log files.

`-c, --configuration-files`
Remove configuration files.

`-b, --backup-files`
Remove all backup files.

`-e, --ldif-files`
Remove LDIF files.

`-f, --forceOnError`
Specifies whether the uninstall should continue if there is an error updating references to this server in remote server instances or not. This argument can only be used with the `--no-prompt` argument.

#### A.2.18.3.2 LDAP Connection Options

`-I, --adminUID` ***user-ID***
Specify the user ID of the global administrator to bind to the server.

`-j, --bindPasswordFile` ***filename***
Use the bind password in the specified file when authenticating to the directory server.

`-o, --saslOption` ***name*=*value***
Use the specified options for SASL authentication.

`-X, --trustAll`
Trust any certificate that the server presents. This option can be used for testing purposes, but for security reasons, a trust store should be used to determine whether the client should accept the server certificate.

`-P, --trustStorePath` ***path***
Use the client trust store certificate in the specified path. This option is not needed if `--trustAll` is used, although a trust store should be used when working in a production environment.

`-U, --trustStorePasswordFile` ***filename***
Use the password in the specified file to access the certificates in the client trust store. This option is only required if `--trustStorePath` is used and the specified trust store requires a password to access its contents (most trust stores do not require this).

`-K, --keyStorePath` ***path***
Use the client keystore certificate in the specified path.

`-u, --keyStorePasswordFile` ***filename***
Use the password in the specified file to access the certificates in the client keystore. This option is only required if `--keyStorePath` is used.

`-N, --certNickname` ***nickname***
Use the certificate for SSL client authentication.

`--connectTimeout` ***timeout***
Maximum length of time that can be taken to establish a connect in milliseconds. Use `0` to specify no timeout. The default value is 30000.

`-h, --referencedHostName` ***host***
Specify the name of this host (or IP address) as it is referenced in remote servers for replication.

### A.2.18.4 Removing a Proxy Server

This section describes the options to remove a proxy server instance.

#### A.2.18.4.1 Options

The `uninstall` command accepts an option in either its short form (for example, `-i`) or its long form equivalent (for example, `--cli`).

`-i, --cli`
Use the command line install. If not specified the graphical interface will be launched. The rest of the options (excluding help and version) will only be taken into account if this option is specified.

`-a, --remove-all`
Remove all components of the server (this option is not compatible with the rest of the remove options).

`-l, --server-libraries`
Remove server libraries and administrative tools.

`-L, --log-files`
Remove all log files.

`-c, --configuration-files`
Remove configuration files.

`-b, --backup-files`
Remove all backup files.

`-e, --ldif-files`
Remove LDIF files.

`-f, --forceOnError`
Specifies whether the uninstall should continue if there is an error updating references to this server in remote server instances or not. This argument can only be used with the `--no-prompt` argument.

#### A.2.18.4.2 LDAP Connection Options

`-I, --adminUID` ***user-ID***
Specify the user ID of the global administrator to bind to the server.

`-j, --bindPasswordFile` ***filename***
Use the bind password in the specified file when authenticating to the directory server.

`-o, --saslOption` ***name*=*value***
Use the specified options for SASL authentication.

`-X, --trustAll`
Trust any certificate that the server presents. This option can be used for testing purposes, but for security reasons, a trust store should be used to determine whether the client should accept the server certificate.

`-P, --trustStorePath` ***path***
Use the client trust store certificate in the specified path. This option is not needed if `--trustAll` is used, although a trust store should be used when working in a production environment.

`-U, --trustStorePasswordFile` ***filename***
Use the password in the specified file to access the certificates in the client trust store. This option is only required if `--trustStorePath` is used and the specified trust store requires a password to access its contents (most trust stores do not require this).

`-K, --keyStorePath` ***path***
Use the client keystore certificate in the specified path.

`-u, --keyStorePasswordFile` ***filename***
Use the password in the specified file to access the certificates in the client keystore. This option is only required if `--keyStorePath` is used.

`-N, --certNickname` ***nickname***
Use the certificate for SSL client authentication.

`--connectTimeout` ***timeout***
Maximum length of time that can be taken to establish a connect in milliseconds. Use `0` to specify no timeout. The default value is 30000.

`-h, --referencedHostName` ***host***
Specify the name of this host (or IP address) as it is referenced in remote servers for replication.

### A.2.18.5 Removing a Replication Gateway Server

This section describes the options for removing an instance of the replication gateway server.

#### A.2.18.5.1 Options

The `uninstall` command accepts an option in either its short form (for example, `-i`) or its long form equivalent (for example, `--cli`).

`-i, --cli`
Use the command line install. If not specified the graphical interface will be launched. The rest of the options (excluding help and version) will only be taken into account if this option is specified.

`-f, --forceOnError`
Specifies whether the uninstall should continue if there is an error updating references to this server in remote server instances or not. This argument can only be used with the `--no-prompt` argument.

#### A.2.18.5.2 Gateway Connection Options

`-h, --hostname` ***hostname***
The fully-qualified name of the host where the replication gateway is installed. This name must be the one provided during the setup of the replication gateway.

#### A.2.18.5.3 Oracle Unified Directory Server Connection Options

`-I, --adminUID` ***adminUID***
User ID of the Global Administrator to use to bind to the Oracle Unified Directory server. If no Global Administrator was defined previously in the new generation server, then provide a Bind DN. The default value is `admin`.

`--adminPasswordFile` ***bindPasswordFile***
File containing the password of the Global Administrator (or of the bind DN) to use to bind to the Oracle Unified Directory server.

#### A.2.18.5.4 Oracle Directory Server Enterprise Edition Server Connection Options

`--bindDNLegacy` ***bindDN***
Specifies the DN that is used to bind theOracle Directory Server Enterprise Edition server whose contents whose contents are replicated through the replication gateway. The default value is `cn=Directory Manager`.

`--bindPasswordFileLegacy` ***bindPasswordFile***
Specifies the file that stores the password that is used to bind theOracle Directory Server Enterprise Edition server whose contents are replicated through the replication gateway.

#### A.2.18.5.5 Secure Connection Options

`-o, --saslOption` ***name*=*value***
These are SASL bind options.

SASL is not supported for a proxy server instance.

`-X, --trustAll`
Trust all server SSL certificates that the server presents. This option can be used for convenience and testing purposes, but for security reasons a trust store should be used to determine whether the client should accept the server certificate.

`-P, --trustStorePath` ***path***
Use the trust store certificate in the specified path. This option is not needed if `--trustAll` is used, although a trust store should be used when working in a production environment.

`-U, --trustStorePasswordFile` ***path***
Use the password in the specified file to access the certificates in the trust store. This option is only required if `--trustStorePath` is used and the specified trust store requires a password in order to access its contents (most trust stores do not require this).

`-K, --keyStorePath` ***path***
Use the keystore certificate in the specified path.

`-u, --keyStorePasswordFile` ***filename***
Use the password in the specified file to access the certificates in the keystore. This option is only required if `--keyStorePath` is used.

`-N, --certNickname` ***nickname***
Use the specified certificate for SSL client authentication.

`--connectTimeout` ***timeout***
Specifies the maximum length of time (in milliseconds) that can be taken to establish a connection. Use `0`to specify no time out. The default value is 30000.

### A.2.18.6 Command Input/Output Options

`-n, --no-prompt`
Run `setup` in non-interactive mode. If some data in the command is missing, the user will not be prompted and the command will fail.

`-Q, --quiet`
Run in quiet mode. No output will be generated unless a significant error occurs during the process.

`-v, --verbose`
Run in verbose mode, displaying diagnostics on standard output.

`--noPropertiesFile`
Indicate that the command will not use a properties file to get the default command-line options.

`--propertiesFilePath` ***path***
Specify the path to the properties file that contains the default command-line options.

### A.2.18.7 General Options

`-?, -H, --help`
Display command-line usage information for the command and exit without making any attempt to stop or restart the server.

`--version`
Display the version information for the directory server and exit rather than attempting to run this command.

### A.2.18.8 Examples

The following examples show how to use the server commands.

#### *Example A–54 Uninstalling by Using the Graphical Uninstaller*

The following command opens the Uninstaller GUI and prompts you to select the components that must be deleted:

```
$ uninstall
```

#### *Example A–55 Uninstalling by Using the Command Line*

The following command prompts you to indicate whether all components, or specific components, should be removed, and then runs the `uninstall` command. If the server is running, you are prompted to stop the server before continuing.

```
$ uninstall --cli
```

#### *Example A–56 Uninstalling in Non-Interactive CLI Mode*

This mode enables you to create an uninstallation script with the `uninstall` command. It requires the `--no-prompt` (`-n`) and `--quiet` (`-Q`) options to be provided. If no option is present, the `uninstall` command defaults to interactive mode. Both, `-n` and `-Q` options work in the CLI mode only.

The following command uninstalls all instance components in non-interactive CLI mode.

```
$ uninstall --cli -a -n -Q
```

### A.2.18.9 Exit Codes

The following exit codes are applicable for a directory server and a proxy server:

`0`
Successful.

`1`
User cancelled the operation.

`2`
User provided invalid data.

`3`
Error accessing file system (reading/writing).

`5`
Error during the configuration of the Directory Server.

`7`
Error starting the Oracle Unified Directory server.

`8`
Error stopping the Oracle Unified Directory server.

`9`
Error disabling the Windows service.

`10`
Application specific error.

`11`
Error invoking an Oracle Unified Directory tool.

`12`
Bug.

`13`
Java version non-compatible.

`14`
User provided invalid input.

`50`
Print Version.

`51`
Print Usage.

`100`
Return code for errors that are non-specified.

The following exit codes are applicable for a gateway server:

`0`
Successful uninstall.

`1`
Unexpected error (potential bug).

2
Cannot parse arguments or data provided by user is not valid.

3
The user canceled the uninstall.

4
Incompatible Java version.

5
Error initializing the replication gateway configuration (loading the admin framework classes, and so on).

6
Error stopping the replication gateway.

7
Error unconfiguring windows service.

8
Error input limit.

9
Error updating ADS Contents.

10
An error with the configuration of the legacy server. The base DN specified in the replica configuration is not a valid DN.

11
One of the specified legacy (Oracle Directory Server Enterprise Edition) servers is not compatible.

12
One of the specified new generation (Oracle Unified Directory based) servers is not compatible.

13
The user does not accept the certificate.

14
The user does not want to continue because there were issues loading the configuration of some servers.

15
An error with the configuration of the replication gateway.

16
The user overcame the maximum number of tries in interactive mode.

17
The user aborted the uninstall.

18
Error accessing file system (for instance deleting installation files).

### A.2.18.10 Using a Properties File

The directory server supports the use of a *properties file* that passes in any default option values used with the `uninstall` command. The properties file is convenient when working in different configuration environments, especially in scripted or embedded applications. For more information, see Section A.1.2, "Using a Properties File With Server Commands."

The following options can be stored in a properties file:

- `adminUID`
- `bindPasswordFile`
- `certNickname`
- `hostname`
- `keyStorePasswordFile`
- `keyStorePath`
- `saslOption`

  SASL is not supported for Oracle Unified Directory.
- `trustAll`
- `trustStorePasswordFile`
- `trustStorePath`

Entries in the properties file have the following format:

```
toolname.propertyname=propertyvalue
```

For example:

```
uninstall.bindPasswordFile=/path/pwd-file
```

### A.2.18.11 Log Files

The `uninstall` command writes a log file named `oud-uninstall-`*IDnumber*, where `IDnumber` is a decimal number. The log files are located at these paths:

- UNIX (Solaris): `/var/tmp/`
- Linux: `/tmp/`
- Windows: The `%TEMP%` folder. By default, this folder is `C:\Documents and Settings\user\Local Settings\Temp`.

### A.2.18.12 Location

The `uninstall` command is located at these paths:

- UNIX and Linux: *INSTANCE_DIR*`/OUD/uninstall`
- Windows: *INSTANCE_DIR*`\OUD\uninstall.bat`

### A.2.18.13 Related Commands

- Section A.2.12, "oud-replication-gateway-setup"
- Section A.2.13, "oud-setup"

## A.2.19 windows-service

The `windows-service` command manually enables or disables the server as a Windows service.

### A.2.19.1 Synopsis

`windows-service` [*options*]

### A.2.19.2 Description

The `windows-service` command can be used to manually enable (or disable) the server as a Windows service. Windows services are applications similar to UNIX daemons that run in the background and are not in direct control by the user.

### A.2.19.3 Command Options

The `windows-service` command accepts an option in either its short form (for example, `-d`) or its long form equivalent (for example, `--disableService`):

`-c,--cleanupService` ***service-name***
Disable the service and clean up the Windows registry information associated with the provided service name.

`-d, --disableService`
Disable server as a Windows service.

`-e, --enableService`
Enable server as a Windows service.

`-s, --serviceState`
Display the state of the server as a Windows service.

### A.2.19.4 General Options

`-?, -H, --help`
Display command-line usage information for the command and exit without making any attempt to stop or restart the server.

`-V, --version`
Display the version information for the server and exit rather than attempting to run this command.

### A.2.19.5 Examples

The following examples show how to use the `windows-service` command.

***Example A–57 Enabling the Server as a Windows Service***

The following command enables the server as a Windows service:

```
$ windows-service -e
```

***Example A–58 Disabling the Server as a Windows Service***

The following command disables the server as a Windows service:

```
$ windows-service -d
```

***Example A–59 Displaying a Status***

The following command displays a status of the server as a Windows service:

```
$ windows-service -s
```

#### A.2.19.6 Exit Codes

`0`
Server started/stopped successfully.

`1`
Service not found.

`2`
Server start error. Server already stopped

`3`
Server stop error.

#### A.2.19.7 Location

*INSTANCE_DIR*`\OUD\bat\windows-service.bat`

#### A.2.19.8 Related Commands

Section A.2.13, "oud-setup"

Section A.2.14, "oud-proxy-setup"

Section A.2.12, "oud-replication-gateway-setup"

## A.3 Data Administration Commands

The following sections describe the data administration commands:

- Section A.3.1, "backup"
- Section A.3.2, "base64"
- Section A.3.3, "dbtest"
- Section A.3.4, "encode-password"
- Section A.3.5, "export-ldif"
- Section A.3.6, "import-ldif"
- Section A.3.7, "ldif-diff"
- Section A.3.8, "ldifmodify"
- Section A.3.9, "ldifsearch"
- Section A.3.10, "list-backends"
- Section A.3.11, "make-ldif"
- Section A.3.12, "manage-account"
- Section A.3.13, "rebuild-index"
- Section A.3.14, "restore"
- Section A.3.15, "split-ldif"

- Section A.3.16, "verify-index"

## A.3.1 backup

The `backup` command archives the contents of one or more directory server back ends.

### A.3.1.1 Synopsis

`backup` [*options*]

### A.3.1.2 Description

The `backup` command archives the contents of one or more directory server back ends. The command can perform this operation immediately or at a scheduled time. For more information, see Section 14.4, "Configuring Commands As Tasks."

The `backup` command can be run when the server is online or offline. If the backup is run while the server is online, the command contacts the server over SSL, through the administration connector, and registers a backup task. For more information about the administration connector, see Section 14.3, "Managing Administration Traffic to the Server."

### A.3.1.3 Options

The `backup` command accepts an option in either its short form (for example, `-B` *backupID*) or its long form equivalent (for example, `--incrementalBaseID` *backupID*).

`-a, --backUpAll`
Back up all configured back ends. This option must not be used in conjunction with `--backendID`.

`-A, --hash`
Generate a hash, or message digest, of the contents of the backup archive. The hash can be used as a checksum during the restore process to ensure that the backup has not been altered.

`-B, --incrementalBaseID` ***backupID***
Specify the backup ID for the existing backup against which to take an incremental backup. If this ID is not provided, the incremental backup is based on the latest incremental or full backup contained in the backup directory.

`-c, --compress`
Compress the contents of the backup archive. The compression algorithm used may vary based on the back end type.

`-d, --backupDirectory` ***path***
Write the backup files to the specified directory. If multiple back ends are archived, a subdirectory is created below this path for each back end. Otherwise, the backup files are placed directly in this directory. Note that multiple backups for the same back end can be placed in the same directory. If an incremental backup is to be performed, the backup directory must already contain at least one full backup. This is a required option.

For an online backup, the root for relative paths is the instance directory, and not the current working directory. For example, if you specify `-d bknov2011`, the backup files will be placed in `instance-dir/bknov2011`.

`-i, --incremental`
Perform an incremental backup rather than a full backup. An incremental backup includes only the data that has changed since a previous incremental or full backup. Thus, running an incremental backup can be notably faster than a full backup. When restoring an incremental backup, it is first necessary to restore the original full backup and then any intermediate incremental backups, which can make the restore process somewhat slower than restoring just a full backup. Note that some types of back ends might not support performing incremental backups. In this case, this option is ignored and a full backup is performed.

`-I, --backupID` ***backupID***
Specify an identifier to use for the backup. If this is not provided, a backup ID is generated, based on the current time. The backup ID must be unique among all backups in the provided backup directory.

`-n, --backendID` ***backendID***
Specify the ID of the back-end to be saved. This option can be used multiple times in a single command to indicate that multiple back ends should be backed up. The available back ends in the server can be determined by using the `dsconfig list-backends` command.

`-s, --signHash`
Generate a signed hash. This provides even stronger assurance that neither the backup archive nor the hash of its contents have been altered. This option can only be used if a connection to an online directory server instance is present. In this case, you must specify the `--hostname`, `--port`, `--bindDN`, and `--bindPasswordFile` options of the online directory server that will generate a signed hash of the archive.

`-y, --encrypt`
Encrypt the contents of the backup archive. This option can only be used if a connection to an online server instance is present. In this case, you must specify the `--hostname`, `--port`, `--bindDN`, and `--bindPasswordFile` options of the online directory server that will encrypt the archive.

### A.3.1.4 Task Back End Connection Options

Running an online backup requires access to the tasks back end. Access to the tasks back end is provided over SSL through the administration connector. These connection options are used when the backup runs online.

`-D, --bindDN` ***bindDN***
Use the bind DN to authenticate to the directory server. This option is used when performing simple authentication and is not required if SASL authentication is to be used. The default value for this option is `cn=Directory Manager`.

`-h, --hostname` ***hostname***
Contact the directory server on the specified hostname or IP address. If this option is not provided, a default of `localhost` is used.

`-j, --bindPasswordFile` ***filename***
Use the bind password in the specified file when authenticating to the directory server.

`-K, --keyStorePath` ***path***
Use the client keystore certificate in the specified path.

`-N, --certNickname` ***nickname***
Use the specified certificate for client authentication.

`-o, --saslOption` ***name*=*value***
Use the specified options for SASL authentication.

`-p, --port` ***port***
Contact the directory server at the specified administration port. If this option is not provided, a default administration port of `4444` is used.

`-P, --trustStorePath` ***path***
Use the client trust store certificate in the specified path. This option is not needed if `--trustAll` is used, although a trust store should be used when working in a production environment.

`-u, --keyStorePasswordFile` ***filename***
Use the password in the specified file to access the certificates in the client keystore. This option is only required if `--keyStorePath` is used.

`-U, --trustStorePasswordFile` ***filename***
Use the password in the specified file to access the certificates in the client trust store. This option is only required if `--trustStorePath` is used and the specified trust store requires a password in order to access its contents (most trust stores do not require this).

`-X, --trustAll`
Trust all server SSL certificates that the directory server presents. This option can be used for convenience and testing purposes, but for security reasons a trust store should be used to determine whether the client should accept the server certificate.

### A.3.1.5 Task Scheduling Options

These options are used when you specify that the backup should run as a scheduled task.

`--completionNotify` ***emailAddress***
Specify the email address of a recipient to be notified when the task completes. This option can be specified more than once in a single command.

`--dependency` ***taskId***
Specify the ID of a task upon which this task depends. A task does not start executing until all of its dependencies have completed execution.

`--errorNotify` ***emailAddress***
Specify the email address of a recipient to be notified if an error occurs when this task executes. This option can be specified more than once in a single command.

`--failedDependencyAction` ***action***
Specify the action that this task will take if one of its dependent tasks fails. The value must be one of `PROCESS`, `CANCEL`, or `DISABLE`. If no value is specified, the default action is `CANCEL`.

`--recurringTask` ***schedulePattern***
Indicates that the task is recurring and will be scheduled according to the `schedulePattern`, expressed as a crontab(5) compatible time and date pattern.

`-t, --start` ***startTime***
Indicates the date and time at which the operation starts when scheduled as a directory server task expressed in the format `YYYYMMDDhhmmss`. A value of 0 schedules the task for immediate execution. When this option is specified, the operation is scheduled to start at the specified time after which the command exits immediately.

### A.3.1.6 Command Input/Output Options

`--noPropertiesFile`
Indicates that a properties file is not used to obtain the default command-line options.

`--propertiesFilePath` ***path***
Specify the path to the properties file that contains the default command-line options.

### A.3.1.7 General Options

`-?, -H, --help`
Display command-line usage information for the command and exit without making any attempt to back up data.

`-V, --version`
Display the version information for the directory server and exit rather than attempting to run this command.

### A.3.1.8 Examples

The following examples show how to use the directory server commands.

#### *Example A–60 Backing Up All Configured Back Ends*

The following command archives all directory server back ends (`-a`), compresses them (`-c`), and saves them to a specified directory (`-d`).

```
$ backup -a -c -d /tmp/backup
```

Display the contents of the backup directory, to see the subdirectories for each back end:

```
$ ls /tmp/backup
config    schema    tasks     userRoot
```

Display the contents of a subdirectory, to see that the system assigned a backup ID based on the current time.

```
$ ls /tmp/backup/userRoot/
backup-userRoot-20081015151640Z  backup.info
```

You can assign your own unique backup ID by using the `-I` option. For example:

```
$ backup -a -c -d /tmp/backup -I October08
```

Display the contents of the `userRoot` subdirectory to see the assigned backup ID.

```
$ ls /tmp/backup/userRoot/
backup-userRoot-October08        backup.info
```

#### *Example A–61 Backing Up a Specific Back End*

Use the `-n` option to specify a back end to be backed up. The following command archives the `userRoot` back end only.

```
$ backup -n userRoot -d /tmp/backup
```

#### *Example A–62 Running an Incremental Backup*

The following command archives all directory server back ends (`-a`), using incremental backup (`-i`), compresses them (`-c`), and saves the data to a directory (`-d`).

```
$ backup -a -i -c -d /tmp/backup
```

#### *Example A–63 Running an Incremental Backup on a Specific Back End*

Use the `list-backends` command to display the current configured back ends.

```
$ list-backends
Backend ID     : Base DN
---------------:--------------------
adminRoot      : cn=admin data
ads-truststore : cn=ads-truststore
backup         : cn=backups
config         : cn=config
monitor        : cn=monitor
schema         : cn=schema
tasks          : cn=tasks
userRoot       : "dc=example,dc=com"
```

The following command runs an incremental backup (`-i`) on the `userRoot` back end (`-n`), compresses the backup (`-c`), and saves the data to a directory (`-d`).

```
$ backup -i -n userRoot -c -d /tmp/backup/userRoot
```

#### *Example A–64 Running an Incremental Backup Against an Existing Backup*

Assume that you have created two archived incremental backup files by using the `-I` or `--backupID` option and assigned the IDs `1234` and `4898` to the two files, respectively:

```
/tmp/backup/userRoot> ls
./     backup-userRoot-1234   backup.info
../    backup-userRoot-4898   backup.info.save
```

The following command runs an incremental backup (`-i`) on all configured back ends (`-a`) based on the backup ID `1234` (`-B`), assigns a backup ID of `5438` to the incremental backup, and saves the data to a directory (`-d`).

```
$ backup -a -i -B 1234 -I 5438 -d /tmp/backup
```

The contents of `backup.info` show that the latest incremental backup (`backup_id=5438`) has a dependency on `backup_id=1234`:

```
$ backend_dn=ds-cfg-backend-id=userRoot,cn=Backends,cn=config

backup_id=4898
backup_date=20070727202906Z
incremental=false
compressed=false
```

```
encrypted=false
signed_hash=VmBG/VkfMAMMPnR6M8b5kZil7FQ=
property.last_logfile_name=00000000.jdb
property.archive_file=backup-userRoot-4898
property.cipher_algorithm=AES/CBC/PKCS5Padding
property.mac_algorithm=HmacSHA1
property.last_logfile_size=490554

backup_id=1234
backup_date=20070727202934Z
incremental=false
compressed=false
encrypted=false
signed_hash=VmBG/VkfMAMMPnR6M8b5kZil7FQ=
property.last_logfile_name=00000000.jdb
property.archive_file=backup-userRoot-1234
property.cipher_algorithm=AES/CBC/PKCS5Padding
property.mac_algorithm=HmacSHA1
property.last_logfile_size=490554

backup_id=5438
backup_date=20070727203107Z
incremental=true
compressed=false
encrypted=false
dependency=1234
property.last_logfile_name=00000000.jdb
property.archive_file=backup-userRoot-5438
property.last_logfile_size=490554
```

#### *Example A–65 Backing Up All Configured Back Ends with Encryption and Signed Hash*

The directory server provides support for backup encryption (using `--encrypt`), hash generation (using `--hash`), and signed hash (using `--signHash`) to secure archived data. These options require a connection to an online server instance, over SSL through the administration connector. When you use these options, you must therefore specify the connection details, including the host, administration port, bind DN and bind password file. You must also specify the certificate details for the SSL connection.

The following command archives all directory server back ends (`-a`), compresses them (`-c`), generates a hash (`-A`), signs the hash (`-s`), encrypts the data while archiving the data (`-y`), assigns a back end ID of `123`, and saves the data to a directory (`-d`). The self signed certificate is trusted using the `-X` (`--trustAll`) option.

```
$ backup -h localhost -D "cn=Directory Manager" -j /path/pwd-file -p 4444 -X \
  -a -c -A -s -y -I 123 -d /tmp/backup
Backup task 2008101609295810 scheduled to start immediately
...
```

#### *Example A–66 Scheduling a Backup*

Scheduling a backup requires online access to the tasks back end. Access to this back end is provided over SSL through the administration connector. When you schedule a backup, you must therefore specify the connection details, including the host, administration port, bind DN and bind password file. You must also specify the certificate details for the SSL connection.

The following command schedules a backup of all components (`-a`) and writes it to the `/tmp/backups` directory (`-d`). The start time is specified with the `--start` option. The backup sends a completion notification and error notification to `admin@example.com`. The self signed certificate is trusted using the `-X` (`--trustAll`) option.

```
$ backup -h localhost -D "cn=Directory Manager" -j /path/pwd-file -p 4444 -X \
  -a -d /tmp/backups --start 20090124121500 --completionNotify admin@example.com \
  --errorNotify admin@example.com
Backup task 2007102914530410 scheduled to start Jan 24, 2009 12:15:00 PM SAST
```

You can view this scheduled task by using the `manage-tasks` command. For more information, see Section 14.4, "Configuring Commands As Tasks."

#### A.3.1.9 Exit Codes

An exit code of 0 indicates that the operation completed successfully. An exit code of 1 indicates that an error occurred during processing.

#### A.3.1.10 Using a Properties File

The directory server supports the use of a *properties file* that passes in any default option values used with the `backup` command. The properties file is convenient when working in different configuration environments, especially in scripted or embedded applications. For more information, see Section A.1.2, "Using a Properties File With Server Commands."

#### A.3.1.11 Location

The `backup` command is located at these paths:

- UNIX and Linux: *INSTANCE_DIR*`/OUD/bin/backup`
- Windows: *INSTANCE_DIR*`\OUD\bat\backup.bat`

#### A.3.1.12 Related Commands

- Section A.3.14, "restore"
- Section A.3.10, "list-backends"
- Section A.2.8, "manage-tasks"

### A.3.2 base64

The `base64` command encodes binary strings using the base64 encoding format.

#### A.3.2.1 Synopsis

`base64` *subcommand*[*options*]

#### A.3.2.2 Description

The `base64` command encodes binary strings into text representations using the base64 encoding format. Base64 encoding is often used in LDIF files to represent non-ASCII character strings. It is also frequently used to encode certificate contents or the output of message digests such as MD5 or SHA.

#### A.3.2.3 Subcommands

The following subcommands are used with the `base64` command.

`decode`
Decodes base64-encoded information into raw data. Suboptions are as follows:

`-d, --encodedData` *encoded-data*. Base64-encoded data to be decoded to raw data.

`-f, --encodedDataFile` *filename*. Path to the file that contains the base64-encoded data to be decoded.

`-o, --toRawFile` *filename*. Path to the file to which the raw data should be written.

`encode`
Encodes raw data to base64. Suboptions are as follows:

`-d, --rawData` *raw-data*. Raw data to be base64-encoded.

`-f, --rawDataFile` *filename*. Path to the file that contains the raw data to be base64-encoded.

`-o, --toEncodedFile` *filename*. Path to the file to which the base64-encoded data should be written.

### A.3.2.4 Global Options

`-?, -H, --help`
Display usage information.

`-V, --version`
Display directory server version information.

### A.3.2.5 Examples

The following examples show how to use the directory server commands.

***Example A–67 Base64 Encoding a String***

The following command base64-encodes the string `opends`.

```
$ base64 encode -d opends
b3BlbmRz
```

***Example A–68 Base64 Encoding the Contents of a File***

The following command base64-encodes the file (`-f`) and writes to an output file (`-o`).

```
$ base64 encode -f myrawdata -o myencodeddata
```

***Example A–69 Decoding a Base64-Encoded String***

The following command decodes a base64-encoded string.

```
$ base64 decode -d b3BlbmRz
opends
```

***Example A–70 Decoding the Contents of a Base64-Encoded File***

The following command decodes the file base64-encoded file (`-f`) and writes to an output file (`-o`).

```
$ base64 encode -f myencodeddata -o myoutput
```

***Example A–71 Base64-Encoding and Decoding on Linux Systems***

The following command encodes and decodes on Linux from the command-line. After you enter the clear-text string, press `Control-D` to signal the end of input on the command line.

```
$ base64 encode
hello world
<CTRL-D>
aGVsbGBqd29ybGQK

$ base64 decode
aGVsbG8gd29ybGQK
<CTRL-D>
hello world
```

#### A.3.2.6 Exit Codes

An exit code of 0 indicates that the operation completed successfully. An exit code of 1 indicates that an error occurred during processing.

#### A.3.2.7 Location

- UNIX and Linux: *INSTANCE_DIR*`/OUD/bin/base64`
- Windows: *INSTANCE_DIR*`\OUD\bat\base64.bat`

### A.3.3 dbtest

The `dbtest` command debugs an Oracle Berkeley Java Edition (JE) back end.

#### A.3.3.1 Synopsis

`dbtest` *subcommands* [*options*]

#### A.3.3.2 Description

The `dbtest` command is used to debug an Oracle Berkeley Java Edition (JE) back end. The command lists the root, entry, database containers, and the status of indexes in the database. The command also provides a dump of the database for debugging purposes.

A *back end* is a repository for storing data on a directory server. The back end uses some type of database (DB) to store data and to maintain a set of indexes that allow the back end to locate the entries in the directory. The primary database for the directory server is the Berkeley Java Edition (JE) database, which organizes its data as a single collection of keyed records in B-tree form.

You can use the `dbtest` command to access the following information:

- **Root container**. Specifies the back end ID and the directory for the back end.
- **Entry container**. Specifies the base DN that the entry container stores on disk, the database prefix to use for the database names, and the number of entries in the database. Each base DN of a JE back end is given its own entry container.
- **Database container**. Specifies the database name, type, and JE database name for the specific back end ID.
- **Index Status**. Specifies the index name, type, status and associated JE database.

Currently, the dbtest command is a read-only command and cannot alter the database. The command can run in online or offline mode. However, running dbtest in online mode can take considerably longer than running it in offline mode.

### A.3.3.3 Subcommands

dump-database-container
Dump records from the database container. Suboptions are as follows:

-b, --baseDN baseDN. Base DN of the entry container to debug. Required.

-d, --databaseName databaseName. The name of the database container to debug. Required.

-k, --minKeyValue value. Only show records with keys that should be ordered after the provided value using the comparator for the database container.

-K, --maxKeyValue value. Only show records with keys that should be ordered before the provided value using the comparator for the database container.

-n, --backendID backendID. ID of the local DB back end to debug. Required.

-p, --skipDecode. Skip decoding the local database to its appropriate types.

-q, --statsOnly. Display the statistics only, rather than the complete data.

-s, --minDataSize size. Only show records whose data is no smaller than the provided value.

-S, --maxDataSize size. Only show records whose data is no larger than the provided value.

list-database-containers
List the database containers for the entry container. Suboptions are as follows:

-b, --baseDN baseDN. Base DN of the entry container to debug. Required.

-n, --backendID backendID. ID of the local DB back end to debug. Required.

list-entry-containers
List the entry containers for a root container. Suboptions are as follows:

-n, --backendID backendID. ID of the local DB back end to debug. Required.

list-index-status
List the status of indexes in an entry container. Suboptions are as follows:

-b, --baseDN *baseDN*. Base DN of the entry container to debug. Required.

-n, --backendID *backendID*. ID of the local DB back end to debug. Required.

list-root-containers
List the root containers used by all local DB back ends.

### A.3.3.4 Global Options

The dbtest command accepts an option in either its short form (for example, -H) or its long form equivalent (for example, --help).

-?, -H, --help
Display the usage information.

`-V, --version`
Display directory server version information.

### A.3.3.5 Examples

The following examples show how to use the directory server commands.

***Example A–72 Displaying the List of Root Containers***

The following command lists the root containers used by all local DB back ends:

```
$ dbtest list-root-containers
Backend ID  Database Directory
------------------------------
userRoot    db

Total: 1
```

***Example A–73 Displaying a List of Entry Containers***

The following command displays the list of entry containers on the local DB back end:

```
$ dbtest list-entry-containers -n userRoot
Base DN            JE Database Prefix  Entry Count
-------------------------------------------------
dc=example,dc=com  dc_example_dc_com   102

Total: 1
```

***Example A–74 Displaying a List of Database Containers***

The following command displays the list of database containers on the local DB back end:

```
$ dbtest list-database-containers -b dc=example,dc=com -n userRoot
Database Name              Database  JE Database Name                       Entry
Count
                           Type
-------------------------------------------------------------------------------
-----
dn2id                      DN2ID     dc_example_dc_com_dn2id
102
id2entry                   ID2Entry  dc_example_dc_com_id2entry
102
referral                   DN2URI    dc_example_dc_com_referral
0
id2children                Index     dc_example_dc_com_id2children
2
id2subtree                 Index     dc_example_dc_com_id2subtree
2
state                      State     dc_example_dc_com_state
19
objectClass.equality       Index     dc_example_dc_com_objectClass.equality
6
givenName.equality         Index     dc_example_dc_com_givenName.equality
100
givenName.substring        Index     dc_example_dc_com_givenName.substring
396
member.equality            Index     dc_example_dc_com_member.equality
0
```

```
uid.equality               Index     dc_example_dc_com_uid.equality
100
cn.equality                Index     dc_example_dc_com_cn.equality
100
cn.substring               Index     dc_example_dc_com_cn.substring
1137
uniqueMember.equality      Index     dc_example_dc_com_uniqueMember.equality
0
telephoneNumber.equality   Index     dc_example_dc_com_telephoneNumber.equality
100
telephoneNumber.substring  Index     dc_example_dc_com_telephoneNumber.substring
956
sn.equality                Index     dc_example_dc_com_sn.equality
100
sn.substring               Index     dc_example_dc_com_sn.substring
541
ds-sync-hist.ordering      Index     dc_example_dc_com_ds-sync-hist.ordering
0
mail.equality              Index     dc_example_dc_com_mail.equality
100
mail.substring             Index     dc_example_dc_com_mail.substring
525
entryUUID.equality         Index     dc_example_dc_com_entryUUID.equality
102
aci.presence               Index     dc_example_dc_com_aci.presence
0

Total: 23
```

***Example A–75 Dumping the Contents of a Database and Skipping Decode***

The following command dumps the contents of a database and displays the indexed values of the entry, but skips the decode.

```
$ dbtest dump-database-container -b dc=example,dc=com -n userRoot \
  -d objectClass.equality -p

Key (6 bytes):
64 6F 6D 61 69 6E domain

Data (8 bytes):
00 00 00 00 00 00 00 01

Key (18 bytes):
67 72 6F 75 70 6F 66 75 6E 69 71 75 65 6E 61 6D groupofu niquenam
65 73 es

Data (40 bytes):
00 00 00 00 00 00 00 03 00 00 00 00 00 00 00 9C
00 00 00 00 00 00 00 9D 00 00 00 00 00 00 00 9E
00 00 00 00 00 00 00 9F
...
```

### A.3.3.6 Exit Codes

An exit code of 0 indicates that the operation completed successfully. An exit code of 1 indicates that an error occurred during processing.

#### A.3.3.7 Location

- UNIX and Linux: *INSTANCE_DIR*`/OUD/bin/dbtest`
- Windows: *INSTANCE_DIR*`\OUD\bat\dbtest.bat`

#### A.3.3.8 Related Commands

- Section A.2.4, "dsconfig"
- Section A.3.6, "import-ldif"
- Section A.3.5, "export-ldif"

### A.3.4 encode-password

The `encode-password` command encodes and compares user passwords.

This command is not supported for the proxy.

#### A.3.4.1 Synopsis

`encode-password` *options*

#### A.3.4.2 Description

The `encode-password` command can be used to interact with the password storage schemes defined in the directory server. It has three modes of operation:

- **List schemes mode**. List the password storage schemes that are available in the directory server. In this mode, only the `--listSchemes` option is required.
- **Encode clear-text mode**. Encode a clear-text password using a provided password storage scheme. In this mode, the `--storageScheme` option is required, along with a clear-text password that is read from a file (`--clearPasswordFile`).
- **Validate password mode**. Determine whether a given clear-text password is correct for a provided encoded password. In this mode, a clear-text password (from `--clearPasswordFile`) and an encoded password (from `--encodedPasswordFile)` are required.

The set of authentication passwords available for use in the directory server can be retrieved from the `supportedAuthPasswordSchemes` attribute of the root DSE entry. You can use `ldapsearch` to view this information.

#### A.3.4.3 Options

The `encode-password` command accepts an option in either its short form (for example, `-f` *filename*) or its long form equivalent (for example, `--clearPasswordFile` *filename*).

`-a, --authPasswordSyntax`
Use the Authentication Password Syntax (as defined in RFC 3112 (`http://www.ietf.org/rfc/rfc3112.txt`)), which encodes values in a form *scheme$authInfo$authValue*. If this option is not provided, then the user password syntax (which encodes values in a form *scheme$value* will be used.

`-E, --encodedPasswordFile` ***filename***
Use the encoded password from the specified file to compare against a given clear-text password. If the `--authPasswordSyntax` option is also provided, then this password must be encoded using the authentication password syntax. Otherwise, it should be encoded using the user password syntax.

`-f, --clearPasswordFile` ***filename***
Use the clear-text password from the specified file when either encoding a clear-text password or comparing a clear-text password against an encoded password.

`-i, --interactivePassword`
The password to encode or to compare against an encoded password is interactively requested from the user.

`-l, --listSchemes`
Display a list of the password storage schemes that are available for use in the directory server. If the option is used by itself, it displays the names of the password storage schemes that support the user password syntax. If the option used in conjunction with `--authPasswordSyntax`, it displays the names of the password storage schemes that support the authentication password syntax.

`-r, --useCompareResultCode`
Use an exit code that indicates whether a given clear-text password matched a provided encoded password. If this option is provided, the directory server results in an exit code of `6` (`COMPARE_TRUE`) or an exit code of `5` (`COMPARE_FALSE`). Any other exit code indicates that the command failed to complete its processing to make the necessary determination. If this option is not provided, an exit code of zero will be used to indicate that the command completed its processing successfully, or something other than zero if an error occurred.

`-s, --storageScheme` ***storageScheme***
Specify the name of the password storage scheme to use when encoding a clear-text password. If the `--authPasswordSyntax` option is provided, the value must be the name of a supported authentication password storage scheme. Otherwise, specify the name of a supported user password storage scheme.

`-?, -H, --help`
Display the command-line usage information for the command and exit immediately without taking any other action.

`-V, --version`
Display the version information for the directory server.

### A.3.4.4 Examples

The following examples show how to use the `encode-password` command.

#### *Example A–76 Listing the Storage Schemes on the Server*

The following command lists the storage schemes (`-l`) available for use on the directory server.

```
$ encode-password -l
3DES
AES
BASE64
BLOWFISH
CLEAR
CRYPT
MD5
RC4
SHA
SMD5
SSHA
SSHA256
```

```
SSHA384
SSHA512
```

#### *Example A–77 Listing the Authenticated Passcode Syntax Storage Schemes on the Server*

The following command lists the storage schemes (`-l`) that support the authentication passcode syntax (`-a`) on the directory server.

```
$ encode-password -l -a

MD5
SHA1
SHA256
SHA384
SHA512
```

#### *Example A–78 Encoding a Clear-Text Password to Another Scheme*

The following command encodes a clear-text password in a file (`-f`) using the specified scheme (`-s`).

```
$ encode-password -f /path/clear-pwd-file -s MD5

Encoded Password:  "{MD5}AjxHKRFkRwxx3j9lM2HMow=="
```

#### *Example A–79 Encoding a Clear-Text Password to Another Scheme using the Authentication Password Syntax*

The following command encodes a clear-text password in a file (`-f`) using the specified scheme (`-s`) and the authentication password syntax (`-a`).

```
$ encode-password -f /path/clear-pwd-file -s MD5 -a

Encoded Password:  "MD5$/imERhcEu3U=$AFqmpZi8EiTIvMFwkcrf8A=="
```

#### *Example A–80 Comparing a Clear-Text Password to an Encoded Password*

The following command compares a clear-text password in a file (`-f`) with an encoded password in a file (`-E`). Do not include the password scheme (for example, `MD5`) in your encoded password.

```
$ encode-password -f /path/clear-pwd-file -E /path/encoded-pwd-file -s MD5

The provided clear-text and encoded passwords match
```

#### *Example A–81 Compare a Clear-Text Password to an Encoded Password and Return an Exit Code*

The following command compares a clear-text password in a file (`-f`) with an encoded password in a file (`-E`) using the scheme (`-s`) and returns the exit code (`-r`) (`6` for COMPARETRUE; `5` for COMPAREFALSE). Do not include the password scheme (for example, `MD5`) in your encoded password.

```
$ encode-password -f /path/clear-pwd-file -E /path/encoded-pwd-file -s MD5 -r

The provided clear-text and encoded passwords match
```

```
echo $?
6
```

***Example A–82 Encoding a Password Contained in a File using SSHA***

The following command encodes a clear-text password in a file (`-f`) using the specified scheme (`-s`). For Windows platforms, specify the path to your clear-text password file (for example, `-f \temp\testpassword`):

```
$ encode-password -s SSHA -f /path/clear-pwd-file

Encoded Password:   "{SSHA}QX2fMu+2N22N9qI+zu6fIZxsBVID3EsUlYYEbQ=="
```

#### A.3.4.5 Exit Codes

***Table A–3 Exit Codes***

| Exit Code | Description |
|---|---|
| `0` | Operation completed successfully. |
| `1` | Error occurred during operation. |
| `5` | COMPARE_FALSE. Used with the `--r` or `--useCompareCodeResult` option, an exit code of `5` indicates a given clear-text password does not match the provided encoded password. |
| `6` | COMPARE_TRUE. Used with the `--r` or `--useCompareCodeResult` option, an exit code of `6` indicates that a given clear-text password matches the provided encoded password. |

#### A.3.4.6 Location

- UNIX and Linux: *INSTANCE_DIR*`/OUD/bin/encode-password`
- Windows: *INSTANCE_DIR*`\OUD\bat\encode-password.bat`

### A.3.5 export-ldif

The `export-ldif` command exports the contents of a directory server back end to LDIF format.

#### A.3.5.1 Synopsis

`export-ldif` [*options*]

#### A.3.5.2 Description

The `export-ldif` command exports the contents of a directory server back end to LDIF format. This command can run the export immediately or can be scheduled to run at a specified date and time. For more information, see Section 14.4, "Configuring Commands As Tasks."

Because some back ends cannot be imported to the directory server, the `export-ldif` command does not export the following back ends: `monitor`, `ads-truststore`, `backup`, and `config-file-handler`.

You can run the `export-ldif` command in online or offline mode.

- **Online mode**. In online mode, `export-ldif` contacts a running directory server instance over SSL, through the administration connector, and registers an export task. The command runs in online mode automatically if you specify any of the task back end connection options. For more information about the administration connector, see Section 14.3, "Managing Administration Traffic to the Server."
- **Offline mode**. In offline mode, `export-ldif` accesses the database directly rather than through a directory server instance. To perform an offline export, the directory server must be stopped.

### A.3.5.3 Options

The `export-ldif` command accepts an option in either its short form (for example, `-b` *branchDN*) or its long form equivalent (for example, `--includeBranch` *branchDN*).

`-a, --appendToLDIF`
Append the export to an existing LDIF file rather than overwriting it. If this option is not provided, the directory server overwrites the specified LDIF file, if it exists.

`-b, --includeBranch` ***branchDN***
Specify the base DN for a branch or subtree of the data to be exported. This option can be used multiple times to specify multiple base DNs. If this option is provided, entries contained in the back end that are not at or below one of the provided base DNs are skipped.

`-B, --excludeBranch` ***branchDN***
Specify the base DN for a branch or subtree of the data to be omitted from the export. This option can be used multiple times to specify multiple base DNs. If this option is provided, any entries contained in the back end that are at or below one of the provided base DNs are skipped. Note that the use of the `--excludeBranch` option takes precedence over the `--includeBranch` option. If an entry is at or below a DN contained in both the included and excluded lists, it is not included. This capability makes it possible to include data for only part of a branch. For example, you can include all entries below `dc=example,dc=com` except those below `ou=People,dc=example,dc=com`.

`-c, --compress`
Compress the LDIF data as it is written. The data is compressed using the GZIP format, which is the format used by the `--isCompressed` option of the `import-ldif` command.

`-e, --excludeAttribute` ***attribute***
Exclude the specified attribute name during the export. This option can be used multiple times to specify multiple attributes. If this option is provided, any attributes listed are omitted from the entries that are exported.

`-E, --excludeFilter` ***filter***
Exclude the entries identified by the specified search filter during the export. This option can be used multiple times to specify multiple filters. If this option is provided, any entry in the back end that matches the filter is skipped. Note that the use of the `--excludeFilter` option takes precedence over the `--includeFilter` option. If an entry matches filters in both the included and excluded lists, the entry is skipped.

`-i, --includeAttribute` ***attribute***
Include the specified attribute name in the export. This option can be used multiple times to specify multiple attributes. If this option is provided, any attributes not listed are omitted from the entries that are exported.

`-g, --algorithm` ***algorithm***
The specified algorithm used in the export. This option is optional and you can enter one the following values:

- `diskOrder`: This option enables to read data from an Oracle Berkeley DB Java Edition (JE) back end in the order with which they are stored on the disk.

  The benefits of this option are as follows:

  - It is recommended when the database does not fit entirely in the database cache.
  - It temporarily uses `20%` of the database cache to run and then releases it. The database cache memory is decreased by `20%` during an export.

  **Note:** This algorithm uses a special feature called Disk Ordered from the JE backend and may cause an error, when the server is running and if you access it for modifications at the same time as the export. You can perform read operations.

- `entryIdOrder`: This option enables to read data from an Oracle Berkeley DB Java Edition (JE) back end in the order with which they are stored logically at the database level.

  The benefits of this option are as follows:

  - This option provides better performance compare to the `diskOrder` algorithm, if the database entirely fits into the database cache.
  - It does not temporarily extract any memory from the database cache.
  - You can run this algorithm even when the server is running and if you access it for modifications at the same time as the export.

- `auto`: This option automatically selects `diskOrder` in an offline mode when the server is down or `entryIdOrder` in an online mode when the server is running.

`-I, --includeFilter` ***filter***
Include the entries identified by the specified search filter in the export. This option can be used multiple times to specify multiple filters. If this option is provided, any entry in the back end that does not match the filter is skipped.

`-l, --ldifFile` ***filename***
Export the data to the specified LDIF file. This is a required option.

For online exports, the root for relative paths is the *instance root*, rather than the current working directory. So, for example, a path of `exports/ldif.ldif` here refers to `instance-root/exports/ldif.ldif`.

`-n, --backendID` ***backendID***
Specify the back end ID of the data to be exported. The available back ends in the directory server can be determined using the `list-backends` command. This is a required option.

`-O, --excludeOperational`
Exclude operational attributes in the export.

`--wrapColumn` ***column***
Specify the column at which to wrap long lines when writing to the LDIF file. A value of `0` indicates that the data should not be wrapped.

### A.3.5.4 Task Back End Connection Options

Running an online export requires access to the tasks back end. Access to the tasks back end is provided over SSL through the administration connector. These connection options are used when the export runs online.

`-D, --bindDN` ***bindDN***
Use the bind DN to authenticate to the directory server. This option is used when performing simple authentication and is not required if SASL authentication is to be used. The default value for this option is `cn=Directory Manager`.

`-h, --hostname` ***hostname***
Contact the directory server on the specified hostname or IP address. If this option is not provided, a default of `localhost` is used.

`-j, --bindPasswordFile` ***filename***
Use the bind password in the specified file when authenticating to the directory server.

`-K, --keyStorePath` ***path***
Use the client keystore certificate in the specified path.

`-N, --certNickname` ***nickname***
Use the specified certificate for client authentication.

`-o, --saslOption` ***name***=***value***
Use the specified options for SASL authentication.

`-p, --port` ***port***
Contact the directory server at the specified administration port. If this option is not provided, a default administration port of `4444` is used.

`-P, --trustStorePath` ***path***
Use the client trust store certificate in the specified path. This option is not needed if `--trustAll` is used, although a trust store should be used when working in a production environment.

`-u, --keyStorePasswordFile` ***filename***
Use the password in the specified file to access the certificates in the client keystore. This option is only required if `--keyStorePath` is used.

`-U, --trustStorePasswordFile` ***filename***
Use the password in the specified file to access the certificates in the client trust store. This option is only required if `--trustStorePath` is used and the specified trust store requires a password in order to access its contents (most trust stores do not require this).

`-X, --trustAll`
Trust all server SSL certificates that the directory server presents. This option can be used for convenience and testing purposes, but for security reasons a trust store should be used to determine whether the client should accept the server certificate.

### A.3.5.5 Task Scheduling Options

These options are used when you specify that the export should run as a scheduled task.

`--completionNotify` ***emailAddress***
Specify the email address of a recipient to be notified when the task completes. This option can be specified more than once in a single command.

`--dependency` ***taskId***
Specify the ID of a task upon which this task depends. A task does not start executing until all of its dependencies have completed execution.

`--errorNotify` ***emailAddress***
Specify the email address of a recipient to be notified if an error occurs when this task executes. This option can be specified more than once in a single command.

`--failedDependencyAction` ***action***
Specify the action that this task will take if one of its dependent tasks fails. The value must be one of `PROCESS`, `CANCEL`, or `DISABLE`. If no value is specified, the default action is `CANCEL`.

`--recurringTask` ***schedulePattern***
Indicates that the task is recurring and will be scheduled according to the `schedulePattern`, expressed as a crontab(5) compatible time and date pattern.

`-t, --start` ***startTime***
Indicates the date and time at which the operation starts when scheduled as a directory server task expressed in the format `YYYYMMDDhhmmss`. A value of 0 schedules the task for immediate execution. When this option is specified, the operation is scheduled to start at the specified time after which the command exits immediately.

### A.3.5.6 Command Input/Output Options

`--noPropertiesFile`
Indicates that a properties file is not used to obtain the default command-line options.

`--propertiesFilePath` ***path***
Specify the path to the properties file that contains the default command-line options.

### A.3.5.7 General Options

`-?, -H, --help`
Display command-line usage information for the command and exit without making any attempt to run an export.

`-V, --version`
Display the version information for the directory server and exit rather than attempting to run this command.

### A.3.5.8 Examples

The following examples show how to use the directory server commands.

***Example A–83 Performing an Offline Export***

The following example exports the `userRoot` back end, starting at the base DN specified by the `-b` option. The command exports the data to an LDIF file specified by `-l`. The directory server must be stopped before performing an offline export.

```
$ stop-ds
$ export-ldif -b dc=example,dc=com -n userRoot -l /usr/tmp/export.ldif
[17/Oct/2008:12:24:33 +0200] category=JEB severity=NOTICE msgID=8847447
msg=Exported 102 entries and skipped 0 in 0 seconds (average rate 159.4/sec)
```

***Example A–84 Performing an Online Export***

An export is automatically run online if you specify any of the task back end connection options. Because an online export contacts the server over SSL, you must specify how to trust the SSL server certificate. This examples uses the `-X` option to trust all certificates.

```
$ export-ldif -h localhost -p 4444 -D "cn=Directory Manager" -j /path/pwd-file -X
\
  --includeBranch "dc=example,dc=com" --backendID userRoot \
  --ldifFile /usr/tmp/export.ldif
```

***Example A–85 Scheduling an Export***

You can schedule an export to run at some future date by using the `-t` or `--start` option to specify the start time. Like a regular online export, a scheduled export contacts the task back end of a running directory server and the relevant task back end connection options must be specified.

This example schedules an export of the `userRoot` back end to start on December 24.

```
$ export-ldif -h localhost -p 4444 -D "cn=Directory Manager" -j /path/pwd-file -X
\
  --includeBranch "dc=example,dc=com" --backendID userRoot \
  --ldifFile /usr/tmp/export.ldif --start 20081224121500
Export task 2008101712361910 scheduled to start Dec 24, 2008 12:15:00 PM SAST
```

You can view a scheduled task by using the `manage-tasks` command. For more information, see Section 14.4, "Configuring Commands As Tasks."

### A.3.5.9 Exit Codes

- **Offline mode**. An exit code of 0 indicates that the operation completed successfully. A non-zero exit code indicates that an error occurred during processing.
- **Online mode**. If `-t` or `--start` is specified, an exit code of 0 indicates that the task was created successfully. A nonzero exit code indicates that an error occurred when the task was created. If `-t` or `--start` is not specified, the exit codes are the same as those specified for offline mode.

### A.3.5.10 Using a Properties File

The directory server supports the use of a *properties file* that passes in any default option values used with the `export-ldif` command. The properties file is convenient when working in different configuration environments, especially in scripted or embedded applications. For more information, see Section A.1.2, "Using a Properties File With Server Commands."

#### A.3.5.11 Location

- UNIX and Linux: *INSTANCE_DIR*`/OUD/bin/export-ldif`
- Windows: *INSTANCE_DIR*`\OUD\bat\export-ldif.bat`

#### A.3.5.12 Related Commands

- Section A.3.6, "import-ldif"
- Section A.3.7, "ldif-diff"
- Section A.3.8, "ldifmodify"
- Section A.3.9, "ldifsearch"
- Section A.2.8, "manage-tasks"

### A.3.6 import-ldif

The `import-ldif` command populates an Oracle Berkeley DB Java Edition (JE) back end with data that is read from an LDIF file.

#### A.3.6.1 Synopsis

`import-ldif` *options*

#### A.3.6.2 Description

The `import-ldif` command populates an Oracle Berkeley DB Java Edition (JE) back end with data that is read from an LDIF file, or with data generated based on a MakeLDIF template. In most cases, using `import-ldif` is significantly faster than adding entries by using `ldapmodify`. Note that a complete import to an entire JE back end has better performance than a partial import to a branch of the JE back end.

The `import-ldif` command can run the import immediately or can schedule the import to run at a specified date and time. For more information, see Section 14.4, "Configuring Commands As Tasks."

You can run the `import-ldif` command in online or offline mode.

- **Online mode**. In online mode, `import-ldif` contacts a running directory server instance over SSL, through the administration connector, and registers an import task. The command runs in online mode automatically if you specify any of the task back end connection options. For more information about the administration connector, see Section 14.3, "Managing Administration Traffic to the Server."
- **Offline mode**. In offline mode, `import-ldif` accesses the database directly rather than through a directory server instance. To perform an offline import, the directory server must be stopped.

#### A.3.6.3 Options

The `import-ldif` command accepts an option in either its short form (for example, `-b` *baseDN*) or its long form equivalent (for example, `--includeBranch` *baseDN*).

`-a, --append`
Append the imported data to the data that already exists in the back end, rather than clearing the back end before starting the import.

`-A, --templateFile` ***filename***
Specify the path to a MakeLDIF template to generate the import data.

`-b, --includeBranch` ***branchDN***
Specify the base DN for a branch or subtree of the data that should be included in the import. This option can be used multiple times to specify multiple base DNs. If this option is provided, entries contained in the import source that are not at or below one of the provided base DNs are skipped. Any existing entries above the provided base DNs are preserved.

`-B, --excludeBranch` ***branchDN***
Specify the base DN branch or subtree that should be omitted from the import. This option can be used multiple times to specify multiple base DNs. If this option is provided, entries contained in the import source that are at or below one of the base DNs are skipped. Note that the use of the `--excludeBranch` option takes precedence over the `--includeBranch` option. If an entry is at or below a DN contained in both the included and excluded lists, it is omitted from the import. This capability makes it possible to include data for only a part of a branch (for example, all entries below `dc=example,dc=com` except those below `ou=People,dc=example,dc=com`).

`-c, --isCompressed`
Specify that the LDIF import file is compressed. The file should be compressed using the GZIP format, which is the format used by the `--compressLDIF` option of the `export-ldif` command.

`--countRejects`
Return the number of rejected entries during import. If the number of rejected entries is between 0 and 255, that number is returned. If the number of rejected entries is greater than 255, the command returns the value `255`. For example, if you run `import-ldif` with the `--countRejects` option and get 16 rejected entries, the command returns the value `16`. If you run `import-ldif` and get 300 rejected entries, the command returns the value `255`. Note that this option is not supported for online imports.

`-e, --excludeAttribute` ***attribute***
Specify the name of an attribute that should be excluded from the import. This option can be used multiple times to specify multiple attributes.

`-E, --excludeFilter` ***filter***
Specify the search filter to identify entries that should be excluded from the import. This option can be used multiple times to specify multiple filters. If this option is provided, any entry in the import source that matches the filter is skipped. Note that the `--excludeFilter` option takes precedence over the `--includeFilter` option. If an entry matches filters in both the include and exclude filters, the entry is skipped during import.

`-F, --clearBackend`
Confirm deletion of all existing entries for all base DNs in the specified back end when importing without the `--append` option. This only applies when importing a multiple base DN back end specified by the back end ID. This option is implied for back ends with only one base DN.

`-i, --includeAttribute` ***attribute***
Specify the attributes that should be included in the import. This option can be used multiple times to specify multiple attributes. If this option is used, attributes not listed in this set are omitted from the entries that are imported.

`-I, --includeFilter` ***filter***
Specify the search filter to identify entries that should be included in the import. This option can be used multiple times to specify multiple filters. If this option is provided, any entry in the import source that does not match the results of the filter is skipped.

`-l, --ldifFile` ***filename***
Read the LDIF file located at the specified path. This option must not be used in conjunction with `--templateFile`.

For online imports, the root for relative paths is the *instance root*, rather than the current working directory. So, for example, a path of `imports/ldif.ldif` here refers to `instance-root/imports/ldif.ldif`.

`-n, --backendID` ***backendID***
Specify the ID of the back end into which the data should be imported. To display the available back ends in the server, use the `list-backends` command.

`-O, --overwrite`
Overwrite the specified skip file or reject file, if it already exists. If this option is not provided, any skipped or rejected entries are appended to their corresponding files rather than overwriting them. This option is only applicable if the `--rejectFile` or `--skipFile` options are provided.

`-r, --replaceExisting`
Replace existing data with the content from the import. If this option is not provided, existing entries are not overwritten. This is only applicable if the `--append` option has also been provided.

`-R, --rejectFile` ***filename***
Use the specified file to hold any rejected entries during the import. Rejected entries occur if entries are not compliant with the default schema. A comment is included before the entry indicating the reason that it was rejected. If this option is not provided, no reject file is written.

`-s, --randomSeed` ***seed***
Use the specified seed number for the random number generator when generating entries from a MakeLDIF template. Seeding the random number generator with a particular value can help to ensure that the same template and random seed always generate exactly the same data.

`--skipDNValidation`
Perform limited parental DN validation during a later part of the LDIF import. If this option is specified, no duplicate DN checking is done. Do not use this option if you are not certain that your LDIF import file is correct.

`--skipFile filename`
Use the specified file to identify entries that were skipped during the import. Skipped entries occur if entries cannot be placed under any specified base DN during an import or if the `--excludeBranch`, `--excludeAttribute`, or `--excludeFilter` option is used.

`-S, --skipSchemaValidation`
Do not perform any schema validation on the entries as they are imported. This option can provide improved import performance, but should only be used if you are certain that the import data is valid.

`--threadCount` ***count***
Specify the number of threads that are used to read the LDIF file. If this option is not specified, a default of two threads per CPU is used.

You can use this option to increase the number of threads if you are importing particularly large LDIF files, but you should not use the option unless you are certain of the resulting impact on performance.

`--tmpDirectory` ***directory***
Use the specified directory for index scratch files created during the import. If no directory is specified, the default *INSTANCE_DIR*`/OUD/import-tmp` is used.

### A.3.6.4 Task Back End Connection Options

Running an online import requires access to the tasks back end. Access to the tasks back end is provided over SSL through the administration connector. These connection options are used when the import runs online.

`-D, --bindDN` ***bindDN***
Use the bind DN to authenticate to the directory server. This option is used when performing simple authentication and is not required if SASL authentication is to be used. The default value for this option is `cn=Directory Manager`.

`-h, --hostname` ***hostname***
Contact the directory server on the specified hostname or IP address. If this option is not provided, a default of `localhost` is used.

`-j, --bindPasswordFile` ***filename***
Use the bind password in the specified file when authenticating to the directory server.

`-K, --keyStorePath` ***path***
Use the client keystore certificate in the specified path.

`-N, --certNickname` ***nickname***
Use the specified certificate for client authentication.

`-o, --saslOption` ***name***=***value***
Use the specified options for SASL authentication.

`-p, --port` ***port***
Contact the directory server at the specified administration port. If this option is not provided, a default administration port of `6664` is used.

`-P, --trustStorePath` ***path***
Use the client trust store certificate in the specified path. This option is not needed if `--trustAll` is used, although a trust store should be used when working in a production environment.

`-u, --keyStorePasswordFile` ***filename***
Use the password in the specified file to access the certificates in the client keystore. This option is only required if `--keyStorePath` is used.

`-U, --trustStorePasswordFile` ***filename***
Use the password in the specified file to access the certificates in the client trust store. This option is only required if `--trustStorePath` is used and the specified trust store requires a password in order to access its contents (most trust stores do not require this).

`-X, --trustAll`
Trust all server SSL certificates that the directory server presents. This option can be used for convenience and testing purposes, but for security reasons a trust store should be used to determine whether the client should accept the server certificate.

### A.3.6.5 Task Scheduling Options

These options are used when you specify that the import should run as a scheduled task.

`--completionNotify` ***emailAddress***
Specify the email address of a recipient to be notified when the task completes. This option can be specified more than once in a single command.

`--dependency` ***taskId***
Specify the ID of a task upon which this task depends. A task does not start executing until all of its dependencies have completed execution.

`--errorNotify` ***emailAddress***
Specify the email address of a recipient to be notified if an error occurs when this task executes. This option can be specified more than once in a single command.

`--failedDependencyAction` ***action***
Specify the action that this task will take if one of its dependent tasks fails. The value must be one of `PROCESS`, `CANCEL`, or `DISABLE`. If no value is specified, the default action is `CANCEL`.

`--recurringTask` ***schedulePattern***
Indicates that the task is recurring and will be scheduled according to the `schedulePattern`, expressed as a crontab(5) compatible time and date pattern.

`-t, --start` ***startTime***
Indicates the date and time at which the operation starts when scheduled as a directory server task expressed in the format `YYYYMMDDhhmmss`. A value of 0 schedules the task for immediate execution. When this option is specified, the operation is scheduled to start at the specified time after which the command exits immediately.

### A.3.6.6 Command Input/Output Options

`--noPropertiesFile`
Indicates that a properties file is not used to obtain the default command-line options.

`--propertiesFilePath` ***path***
Specify the path to the properties file that contains the default command-line options.

`-Q, --quiet`
Run in quiet mode. Using quiet mode, no output is generated unless a significant error occurs during the import process.

`-d, --debug`
Use debug mode (verbose). Using debug mode, all advanced or debug messages are output.

### A.3.6.7 General Options

`-?, -H, --help`
Display command-line usage information for the command and exit without making any attempt to run an import.

`-V, --version`
Display the version information for the directory server and exit rather than attempting to run this command.

### A.3.6.8 Examples

The following examples show how to use the directory server commands.

#### *Example A–86 Running an Offline Import*

This example imports an LDIF file to the `userRoot` back end. The LDIF file path must be an absolute path on all platforms. The directory server must be stopped before running an offline import.

```
$ stop-ds
$ import-ldif -b dc=example,dc=com -n userRoot -l /usr/tmp/Example.ldif
```

#### *Example A–87 Importing Part of an LDIF File Offline*

This example imports part of an LDIF file to the `userRoot` back end. The import includes the base DN `dc=example,dc=com` but excludes the branch `ou=people`. Existing entries are replaced (`-r`) and information about any rejected entries are written to `/usr/tmp/rejects.ldif`. The LDIF file path must be an absolute path on all platforms. The directory server must be stopped before running an offline import.

```
$ stop-ds
$ import-ldif -b dc=example,dc=com -B "ou=people,dc=example,dc=com" \
  -l /usr/tmp/Example.ldif -n userRoot -r -R /usr/tmp/rejects.ldif
```

#### *Example A–88 Importing Data From a MakeLDIF Template*

This example imports sample data from a MakeLDIF template to the `userRoot` back end. The random seed (`-s`) determines the randomness of the data. The directory server must be stopped before running an offline import.

```
$ stop-ds
$ import-ldif -n userRoot -A example.template -s 0
```

#### *Example A–89 Importing User Attributes Only*

This example imports an LDIF file to the `userRoot` back end. Only user attributes are imported, specified by `-i "*"`. The LDIF file path must be an absolute path on all platforms. On some systems, you might be required to enclose the asterisk in quotation marks (`"*"`) or to escape the asterisk using a character appropriate to your shell. The directory server must be stopped before running an offline import.

```
$ stop-ds
$ import-ldif -b dc=example,dc=com -n userRoot -l /usr/tmp/Example.ldif -i "*"
```

#### *Example A–90 Importing User Attributes and Excluding an Attribute*

This example imports an LDIF file to the `userRoot` back end. All user attributes are imported, specified by `-i "*"`, but the `roomnumber` attribute is excluded. The LDIF file path must be an absolute path on all platforms. On some systems, you might be

required to enclose the asterisk in quotation marks ("*") or to escape the asterisk using a character appropriate to your shell. The directory server must be stopped before running an offline import.

```
$ stop-ds
$ import-ldif -b dc=example,dc=com -n userRoot -l /usr/tmp/Example.ldif \
  -i "*" -e "roomnumber"
```

#### *Example A–91 Importing Operational Attributes Only*

This example imports an LDIF file to the `userRoot` back end. Only operational attributes are imported, specified by `-i "+"`. The LDIF file path must be an absolute path on all platforms. On some systems, you might be required to enclose the plus sign in quotation marks ("+") or to escape the plus sign using a character appropriate to your shell. The directory server must be stopped before running an offline import.

```
$ stop-ds
$ import-ldif -b dc=example,dc=com -n userRoot -l /usr/tmp/Example.ldif -i "+"
```

#### *Example A–92 Importing Selected User and Operational Attributes*

This example imports an LDIF file to the `userRoot` back end. Only the `uid`, `cn`, `sn`, `dc`, and `creatorsname` attributes are imported. The LDIF file path must be an absolute path on all platforms. The directory server must be stopped before running an offline import.

```
$ stop-ds
$ import-ldif -b dc=example,dc=com -n userRoot -l /var/tmp/Example.ldif \
  -i "uid" -i "cn" -i "sn" -i "dc" -i "creatorsname"
```

#### *Example A–93 Running an Online Import*

An import is automatically run online if you specify any of the task back end connection options. Because an online import contacts the server over SSL, you must specify how to trust the SSL server certificate. This examples uses the `-X` option to trust all certificates.

```
$ import-ldif -h localhost -p 6664 -D "cn=Directory Manager" -j /path/pwd-file \
  -X -b dc=example,dc=com -n userRoot -l /usr/tmp/Example.ldif
```

#### *Example A–94 Scheduling an Import*

You can schedule an import to run at some future date by using the `-t` or `--start` option to specify the start time. Like a regular online import, a scheduled import contacts the task back end of a running directory server and the relevant task back end connection options must be specified.

This example schedules an import to the `userRoot` back end to start on December 24.

```
$ import-ldif -h localhost -p 6664 -D "cn=Directory Manager" -j /path/pwd-file \
  -X -b dc=example,dc=com -n userRoot -l /usr/tmp/Example.ldif \
  --start 20081224121500
Import task 2008101712361910 scheduled to start Dec 24, 2008 12:15:00 PM SAST
```

You can view a scheduled task by using the `manage-tasks` command. For more information, see Section 14.4, "Configuring Commands As Tasks."

### A.3.6.9 Exit Codes

- **Offline mode**. An exit code of 0 indicates that the operation completed successfully. A non-zero exit code indicates that an error occurred during processing.

- **Online mode**. If `-t` or `--start` is specified, an exit code of 0 indicates that the task was created successfully. A nonzero exit code indicates that an error occurred when the task was created. If `-t` or `--start` is not specified, the exit codes are the same as those specified for offline mode.

#### A.3.6.10 Using a Properties File

The directory server supports the use of a *properties file* that passes in any default option values used with the `export-ldif` command. The properties file is convenient when working in different configuration environments, especially in scripted or embedded applications. For more information, see Section A.1.2, "Using a Properties File With Server Commands."

#### A.3.6.11 Location

- UNIX and Linux: *INSTANCE_DIR*`/OUD/bin/import-ldif`
- Windows: *INSTANCE_DIR*`\OUD\bat\import-ldif.bat`

#### A.3.6.12 Related Commands

- Section A.3.5, "export-ldif"
- Section A.3.7, "ldif-diff"
- Section A.3.8, "ldifmodify"
- Section A.3.9, "ldifsearch"
- Section A.2.8, "manage-tasks"

### A.3.7 ldif-diff

The `ldif-diff` command identifies the differences between two LDIF files.

#### A.3.7.1 Synopsis

`ldif-diff` *options*

#### A.3.7.2 Description

The `ldif-diff` command can be used to identify the differences between two LDIF files. The resulting output can be displayed on the terminal or saved to an output file. The resulting output contains all of the information necessary for someone to reverse any changes if necessary. For modify operations, only sets of `add` and `delete` change types are used, not the `replace` change type. For delete operations, the contents of the entry that has been removed are included in the changes displayed in the form of comments.

This command was designed to work on small data sets. It is only suitable in cases in which both the source and target data sets can fit entirely in memory at the same time. It is not intended for use on large data sets that cannot fit in available memory.

---

**Note:** The `ldif-diff` command is not intended for large files. Running the `ldif-diff` command on LDIF files over a certain size (around 600 Kbytes on Windows systems, larger on UNIX systems) might result in a memory error similar to the following:

```
Exception in thread "main"
java.lang.OutOfMemoryError: Java heap space.
```

---

### A.3.7.3 Options

The `ldif-diff` command accepts an option in either its short form (for example, `-o` *outputFile*) or its long form equivalent (for example, `--outputLDIF` *outputFile*).

`-a, --ignoreAttrs` ***file***
Specify a file containing a list of attributes to ignore when computing the difference

`--checkSchema`
Consider the syntax of the attributes as defined in the schema to make the value comparison. The specified LDIF files must be conform to the server schema.

`-e, --ignoreEntries` ***file***
Specify a file containing a list of entries (DNs) to ignore when computing the difference

`-o, --outputLDIF` ***outputLDIF***
Specify the path to the output file to record the changes between the source and target LDIF data. If this is not provided, then the change information will be written to standard output.

`-O, --overwriteExisting`
Overwrite the output file specified with the `--outputLDIF` option. This option indicates that if the specified output file already exists that the file should be overwritten rather than appending to it. The option is only applicable if `--outputLDIF` is used.

`-s, --sourceLDIF` ***sourceLDIF***
Specify the path to the source LDIF file, which contains the original data with no changes applied. This option is required.

`-S, --singleValueChanges`
Run in *Single Value Change* mode, in which each modify operation is broken into a separate modification per attribute value. For example, if a single modification adds five values to an attribute, the changes appear in the output as five separate modifications, each adding one attribute.

`-t, --targetLDIF` ***targetLDIF***
Specify the path to the target LDIF file that contains the differences from the source LDIF. This option is required.

`-?, -H, --help`
Display command usage information and exit without attempting to perform any additional processing.

`-V, --version`
Display the directory server version information and exit rather than attempting to run this command.

### A.3.7.4 Examples

The following examples show how to use the `ldif-diff` command.

#### *Example A–95 Comparing Two LDIF files and Sending the Differences to Standard Output*

The following command compares a source file (`-s`) with a target file (`-t`) and outputs the differences. For Windows platforms, specify the paths for the source file (for

example, `-s \temp\quentin.ldif`) and the target file (for example, `-t \temp\quentin.ldif`):

```
$ ldif-diff -s /usr/local/quentin.ldif -t /usr/local/quentinr.ldif

dn: uid=qcubbins,ou=People,dc=example,dc=com
changetype: delete
# objectClass: person
# objectClass: organizationalPerson
# objectClass: top
# objectClass: inetOrgPerson
# cn: Quentin Cubbins
# sn: Cubbins
# uid: qcubbins
# userPassword: qcubbins
# givenName: Quentin
# description: This is Quentin's description.
# mail: qcubbins@example.com

dn: uid=qrcubbins,ou=People,dc=example,dc=com
changetype: add
objectClass: person
objectClass: organizationalPerson
objectClass: top
objectClass: inetOrgPerson
cn: Quentin R Cubbins
sn: Cubbins
uid: qrcubbins
userPassword: qrcubbins
givenName: Quentin
description: This is Quentin R's description.
mail: qrcubbins@example.com
```

***Example A–96 Comparing Two LDIF files and Sending the Differences to a File***

The following command compares a source file (`-s`) with a target file (`-t`) and sends the output to a file (`-o`). For Windows platforms, specify the paths for the source file (for example, `-s \temp\quentin.ldif`) and the target file (for example, `-t \temp\quentin.ldif`):

```
$ ldif-diff -s /usr/local/quentin.ldif -t /usr/local/quentinr.ldif \
  -o output.ldif
```

### A.3.7.5 Exit Codes

An exit code of 0 indicates that the operation completed successfully. An exit code of 1 or greater indicates that an error occurred during processing.

### A.3.7.6 Location

- UNIX and Linux: *INSTANCE_DIR*`/OUD/bin/ldif-diff`
- Windows: *INSTANCE_DIR*`\OUD\bat\ldif-diff.bat`

### A.3.7.7 Related Commands

- Section A.3.9, "ldifsearch"
- Section A.3.8, "ldifmodify"

- Section A.3.11, "make-ldif"

## A.3.8 ldifmodify

The `ldifmodify` command makes changes to the contents of an LDIF file.

### A.3.8.1 Synopsis

`ldifmodify` *options*

### A.3.8.2 Description

The `ldifmodify` command can be used to make changes to the contents of an LDIF file. Although similar to the `ldapmodify` command, the `ldifmodify` command does not connect to the directory server but rather operates locally on the LDIF file. The command also does not accept change information on standard input. It must read all changes from a file.

To make it possible to operate on very large LDIF files with limited amounts of memory, the following limitations will be enforced on the types of changes that can be made:

- **No modify DNs**. Modify DN operations are not supported. Only add, delete, and modify operations will be allowed.
- **No concurrent modify or delete operations**. It is not possible to modify or delete an entry that is to be added during the course of processing.

### A.3.8.3 Options

All options (with the exception of `--help` and `--version`) are required. The `ldifmodify` command accepts an option in either its short form (for example, `-m` *changeFile*) or its long form equivalent (for example, `--changesLDIF` *changeFile*).

`-m, --changesLDIF` ***changeFile***
Specify the path to the file containing the changes to apply. The contents of this file must be in LDIF change format.

`-s, --sourceLDIF` ***sourceFile***
Specify the path to the source LDIF file, which contains the data to be updated.

`-t, --targetLDIF` ***targetFile***
Specify the path to the target LDIF file, which will consist of the data from the source LDIF with all of the specified changes applied.

`-?, -H, --help`
Display command usage information and exit without attempting to perform any additional processing.

`-V, --version`
Display the directory server version information and exit rather than attempting to run this command.

### A.3.8.4 Examples

The following examples show how to use the `ldifmodify` command.

#### *Example A–97 Modifying an LDIF File*

Suppose that the source file is as follows:

```
dn: uid=qcubbins,ou=People,dc=example,dc=com
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
uid: qcubbins
givenName: Quentin
sn: Cubbins
cn: Quentin Cubbins
mail: qcubbins@example.com
userPassword: qcubbins
description: This is Quentin's description.
```

And suppose that the update (change) file is as follows:

```
## Add new telephone number for Quentin Cubbins
dn: uid=qcubbins,ou=People,dc=example,dc=com
changetype: modify
add: telephoneNumber
telephoneNumber: 512-401-1241
```

The following command updates a source file (`-s`) with changes listed in a modify file (`-m`) and outputs to a target file (`-t`). For Windows platforms, use the file paths for the modify file (for example, `-m \temp\update.ldif`), the source file (for example, `-s \temp\quentin.ldif`), and the target file (for example, `-s \temp\quentin_updated.ldif`):

```
$ ldifmodify -m /usr/local/update.ldif -s /usr/local/quentin.ldif \
-t /usr/local/quentin_updated.ldif
```

The updated file is as follows:

```
dn: uid=qcubbins,ou=People,dc=example,dc=com
objectClass: inetOrgPerson
objectClass: person
objectClass: top
objectClass: organizationalPerson
sn: Cubbins
userPassword: qcubbins
description: This is Quentin's description.
cn: Quentin Cubbins
telephoneNumber: 512-401-1241
givenName: Quentin
uid: qcubbins
mail: qcubbins@example.com
```

### A.3.8.5 Exit Codes

An exit code of 0 indicates that the operation completed successfully. An exit code of 1 or greater indicates that an error occurred during processing.

### A.3.8.6 Location

- UNIX and Linux: *INSTANCE_DIR*`/OUD/bin/ldifmodify`
- Windows: *INSTANCE_DIR*`\OUD\bat\ldifmodify.bat`

### A.3.8.7 Related Commands

- Section A.3.9, "ldifsearch"

- Section A.3.7, "ldif-diff"
- Section A.3.11, "make-ldif"

## A.3.9 ldifsearch

The `ldifsearch` command performs searches in an LDIF file.

### A.3.9.1 Synopsis

`ldifsearch` [*options*]

### A.3.9.2 Description

The `ldifsearch` command can be used to perform searches in an LDIF file. Although similar to the `ldapsearch` command, the `ldifsearch` command does not perform any LDAP communication with the directory server but rather operates locally on the LDIF file.

### A.3.9.3 Options

The `ldifsearch` command accepts an option in either its short form (for example, `-b` *baseDN*) or its long form equivalent (for example, `--baseDN` *baseDN*).

`-b, -baseDN` ***baseDN***
Specify the base DN to use for the search operation. Multiple base DNs can be provided by using this option multiple times. If multiple values are provided, then an entry will be examined if it is within the scope of any of the search bases. If no search base is provided, then any entry contained in the LDIF files will be considered in the scope of the search.

`-f, --filterFile` ***filterFile***
Specify the path to a file containing one or more filters to use when processing the search operation. If there are to be multiple filters, then the file should be structured with one filter per line. If this option is used, then any trailing options will be treated as separate attributes. Otherwise, the first trailing option must be the search filter.

`-l, -ldifFile` ***ldifFile***
Specify the path to the LDIF file containing the data to be searched. Multiple LDIF files can be specified by providing this option multiple times. This option is required.

`-o, -outputFile` ***outputFile***
Specify the path to the output file that contains the entries matching the provided search criteria. If this option is not provided, the matching entries will be written to standard output.

`-O, --overwriteExisting`
Overwrite the output file specified with the `--outputFile` option. This option indicates that if the specified output file already exists that the file should be overwritten rather than appending the data to existing data. This is only applicable if the `--outputFile` option is used.

`-s, -searchScope` ***searchScope***
Specify the scope of the search operation. Its value must be one of the following:

- `base` Examine only the entry specified by the `--baseDN` option.
- `one` Examine only the entry specified by the `--baseDN` option and its immediate children.

- `sub` or `subordinate` Examine the entry specified by the `--baseDN` option and its subtree.

Default value `sub` if the option is not specified.

`-t, --timeLimit` ***numSeconds***
Indicate the maximum length of time in seconds that should be spent performing the searches. After this length of time has elapsed, the search ends.

`-z, --sizeLimit` ***sizeLimit***
Set the maximum number of matching entries that the directory server should return to the client. If this is not provided, then there will be no maximum requested by the client. Note that the directory server can enforce a lower size limit than the one requested by the client.

`-T, --dontWrap`
Do not wrap long lines when displaying matching entries. If this option is not provided, long lines will be wrapped (in a manner compatible with the LDIF specification) to fit on an 80-column terminal.

`-?, -H, --help`
Display command usage information and exit without attempting to perform any additional processing.

`-V, --version`
Display the version information for the directory server.

### A.3.9.4 Examples

The following examples show how to use the `ldifsearch` command.

#### *Example A–98 Searching an LDIF File*

The following command specifies the base DN (`-b`) and searches an LDIF file (`-l`) for an entry and returns its result to the screen if any entries match the search filter `cn=Sam Carter`. For Windows platforms, use the path where the LDIF file resides (for example, `-l \temp\Example.ldif`.

```
$ ldifsearch -b dc=example,dc=com -l /usr/local/Example.ldif "(cn=Sam Carter)"

dn: uid=scarter,ou=People,dc=example,dc=com
objectClass: inetOrgPerson
objectClass: person
objectClass: top
objectClass: organizationalPerson
ou: Accounting
ou: People
sn: Carter
facsimiletelephonenumber: +1 408 555 9751
roomnumber: 4600
userpassword: sprain
l: Sunnyvale
cn: Sam Carter
telephonenumber: +1 408 555 4798
uid: scarter
givenname: Sam
mail: scarter@example.com
```

***Example A–99 Searching an LDIF File by Using a Filter File***

Suppose that the file, `filter.ldif`, which contains the following search filter:

```
(&(ou=Accounting)(l=Cupertino))
```

The following command searches the LDIF file for entries that match the filter in the search filter file and outputs the results in an output file. The command specifies the base DN (`-b`) and searches the LDIF file (`-l`) using the search filter file (`-f`) and outputs the results in a file (`-o`). For Windows platforms, use the file paths for the LDIF file (for example, `-l \temp\Example.ldif`), the filter file (for example, `-f \temp\filter.ldif`), and the output file (for example, `-o \temp\results.ldif`):

```
$ ldifsearch -b dc=example,dc=com -l /usr/local/Example.ldif -f
/usr/local/filter.ldif \
-o /home/local/results.ldif
```

#### A.3.9.5 Exit Codes

An exit code of 0 indicates that the operation completed successfully. An exit code of 1 or greater indicates that an error occurred during processing.

#### A.3.9.6 Location

- UNIX and Linux: *INSTANCE_DIR*/OUD/bin/ldifsearch
- Windows: *INSTANCE_DIR*\OUD\bat\ldifsearch.bat

#### A.3.9.7 Related Commands

- Section A.3.8, "ldifmodify"
- Section A.3.7, "ldif-diff"

### A.3.10 list-backends

The `list-backends` command displays information about the available back ends.

#### A.3.10.1 Synopsis

`list-backends` [*options*]

#### A.3.10.2 Description

The `list-backends` command can be used to obtain information about the back ends defined in a directory server instance. Back ends are responsible for providing access to the server database.

The `list-backends` command has three modes of operation:

- **No options.** When invoked with no options, display the back-end IDs for all back ends configured in the server, along with the base DNs for those back ends.
- **With backend ID.** When used with the `--backendID`, list all of the base DNs for the back end with the specified back-end ID.
- **With baseDN.** When used with the `--baseDN` option, list the back-end ID of the back end that should be used to hold the entry with the given DN and also indicate whether that DN is one of the configured base DNs for that back end.

### A.3.10.3 Options

The following are available for use but are not required. The `list-backends` command accepts an option in either its short form (for example, `-b` *baseDN*) or its long form equivalent (for example, `--baseDN` *baseDN*).

### A.3.10.4 Command Options

`-b, --baseDN` *baseDN* Specify the base DN from which the `list-backends` command should list the back-end ID. The option also indicates whether the specified DN is a baseDN for that back end.

`-n, --backendID` *backendID* Specify the back-end ID from which the command should display the associated base DN. This option can be used multiple times to display the base DNs for multiple back ends.

### A.3.10.5 General Options

`-?, -H, --help` Display the command usage information and exit immediately without taking any other action.

`-V, --version` Display the directory server version information and exit rather than attempting to run this command.

### A.3.10.6 Examples

The following examples show how to use the `list-backends` command.

#### *Example A–100 Listing the Current Back Ends*

The following command lists the current back ends on the directory server:

```
$ list-backends

Backend ID  Base DN
----------  -----------------
backup      cn=backups
config      cn=config
monitor     cn=monitor
schema      cn=schema
tasks       cn=tasks
userRoot    dc=example,dc=com
```

#### *Example A–101 Listing the Back-end ID*

The following command lists the back-end ID on the directory server:

```
$ list-backends --backendID monitor

Backend ID  Base DN
----------  ----------
monitor     cn=monitor
```

#### *Example A–102 Listing the Base DN*

The following command lists the base DN on the directory server:

```
$ list-backends --baseDN cn=backups

The provided DN 'cn=backups' is a base DN for the back end 'backup'
```

#### A.3.10.7 Exit Codes

An exit code of 0 indicates that the operation completed successfully. An exit code of 1 indicates that an error occurred during processing.

#### A.3.10.8 Location

- UNIX and Linux: *INSTANCE_DIR*`/OUD/bin/list-backends`
- Windows: *INSTANCE_DIR*`\OUD\bat\list-backends.bat`

### A.3.11 make-ldif

The `make-ldif` command generates LDIF data based on a template file.

#### A.3.11.1 Synopsis

`make-ldif` [*options*]

#### A.3.11.2 Description

The `make-ldif` command can be used to generate LDIF data based on a template file. The command allows you to construct any amount of realistic sample data that is suitable for use in applications, such as performance and scalability testing, or to attempt to reproduce a problem observed in a production environment.

#### A.3.11.3 Options

The `make-ldif` command accepts an option in either its short form (for example, `-o` *ldifFile*) or its long form equivalent (for example, `--ldifFile` *ldifFile*).

`-o, --ldifFile` ***ldifFile***
Specify the path to the LDIF file to which the generated data should be written. This is a required option.

`-s, --randomSeed` ***seed***
Specify the integer value that should be used to seed the random number generator. If a random seed is provided, then generating data based on the same template file with the same seed will always generate exactly the same LDIF output. If no seed is provided, then the same template file will likely generate different LDIF output each time it is used.

`-t, --templateFile` ***templateFile***
Specify the path to the template file that describes the data to be generated. This is a required option. You must specify an absolute path to the template file.

`-?, -H, --help`
Display command-line usage information for the command and exit without making any attempt to run the command.

`-V, --version`
Display the version information for the directory server.

#### A.3.11.4 Examples

The following examples show how to use the `make-ldif` command.

***Example A–103 Creating a Sample LDIF File***

The following command creates an LDIF file using the template (`-t`), writes to an output file (`-o`), and specifies the random seed (`-s`). For Windows platforms, enter the file paths to your output LDIF file (for example, `-o path\to\Example.ldif`) and to your template file (for example, `-t` *INSTANCE_DIR*`\OUD\config\MakeLDIF\example.template`).

The `example.template` file is located in the *INSTANCE_DIR*`/OUD/config/MakeLDIF directory`.

```
$ make-ldif -o /path/to/sample.ldif -s 0 \
-t INSTANCE_DIR/OUD/config/MakeLDIF/example.template

Processed 1000 entries
Processed 2000 entries
Processed 3000 entries
Processed 4000 entries
Processed 5000 entries
Processed 6000 entries
Processed 7000 entries
Processed 8000 entries
Processed 9000 entries
Processed 10000 entries
LDIF processing complete.  10003 entries written
```

***Example A–104 Creating a Large Sample LDIF File***

The `example.template` file (located in the installation directory under *INSTANCE_DIR*`/OUD/config/MakeLDIF`) contains a variable that sets the number of entries generated by the `make-ldif` command. You can change the number to create a very large sample LDIF file for your tests.

Open the `example.template` file, and change the `numusers` variable. By default, the variable is set to `10001`. In this example, set the variable to `1000001`:

```
define suffix=dc=example,dc=com
define maildomain=example.com
define numusers=1000001
...
```

Rerun the `make-ldif` command:

```
$ make-ldif -o /path/to/sample.ldif -s 0 \
-t INSTANCE_DIR/OUD/config/MakeLDIF/example.template
...
Processed 999000 entries
Processed 1000000 entries
LDIF processing complete.  1000003 entries written
```

### A.3.11.5 Exit Codes

An exit code of 0 indicates that the operation completed successfully. An exit code of 1 indicates that an error occurred during processing.

### A.3.11.6 Locations

- UNIX and Linux: *INSTANCE_DIR*`/OUD/bin/make-ldif`
- Windows: *INSTANCE_DIR*`\OUD\bat\make-ldif.bat`

### A.3.11.7 Related Commands

- Section A.3.9, "ldifsearch"
- Section A.3.8, "ldifmodify"
- Section A.3.7, "ldif-diff"

## A.3.12 manage-account

The `manage-account` command manages user account information, primarily related to password policy state details.

### A.3.12.1 Synopsis

`manage-account` *subcommands options*

### A.3.12.2 Description

The `manage-account` command manages user account information, primarily related to password policy state details. The command interacts with the Password Policy State extended operation, which returns account, login, and password information for a user. Although the Password Policy State extended operation allows multiple operations per use, the `manage-account` command can run only one operation at a time. Users must have the `password-reset` privilege to use the Password Policy State extended operation.

Note that all time values are returned in generalized time format. All duration values are returned in seconds.

The `manage-account` command connects to the server over SSL through the administration connector (described in Section 14.3, "Managing Administration Traffic to the Server").

### A.3.12.3 Subcommands

`clear-account-is-disabled`
Clear the disabled state for the user account. This will have the effect of enabling the account if it is disabled.

`get-account-expiration-time`
Return the account expiration time.

`get-account-is-disabled`
Return the disabled state for the user account.

`get-all`
Return all Password Policy State information for the user account.

`get-authentication-failure-times`
Return the authentication failure times for the user account.

`get-grace-login-use-times`
Return the grace login use times for the user account.

`get-last-login-time`
Return the last login time for the user.

`get-password-changed-by-required-time`
Return the password changed by the required time for the user.

`get-password-changed-time`
Return the time the password was last changed.

`get-password-expiration-warned-time`
Return the time the user was first warned about an upcoming password expiration.

`get-password-history`
Return the password history for the user account.

`get-password-is-reset`
Return the password reset state for the user, which indicates whether the user will be forced to change his password on the next login.

`get-password-policy-dn`
Return the DN of the password policy for a given user.

`get-remaining-authentication-failure-count`
Return the number of remaining authentication failures for the user before the user's account is locked.

`get-remaining-grace-login-count`
Return the number of remaining grace logins for the user.

`get-seconds-until-account-expiration`
Return the length of time before the account expires.

`get-seconds-until-authentication-failure-unlock`
Return the length of time before the user's account is automatically unlocked.

`get-seconds-until-idle-lockout`
Return the length of time before the account is idle-locked.

`get-seconds-until-password-expiration`
Return the length of time before the password expires.

`get-seconds-until-password-expiration-warning`
Return the length of time before the user is first warned about an upcoming password expiration.

`get-seconds-until-password-reset-lockout`
Return the length of time before the password reset lockout occurs.

`get-seconds-until-required-change-time`
Return the length of time before the user is required to change his password due to the required change time.

`set-account-is-disabled`
Disable the account. Required suboption:

`--operationValue` *true/false*. If set to `TRUE`, disable the user. If set to `FALSE`, enable the user.

### A.3.12.4 Options

The `manage-account` command accepts an option in either its short form (for example, `-b` *targetDN*) or its long form equivalent (for example, `--targetDN` *targetDN*).

`-b, --targetDN` ***targetDN***
Specify the DN of the user entry for which to get and set password policy state information.

### A.3.12.5 LDAP Connection Options

The `manage-account` command contacts the directory server over SSL through the administration connector. These connection options are used to contact the directory server.

`-D, --bindDN` ***bindDN***
Use the bind DN to authenticate to the directory server. This option is used when performing simple authentication and is not required if SASL authentication is to be used. The default value for this option is `cn=Directory Manager`.

`-h, --hostname` ***hostname***
Contact the directory server on the specified hostname or IP address. If this option is not provided, a default of `localhost` is used.

`-j, --bindPasswordFile` ***filename***
Use the bind password in the specified file when authenticating to the directory server.

`-K, --keyStorePath` ***path***
Use the client keystore certificate in the specified path.

`-N, --certNickname` ***nickname***
Use the specified certificate for client authentication.

`-o, --saslOption` ***name*=*value***
Use the specified options for SASL authentication.

`-p, --port` ***port***
Contact the directory server at the specified administration port. If this option is not provided, a default administration port of `4444` is used.

`-P, --trustStorePath` ***path***
Use the client trust store certificate in the specified path. This option is not needed if `--trustAll` is used, although a trust store should be used when working in a production environment.

`-u, --keyStorePasswordFile` ***filename***
Use the password in the specified file to access the certificates in the client keystore. This option is only required if `--keyStorePath` is used.

`-U, --trustStorePasswordFile` ***filename***
Use the password in the specified file to access the certificates in the client trust store. This option is only required if `--trustStorePath` is used and the specified trust store requires a password in order to access its contents (most trust stores do not require this).

`-X, --trustAll`
Trust all server SSL certificates that the directory server presents. This option can be used for convenience and testing purposes, but for security reasons a trust store should be used to determine whether the client should accept the server certificate.

### A.3.12.6 General Options

`-?, -H, --help`
Display command-line usage information for the command and exit without making any attempt to run the command.

`-V, --version`
Display the version information for the directory server.

### A.3.12.7 Examples

The following examples show how to use the directory server commands.

#### *Example A–105 Viewing All Password Policy State Information for a User*

The following command returns the password policy state information for a user:

```
$ manage-account get-all -h localhost -p 4444 -D "cn=Directory Manager" \
  -j /path/pwd-file -X -b "uid=scarter,ou=People,dc=example,dc=com" \

Password Policy DN:  cn=Default Password Policy,cn=Password Policies,cn=config
Account Is Disabled:  false
Account Expiration Time:
Seconds Until Account Expiration:
Password Changed Time:  19700101000000.000Z
Password Expiration Warned Time:
Seconds Until Password Expiration:
Seconds Until Password Expiration Warning:
Authentication Failure Times:
Seconds Until Authentication Failure Unlock:
Remaining Authentication Failure Count:
Last Login Time:
Seconds Until Idle Account Lockout:
Password Is Reset:  false
Seconds Until Password Reset Lockout:
Grace Login Use Times:
Remaining Grace Login Count:  0
Password Changed by Required Time:
Seconds Until Required Change Time:
```

#### *Example A–106 Disabling a User Account*

The following command disables a user's account `uid=scarter`:

```
$ manage-account set-account-is-disabled --operationValue true \
  -h localhost -p 4444 -D "cn=Directory Manager" -j /path/pwd-file -X \
  -b "uid=scarter,ou=People,dc=example,dc=com"

Account Is Disabled:  true
```

#### *Example A–107 Enabling a User Account*

The following command re-enables a user's disabled account:

```
$ manage-account clear-account-is-disabled \
  -h localhost -p 4444 -D "cn=Directory Manager" -j /path/pwd-file -X \
  -b "uid=scarter,ou=People,dc=example,dc=com"

Account Is Disabled:  false
```

#### A.3.12.8 Exit Codes

An exit code of 0 indicates that the operation completed successfully. A nonzero exit code indicates that an error occurred during processing.

#### A.3.12.9 Location

- UNIX and Linux: *INSTANCE_DIR*`/OUD/bin/manage-account`
- Windows: *INSTANCE_DIR*`\OUD\bat\manage-account.bat`

#### A.3.12.10 Related Commands

Section A.3.16, "verify-index"

### A.3.13 rebuild-index

The `rebuild-index` command rebuilds a directory server index.

#### A.3.13.1 Synopsis

`rebuild-index` *options*

#### A.3.13.2 Description

The `rebuild-index` command is used to rebuild directory server indexes. Indexes are files that contain lists of values, where each value is associated with a list of entry identifiers to suffixes in the directory server database. When the directory server processes a search request, it searches the database using the list of entry identifiers in the indexes, thus speeding up the search. If indexes did not exist, the directory server would have to look up each entry in the database, which dramatically degrades performance.

The `rebuild-index` command is useful in the following cases:

- When the `index-entry-limit` property of an index changes
- When a new index is created

The `rebuild-index` command can be run with the server online. However, the backend database is unavailable while `rebuild-index` is running. Also, the `rebuild-index` command usually runs faster with the server offline, especially if the `--rebuildAll` option is specified.

---

**Note:** As time progresses, the list of entry identifiers becomes unordered. As this happens, the performance of the `rebuild-index` command gradually decreases.

If you can avoid reindexing large databases, you should do so. Otherwise, if the performance of the `rebuild-index` command is severely compromised, reimport the database, to start with a fresh, ordered list of entry identifiers.

---

#### A.3.13.3 Options

The `rebuild-index` command accepts an option in either its short form (for example, `-b` *baseDN*) or its long form equivalent (for example, `--baseDN` *baseDN*).

### A.3.13.4 Command Options

`-b, --baseDN` ***baseDN***
Specify the base DN of a back end that supports indexing. The rebuild operation is performed on indexes within the scope of the given base DN.

`-i, --index` ***index***
Specify the name of the indexes to rebuild. For an attribute index, this is simply an attribute name. At least one index must be specified for rebuild.

`--rebuildAll`
Rebuild all indexes that are contained in the back end that is specified by the base DN. This option not only re-indexes all attribute indexes but also the `dn2id` system index, any extensible and VLV indexes, and the `dn2uri` index. The `rebuildAll` option cannot be used with the `-i` option.

`--tmpDirectory`
Specify the location of a temporary work directory for scratch index files. The default temporary work directory is *INSTANCE_DIR*`/OUD/import-tmp`.

### A.3.13.5 Task Back End Connection Options

Rebuilding an index online requires access to the tasks back end. Access to the tasks back end is provided over SSL through the administration connector. These connection options are used when the rebuild runs online.

`-D, --bindDN` ***bindDN***
Use the bind DN to authenticate to the directory server. This option is used when performing simple authentication and is not required if SASL authentication is used. The default value for this option is `cn=Directory Manager`.

`-h, --hostname` ***hostname***
Contact the directory server on the specified hostname or IP address. If this option is not provided, a default of `localhost` is used.

`-j, --bindPasswordFile` ***filename***
Use the bind password in the specified file when authenticating to the directory server.

`-K, --keyStorePath` ***path***
Use the client keystore certificate in the specified path.

`-N, --certNickname` ***nickname***
Use the specified certificate for client authentication.

`-o, --saslOption` ***name***`=`***value***
Use the specified options for SASL authentication.

`-p, --port` ***port***
Contact the directory server at the specified administration port. If this option is not provided, the default administration port of `4444` is used.

`-P, --trustStorePath` ***path***
Use the client trust store certificate in the specified path. This option is not needed if `--trustAll` is used, although a trust store should be used when working in a production environment.

`-u, --keyStorePasswordFile` ***filename***
Use the password in the specified file to access the certificates in the client keystore. This option is only required if `--keyStorePath` is used.

`-U, --trustStorePasswordFile` ***filename***
Use the password in the specified file to access the certificates in the client trust store. This option is only required if `--trustStorePath` is used and the specified trust store requires a password in order to access its contents (most trust stores do not require this).

`-X, --trustAll`
Trust all server SSL certificates that the directory server presents. This option can be used for convenience and testing purposes, but for security reasons a trust store should be used to determine whether the client should accept the server certificate.

### A.3.13.6 Task Scheduling Options

These options are used when you specify that the index should be rebuilt as a scheduled task.

`--completionNotify` ***emailAddress***
Specify the email address of a recipient to be notified when the task completes. This option can be specified more than once in a single command.

`--dependency` ***taskId***
Specify the ID of a task upon which this task depends. A task does not start executing until all of its dependencies have completed execution.

`--errorNotify` ***emailAddress***
Specify the email address of a recipient to be notified if an error occurs when this task executes. This option can be specified more than once in a single command.

`--failedDependencyAction` ***action***
Specify the action that this task will take if one of its dependent tasks fails. The value must be one of `PROCESS`, `CANCEL`, or `DISABLE`. If no value is specified, the default action is `CANCEL`.

`--recurringTask schedulePattern`
Indicates that the task is recurring and will be scheduled according to the `schedulePattern`, expressed as a crontab(5) compatible time and date pattern.

`-t, --start` ***startTime***
Indicates the date and time at which the operation starts when scheduled as a directory server task expressed in the format `YYYYMMDDhhmmss`. A value of 0 schedules the task for immediate execution. When this option is specified, the operation is scheduled to start at the specified time after which the command exits immediately.

### A.3.13.7 Utility Input/Output Options

`--propertiesFilePath` ***propertiesFilePath***
Path to the file containing default property values used for command line

`--noPropertiesFile`
No properties file will be used to get default command line argument values.

`-v, --verbose`
Use verbose mode.

### A.3.13.8 General Options

`-?, -H, --help`
Display command-line usage information for the command and exit without making any attempt to stop or restart the directory server.

`-V, --version`
Display the version information for the directory server and exit rather than attempting to run this command.

### A.3.13.9 Examples

The following examples show how to use the `rebuild-index` command.

#### *Example A–108 Rebuilding an Index*

First, display a list of indexes by using the `dsconfig` command as follows:

```
$ dsconfig -h localhost -p 4444 -D "cn=Directory Manager" -j /path/pwd-file -X \
  -n list-local-db-indexes --element-name userRoot

Local DB Index    : Type    : index-type
------------------:---------:--------------------
aci               : generic : presence
cn                : generic : equality, substring
displayName       : generic : equality, substring
ds-sync-conflict  : generic : equality
ds-sync-hist      : generic : ordering
entryUUID         : generic : equality
givenName         : generic : equality, substring
mail              : generic : equality, substring
member            : generic : equality
objectClass       : generic : equality
orclMTTenantGuid  : generic : equality
orclMTTenantUName : generic : equality, substring
orclMTUid         : generic : equality
sn                : generic : equality, substring
telephoneNumber   : generic : equality, substring
uid               : generic : equality
uniqueMember      : generic : equality
```

The following command rebuilds indexes (`-i`) with a base DN (`-b`).

Because this command runs offline, the directory server must be stopped before you run it.

```
$ rebuild-index -b dc=example,dc=com -i uid -i mail
[15/Dec/2011:15:28:01 +0100] category=JEB severity=NOTICE msgID=8847497
  msg=Rebuild of index(es) uid started with 202 total entries to process
...
[15/Dec/2011:15:28:02 +0100] category=JEB severity=NOTICE msgID=8847493
  msg=Rebuild complete. Processed 202 entries in 1 seconds (average rate
135.2/sec)
```

#### *Example A–109 Rebuilding All Indexes*

This example uses the `--rebuildAll` option to rebuild all indexes.

```
$ rebuild-index -b "dc=example,dc=com" --rebuildAll
```

***Example A–110 Rebuilding Extensible Indexes***

You can rebuild an extensible index in any of three ways:

- Rebuild all indexes by specifying the `--rebuildAll` option.
- Rebuild the attribute index on which the extensible index is based, by specifying the `-i` option. For example, `-i cn`.

  All indexes based on this attribute are rebuilt, including any extensible indexes that are associated with the attribute.
- Rebuild a specific extensible index by specifying it with the `-i` option. For example, `-i cn.es.lte` or `-i sn.en.sub`.

#### A.3.13.10 Exit Codes

An exit code of 0 indicates that the operation completed successfully. An exit code of 1 indicates that an error occurred during processing.

#### A.3.13.11 Location

- UNIX and Linux: *INSTANCE_DIR*`/OUD/bin/rebuild-index`
- Windows: *INSTANCE_DIR*`\OUD\bat\rebuild-index.bat`

#### A.3.13.12 Related Commands

- Section A.3.16, "verify-index"
- Section A.2.4, "dsconfig"

### A.3.14 restore

The `restore` command restores a backup of a directory server back end.

#### A.3.14.1 Synopsis

`restore` *options*

#### A.3.14.2 Description

The `restore` command restores a backup of a directory server back end. Only one back end can be restored at a time. You can use this command to perform a restore operation immediately, or to schedule a restore to run at a later time. For more information, see Section 14.4, "Configuring Commands As Tasks."

You can restore a back end when the server is offline or schedule a task when the server is online to restore a back end at a later stage. If the server is online, the restore command connects to the server over SSL through the administration connector. For more information about the administration connector, see Section 14.3, "Managing Administration Traffic to the Server."

#### A.3.14.3 Options

The `restore` command accepts an option in either its short form (for example, `-I` *backupID*) or its long form equivalent (for example, `--backupID` *backupID*).

`-d, --backupDirectory` ***path***
Restore using the directory that contains the backup archive. This directory must exist and must contain a backup descriptor file and one or more backups for a given back end. The backup descriptor file is read to obtain information about the available backups and the options used to create them. This is a required option.

`-I, --backupID` ***backupID***
Specify the backup ID of the backup to be restored. If this option is not provided, the latest backup contained in the backup directory is restored.

`-l, --listBackups`
Display information about the available backups contained in the backup directory. This option causes the command to exit without performing any restore.

`-n, --dry-run`
Verify that the specified backup is valid (that is, ensure that it appears to be a valid archive, and that any hash, signature matches its contents, or both). This option does not actually attempt to restore the backup.

### A.3.14.4 Task Back End Connection Options

Running an online restore requires access to the tasks back end. Access to the tasks back end is provided over SSL through the administration connector. These connection options are used when the restore runs online.

`-D, --bindDN` ***bindDN***
Use the bind DN to authenticate to the directory server. This option is used when performing simple authentication and is not required if SASL authentication is to be used. The default value for this option is `cn=Directory Manager`.

`-h, --hostname` ***hostname***
Contact the directory server on the specified hostname or IP address. If this option is not provided, a default of `localhost` is used.

`-j, --bindPasswordFile` ***filename***
Use the bind password in the specified file when authenticating to the directory server.

`-K, --keyStorePath` ***path***
Use the client keystore certificate in the specified path.

`-N, --certNickname` ***nickname***
Use the specified certificate for client authentication.

`-o, --saslOption` ***name*=*value***
Use the specified options for SASL Authentication.

`-p, --port` ***port***
Contact the directory server at the specified administration port. If this option is not provided, a default administration port of `4444` is used.

`-P, --trustStorePath` ***path***
Use the client trust store certificate in the specified path. This option is not needed if `--trustAll` is used, although a trust store should be used when working in a production environment.

`-u, --keyStorePasswordFile` ***filename***
Use the password in the specified file to access the certificates in the client keystore. This option is only required if `--keyStorePath` is used.

`-U, --trustStorePasswordFile` ***filename***
Use the password in the specified file to access the certificates in the client trust store. This option is only required if `--trustStorePath` is used and the specified trust store requires a password in order to access its contents (most trust stores do not require this).

`-X, --trustAll`
Trust all server SSL certificates that the directory server presents. This option can be used for convenience and testing purposes, but for security reasons a trust store should be used to determine whether the client should accept the server certificate.

### A.3.14.5 Task Scheduling Options

`--completionNotify` ***emailAddress***
Specify the email address of a recipient to be notified when the task completes. This option can be specified more than once in a single command.

`--dependency` ***taskId***
Specify the ID of a task upon which this task depends. A task does not start executing until all of its dependencies have completed execution.

`--errorNotify` ***emailAddress***
Specify the email address of a recipient to be notified if an error occurs when this task executes. This option can be specified more than once in a single command.

`--failedDependencyAction` ***action***
Specify the action this task will take should one if its dependent tasks fail. The value must be one of `PROCESS`,`CANCEL`,`DISABLE`. If not specified, the backup defaults to `CANCEL`.

`--recurringTask` ***schedulePattern***
Indicates that the task is recurring and will be scheduled according to the `schedulePattern`, expressed as a crontab(5) compatible time and date pattern.

`-t, --start` ***startTime***
Indicates the date and time at which the operation starts when scheduled as a directory server task expressed in the format `YYYYMMDDhhmmss`. A value of 0 causes the task to be scheduled for immediate execution. When this option is specified, the operation is scheduled to start at the specified time after which this command exits immediately.

### A.3.14.6 Command Input/Output Options

`--noPropertiesFile`
Indicate that a properties file will not be used to get the default command-line options.

`--propertiesFilePath` ***path***
Specify the path to the properties file that contains the default command-line options.

### A.3.14.7 General Options

`-?, -H, --help`
Display command-line usage information for the command and exit without making any attempt to stop or restart the server.

`-V, --version`
Display the version information for the directory server and exit rather than attempting to run this command.

### A.3.14.8 Examples

The following examples show how to use the `restore` command.

#### *Example A–111 Displaying the Backup Information*

The following command lists (`-l`) the backup information in the backup descriptor file (`backup.info`) for the directory server. You can use this option to display backup information whether the server is running or stopped.

```
$ restore -l -d /tmp/backup/userRoot
Backup ID:          20081016050258Z
Backup Date:        16/Oct/2008:09:30:00 +0200
Is Incremental:     false
Is Compressed:      true
Is Encrypted:       true
Has Unsigned Hash:  false
Has Signed Hash:    true
Dependent Upon:     none
```

#### *Example A–112 Restoring a Backup*

The following command restores a back end from the backup directory. You can only restore one back end at a time. The server must be stopped before you run this command.

```
$ stop-ds
$ restore -d /tmp/backup/userRoot
[16/Oct/2008:10:32:52 +0200] category=JEB severity=NOTICE msgID=8847445
msg=Restored: 00000000.jdb (size 321954)
```

#### *Example A–113 Restoring an Encrypted Backup*

Restoring a hashed or encrypted backup requires a connection to an online server instance, over SSL through the administration connector. When you restore an encrypted backup, you must therefore specify the connection details, including the host, administration port, bind DN and bind password. You must also specify the certificate details for the SSL connection.

The following command restores an encrypted, hashed backup. The self signed certificate is trusted using the `-X` (`--trustAll`) option.

```
$ restore -h localhost -p 4444 -D "cn=directory manager" -j /path/pwd-file -X \
  -d /tmp/backup/userRoot/
Restore task 2008101610403710 scheduled to start immediately
[16/Oct/2008:10:40:38 +0200] severity="NOTICE" msgCount=0 msgID=9896306
 message="The backend userRoot is now taken offline"
[16/Oct/2008:10:40:39 +0200] severity="NOTICE" msgCount=1 msgID=8847445
 message="Restored: 00000000.jdb (size 331434)"
```

```
[16/Oct/2008:10:40:40 +0200] severity="NOTICE" msgCount=2 msgID=8847402
 message="The database backend userRoot containing 102 entries has started"
Restore task 2008101610403710 has been successfully completed
```

***Example A–114 Scheduling a Restore***

Scheduling a restore requires online access to the tasks back end. Access to this back end is provided over SSL through the administration connector. When you schedule a restore, you must therefore specify the connection details, including the host, administration port, bind DN and bind password. You must also specify the certificate details for the SSL connection.

The following command schedules a task to restore the `userRoot` back end at a specific start time by using the `--start` option. The command sends a completion and error notification to `admin@example.com`. The self signed certificate is trusted using the `-X` (`--trustAll`) option.

You can view this scheduled task by using the `manage-tasks` command. For more information, see Section 14.4, "Configuring Commands As Tasks." You must ensure that the server is running prior to the scheduled restore date and time.

```
$ restore -h localhost -p 4444 -D "cn=directory manager" -j /path/pwd-file -X \
  -d /backup/userRoot --start 20081025121500 --completionNotify admin@example.com
\
  --errorNotify admin@example.com
Restore task 2008101610442610 scheduled to start Oct 25, 2008 12:15:00 PM SAST
```

#### A.3.14.9 Exit Codes

An exit code of 0 indicates that the operation completed successfully. An exit code of 1 indicates that an error occurred during processing.

#### A.3.14.10 Using a Properties File

The directory server supports the use of a *properties file* that passes in any default option values used with the `restore` command. The properties file is convenient when working in different configuration environments, especially in scripted or embedded applications. For more information, see Section A.1.2, "Using a Properties File With Server Commands."

#### A.3.14.11 Location

- UNIX and Linux: *INSTANCE_DIR*`/OUD/bin/restore`
- Windows: *INSTANCE_DIR*`\OUD\bat\restore.bat`

#### A.3.14.12 Related Commands

- Section A.3.3, "dbtest"
- Section A.2.8, "manage-tasks"

### A.3.15 split-ldif

The `split-ldif` command splits an LDIF file into multiple LDIF files according to a given distribution workflow element. The generated LDIF files are used to populate the partitions of a distribution deployment.

### A.3.15.1 Synopsis

`split-ldif` *options*

### A.3.15.2 Description

The `split-ldif` command splits an LDIF file into multiple LDIF files according to a given distribution workflow element. The data in the LDIF file is split based on the attributes indicated and based on the distribution type defined. The generated LDIF files are then used to populate the partitions. For each partition the split-ldif command creates a partition file as follows:

*outputDirectory*/*outputFilenamePrefix*-*partitionID*.ldif

Sometimes, the distribution algorithm is not able to determine the partition to which an entry should be sent, either because the entry does not contain all the parameters required by the algorithm, or the required parameters are present but they match no partition. In such a scenario, the output is written to an error file.

All the entries that do not have all the required parameters are written to the following error file:

*outputDirectory*/*outputFilenamePrefix*-`missingrequired-param`.ldif

All the entries that have the required parameters but whose parameters do not match any configured partition are written to the following error file:

*outputDirectory*/*outputFilenamePrefix*-`partition-not-found`.ldif

However, for the global index initialization you use the directory containing the files compatible with the global index format. The `split-ldif` command creates one directory per attribute to be indexed, and each directory contains files for initializing the global index.

The global index catalog is populated using the files in the directory created, which do not have a LDIF format. For more information, see Section A.2.7, "gicadm."

### A.3.15.3 Options

The `split-ldif` command accepts an option in either its short form (for example, `-i` *ldifFile*) or its long form equivalent (for example, `--ldifFile` *ldifFile*).

`-i, --ldifFile` ***ldifFile***
The name of the LDIF file to split. Global Index Options and Split Options can be used to customize the behavior.

`-l, --listDistributionNames`
Lists the enabled distribution workflow elements from the directory server's configuration.

> **Note:** The `-l, --listDistributionNames` option lists only the enabled distributions, because you cannot use a disabled distribution to split an ldif file.

### A.3.15.4 Global Index Options

`-x, --index` ***attributeTypeName***
Generates an index file to be used for the global index catalog, for the listed attribute type.

`-c, --onlyCatalog`
Generates only the index file.

### A.3.15.5 Split Options

`-d, --distributionName` ***distributionName***
The name of the distribution workflow element to split the data.

`-p, --forcePartitionId` ***partitionId***
Generates an index file where all the entries are distributed to the same single partition having the listed partitionId.

`-o, --outputDirectory` ***outputDirectory***
The directory where output LDIF files will be generated.

`-O, --outputFilenamePrefix` ***outputFilePrefix***
The prefix of the filename to generate (will contain the partition ID and the.ldif extension).

`-f, --force`
Overwrites generated files that may already exist from previous use.

### A.3.15.6 General Options

`-V, --version`
Display the version information for the directory server.

`-e, --help-examples`
Display examples of the usage.

`-?, -H, --help`
Display command-line usage information for the command and exit without making any attempt to stop or restart the directory server.

### A.3.15.7 Examples

***Example A–115 Using*** `split-ldif` ***to Populate a Global Index with One Indexed Attribute***

The following command uses an existing database file (`-i`) which it splits into several files, based on the distribution information already defined in the proxy deployment. The command defines the distribution workflow element name (`-d`), the database file (`-i`) to be split, and the attribute to be indexed in the global index files (`-x`). Indicating `-f` will overwrite any existing LDIF files.

You must have deployed a proxy instance with distribution before running this command.

```
$ split-ldif -d "distrib-we" -i database.ldif -x employeenumber -f
```

Assuming, for this example, that your distribution algorithm was numeric, and that you set two partitions with boundaries 1-1000 and 1000-2000. When you run the command above, the following directory and LDIF files are created:

`database-1.ldif`
This file contains all the entries from database with employee numbers from 1-999, which will be used to populate partition 1.

`database-2.ldif`
This file contains all the entries from database with employee numbers from 1000-1999, which will be used to populate partition 2.

`catalog\employeenumber`
This directory contains the global index files for the employee number attribute.

***Example A–116 Using `split-ldif` to Populate a Global Index with Several Indexed Attributes***

The following command uses an existing database file (`-i`) which it splits into several files, based on the distribution information already defined in the proxy deployment. The command defines the distribution workflow element name (`-d`), the database file (`-i`) to be split, and the attributes to be indexed in the global index files (`-x`). Indicating `-f` will overwrite any existing LDIF files.

You must have deployed a proxy instance with distribution before running this command.

```
$ split-ldif -d "distrib-we" -i database.ldif \
-x employeenumber -x uid -f
```

Assuming, for this example, that your distribution algorithm was numeric, and that you set two partitions with boundaries 1-50000 and 50000-100001. When you run the command above, the following LDIF files and directories are created:

- `database-1.ldif` - This file contains all the entries from database with employee numbers from 1-49999, which will be used to populate partition 1.
- `database-2.ldif` - This file contains all the entries from database with employee numbers from 50000-100000, which will be used to populate partition 2.
- `catalog\employeenumber` - This directory contains the global index files for the employee number attribute.
- `catalog\uid` - This directory contains the global index files for the uid attribute.

#### A.3.15.8 Location

- UNIX and Linux: *INSTANCE_DIR*`/OUD/bin/split-ldif`
- Windows: *INSTANCE_DIR*`\OUD\bat\split-ldif.bat`

#### A.3.15.9 Related Commands

`gicadm`

### A.3.16 verify-index

The `verify-index` command validates directory index data.

#### A.3.16.1 Synopsis

`verify-index` *options*

#### A.3.16.2 Description

The `verify-index` command is used to check the consistency between the index and entry data within the directory server database. This command also provides information about the number of index keys that have reached the index entry limit.

The command checks the following information:

- All entries are properly indexed
- All index data reference entries exist
- Data matches the corresponding index data

At the present time, this command is only available for a directory server back end that uses Oracle Berkeley DB Java Edition to store its information. None of the other back end types currently available maintain on-disk indexes. Therefore, there is no need to have any command that can verify index consistency.

Directory administrators can use this command when the directory server is running or stopped. Note, however, that using `verify-index` when the server is running impacts the overall performance of the directory server as well as the command. For example, on a very busy online server, the `verify-index` command could take significantly longer to process compared to running the command on an offline, or stopped, directory server.

To use this command, the `--baseDN` option must be used to specify the base DN of the back end below which to perform the validation.

### A.3.16.3 Options

The `verify-index` command accepts an option in either its short form (for example, `-b` *baseDN*) or its long form equivalent (for example, `--baseDN` *baseDN*).

### A.3.16.4 Command Options

`-b, --baseDN` ***baseDN***
Specify the base DN for which to perform the verification. The provided value must be a base DN for a back end based on the Berkeley DB Java Edition. This is a required option, and only one base DN may be provided.

`-c, --clean`
Verify that an index is "clean", which means that all of the entry IDs in all of the index keys refer to entries that actually exist and match the criteria for that index key. If this option is provided, then exactly one index should be specified using the `--index` option. If this option is not given, then the verification process will clean the `id2entry` database (which is a mapping of each entry ID to the actual data for that entry) and ensure that all of the entry contents are properly indexed.

`--countErrors`
Count the number of errors found during the verification and return that value as the exit code. Values greater than 255 will be returned as `255` due to exit code restrictions.

`-i, --index` ***index***
Specify the name of an index for which to perform the verification. If the `--clean` option is provided, then this argument must be provided exactly once. Otherwise, it may be specified zero or more times. If the option is not provided, then all indexes will be checked. For an attribute index, the index name should be the name of the attribute, and an index must be configured for that attribute in the associated back end. You can also specify the following internal indexes, which are used internally on the server:

`dn2id` - A mapping of entry DNs to their corresponding entry IDs.

`id2children` - A mapping of the entry ID for an entry to the entry IDs of its immediate children.

`id2subtree` - A mapping of the entry ID for an entry to the entry IDs of all of its subordinates.

`-v, --verbose`
Use verbose mode.

### A.3.16.5 General Options

`-?, -H, --help`
Display command-line usage information for the command and exit without making any attempt to stop or restart the server.

`-V, --version`
Display the version information for the directory server and exit rather than attempting to run this command.

### A.3.16.6 Examples

The following examples show how to use the `verify-index` command.

#### *Example A–117 Verifying an Index*

The following command verifies that the uid index (`-i uid`) under `dc=example,dc=com` (`-b dc=example,dc=com`) is "clean" (`-c`). This "clean" option checks that each entry in the `uid` index maps to an actual database entry with the `uid` attribute.

```
$ verify-index -b dc=example,dc=com -c -i uid

[26/Jul/2007:16:42:31 -0500] category=BACKEND severity=NOTICE msgID=8388709
msg=Checked 150 records and found 0 error(s) in 0 seconds (average rate 331.1/sec)
```

#### *Example A–118 Verifying an Index and Counting Errors*

The following command counts the number of discrepancies (`--countErrors`) in the `sn` (surname) index (`-i sn`) under the `dc=example,dc=com` base DN (`-b dc=example,dc=com`):

```
$ verify-index -b dc=example,dc=com -c -i sn --countErrors

[31/Jul/2007:02:23:52 -0500] category=BACKEND severity=NOTICE msgID=8388709 msg=
Checked 466 records and found 0 error(s) in 0 seconds (average rate 1298.1/sec)
[31/Jul/2007:02:23:52 -0500] category=BACKEND severity=NOTICE msgID=8388710 msg=
Number of records referencing more than one entry: 225
[31/Jul/2007:02:23:52 -0500] category=BACKEND severity=NOTICE msgID=8388711 msg=
Number of records that exceed the entry limit: 0
[31/Jul/2007:02:23:52 -0500] category=BACKEND severity=NOTICE msgID=8388712 msg=
Average number of entries referenced is 2.59/record
[31/Jul/2007:02:23:52 -0500] category=BACKEND severity=NOTICE msgID=8388713 msg=
Maximum number of entries referenced by any record is 150
```

### A.3.16.7 Exit Codes

An exit code of 0 indicates that the operation completed successfully. An exit code of 1 or greater indicates that an error occurred during processing.

### A.3.16.8 Location

- UNIX and Linux: *INSTANCE_DIR*`/OUD/bin/verify-index`
- Windows: *INSTANCE_DIR*`\OUD\bat\verify-index.bat`

#### A.3.16.9 Related Commands

- Section A.3.13, "rebuild-index"

## A.4 LDAP Client Commands

The following sections describe the LDAP client utilities:

- Section A.4.1, "ldapcompare"
- Section A.4.2, "ldapdelete"
- Section A.4.3, "ldapmodify"
- Section A.4.4, "ldappasswordmodify"
- Section A.4.5, "ldapsearch"

### A.4.1 ldapcompare

The `ldapcompare` command compares LDAP entries.

#### A.4.1.1 Synopsis

`ldapcompare` *options*

#### A.4.1.2 Description

The `ldapcompare` command is used to issue LDAP compare requests to the directory server. Compare requests can be used to determine whether a given entry or set of entries have a particular attribute-value combination. The only information returned from a successful compare operation is an indication as to whether the comparison evaluated to true or false. No other information about the entry is provided.

The syntax of the `ldapcompare` tool on the command-line can take any of these forms:

```
ldapcompare [ options ] attribute:value [ "targetDN" ... | -f DNfile]
ldapcompare [ options ] attribute::base64value [ "targetDN" ... | -f DNfile ]
ldapcompare [ options ] attribute:fileURL [ "targetDN" ... | -f DNfile ]
```

where

- *options* are the command-line options, described in the following section.
- *attribute* is the name of the attribute type, followed by one of the three ways to specify its comparative value. The attribute type name and value string should be enclosed in single quotes (') for the shell.
- *targetDN* is the distinguished name (DN) or list of DNs in which to search for the given attribute and compare its value.
- *DNfile* is a file with a list of DNs, one per line, to search for the given attribute and compare its value.

#### A.4.1.3 Options

The `ldapcompare` command accepts an option in either its short form (for example, `-D` *bindDN*) or its long form equivalent (for example, `--bindDN` *bindDN*).

### A.4.1.4 Command Options

`--assertionFilter` ***filter***
Perform a search using the LDAP assertion control (as defined in RFC 4528) to indicate that the operation should only be processed if the assertion contained in the provided filter is true.

`-c, --continueOnError`
Continue processing even if an error occurs. This applies when multiple entry DNs have been given either as trailing options or in a file specified with the `--filename` option. If an error occurs while processing a compare request, then the client will continue with the next entry DN if the `--continueOnError` option has been provided, or it will exit with an error if it was not provided.

`-f, --filename` ***filename***
Specify the path to a file that contains one or more filters to use when processing the search operation. If there are to be multiple entry DNs, then the file should be structured with one DN per line. All comparisons will be performed using the same connection to the directory server in the order that they appear in the file. If this option is not provided, at least one entry DN must follow the attribute-value assertion. If this option is used, the only trailing option required is the attribute-value assertion. The `--filename` option takes precedence over any DNs provided as additional command-line options. Additional DNs are simply ignored.

`-J, --control` ***controloid*** `[` ***criticality*** **[** **:** ***value*** **|** **: :** ***b64value*** **|** **:<** ***fileurl*** **]]**
Perform a search with the specified control in search requests sent to the directory server. This option makes it possible to include arbitrary request controls that the client cannot directly support. The value for this option must be in the form:

*oid* **[** **:** *criticality* **[** **:** *value* **|** **: :** *b64value* **|** **:<** *fileurl* **]]**

The elements of this value include:

- `oid` Use the OID for the control. For certain types of controls, a text name may be used instead of the numeric OID (for search operations, this includes `managedsait` for the manage DSA IT control). This element is required. Human-readable names can be used in place of the OID to reference controls that do not require values using the `-J` or `control` option. These OID names are the following:
  - `accountusable` or `accountusability` Use in place of the Account Usability Request Control OID: 1.3.6.1.4.1.42.2.27.9.5.8 (no value)
  - `authzid` or `authorizationidentity` Use in place of the Authorization Identity Request Control OID: 2.16.840.1.113730.3.4.16 (no value)
  - `effectiverights` Use in place of the Get Effective Rights Control OID: 1.3.6.1.4.1.42.2.27.9.5.2 (value = authorization ID)
  - `managedsait` Use in place of the Manage DSA IT Control OID: 2.16.840.1.113730.3.4.2 (no value)
  - `noop` or `no-op` Use in place of the LDAP No-op Control OID: 1.3.6.1.4.1.4203.1.10.2 (no value)
  - `pwpolicy` or `password policy` Use in place of the Password Policy Request OID: 1.3.6.1.4.1.42.2.27.8.5.1 (no value)
  - `subtreedelete` or `treedelete` Use in place of the Subtree Delete Request Control OID: 1.2.840.113556.1.4.805 (no value)

- `criticality` If `true`, the control should be marked critical (meaning that the directory server should not process the operation unless it can meet the requirements of this control). If `false`, the control should not be marked critical. If this subcommand is not provided, then the control is not marked critical.
- *value* Specifies the value for the control. This form should only be used if the value can be expressed as a string. It must not be used in conjunction with either the `::`*b64value* or `:<` *fileurl* forms. If none of these subcommands is present, then the control will not have a value.
- *b64value* Specifies the value for the control in base64-encoded form. This subcommand must not be used in conjunction with either the `:`*value* or `:<` *fileurl* forms. If none of these subcommands is present, then the control will not have a value.
- *fileurl* Specifies a URL that references a file from which the value of the control should be taken. It must not be used in conjunction with either the `:`*value* or `::`*b64value* forms. If none of these subcommands is present, then the control will not have a value.

For example, the value `1.3.6.1.4.1.42.2.27.9.5.2:true:dn:uid=dmiller,ou=people,dc=example,dc=com` will include a critical control with an OID of `1.3.6.1.4.1.42.2.27.9.5.2`, marked as critical (true), and with a string value for the authorization ID `dn:uid=dmiller,ou=people,dc=example,dc=com`. Or, you can use the OID names: `effectiverights:true:dn:uid=dmiller,ou=people,dc=example,dc=com`.

`-n, --dry-run`
Run in `no-op` mode. That is, report what should happen but do not actually perform any searches or communicate with the server in any way.

### A.4.1.5 LDAP Connection Options

`-D, --bindDN` ***bindDN***
Use the bind DN to authenticate to the directory server. This option is used when performing simple authentication and is not required if SASL authentication is to be used. The default value for this option is `cn=Directory Manager`.

`-h, --hostname` ***address***
Contact the directory server on the specified host name or IP address. If it is not provided, then a default address of `localhost` will be used.

`-j, --bindPasswordFile` ***bindPasswordFile***
Use the bind password in the specified file when authenticating to the directory server. The option is used for simple authentication, as well as for password-based SASL mechanisms such as CRAM-MD5, DIGEST-MD5, and PLAIN. It is not required if no authentication is to be performed. This option must not be used in conjunction with `--bindPassword`.

SASL is not supported for a proxy server instance.

`-K, --keyStorePath` ***keyStorePath***
Use the client keystore certificate in the specified path for secure communication when using the SSL or the StartTLS extended operation. This option should only be necessary if the client needs to present a certificate to the directory server, for example, when using SASL EXTERNAL authentication.

SASL is not supported for a proxy server instance.

`-N, --certNickName` ***certNickName***
Use the specified certificate for certificate-based client authentication.

`-o, --saslOption` ***name*=*value***
Use the specified option when performing SASL authentication. Multiple SASL options can be provided by using this option multiple times, once for each option.

SASL is not supported for a proxy server instance.

`-p, --port` ***port***
Contact the directory server at the specified port. If this option is not provided, then a default port of `389` will be used.

`-P, --trustStorePath` ***trustStorePath***
Use the client trust store certificate in the specified path for secure communication when using the SSL or the StartTLS extended operation. This option is not needed if `--trustAll` is used, although a trust store should be used when working in a production environment.

`-q, --useStartTLS`
Use the StartTLS Extended Operation when communicating with the directory server. This option must not be used in conjunction with `--useSSL`.

`-r, --useSASLExternal`
Use the SASL EXTERNAL mechanism for authentication, which attempts to identify the client by using an SSL certificate that it presents to the directory server. If this option is used, then the `--keyStorePath` option must also be provided to specify the path to the client keystore and either the `--useSSL` or the `--useStartTLS` option must be used to establish a secure communication channel with the server.

SASL is not supported for a proxy server instance.

`--trustStorePassword` ***trustStorePassword***
Use the password needed to access the certificates in the client trust store. This option is only required if `--trustStorePath` is used and the specified trust store requires a password in order to access its contents (which most trust stores do not require). This option must not be used in conjunction with `--trustStorePasswordFile`.

`-u, --keyStorePasswordFile` ***keyStorePasswordFile***
Use the password in the specified file to access the certificates in the client keystore. This option is only required if `--keyStorePath` is used. This option must not be used in conjunction with `--keyStorePassword`.

`-U, --trustStorePasswordFile` ***trustStorePasswordFile***
Use the password in the specified file to access the certificates in the client trust store. This option is only required if `--trustStorePath` is used and the specified trust store requires a password in order to access its contents (most trust stores do not require this). This option must not be used in conjunction with `--trustStorePassword`.

`-V, --ldapVersion` ***version***
Set the LDAP protocol version that the client should use when communicating with the directory server. The value must be either `2` (for LDAPv2 communication) or `3` (for LDAPv3). If this option is not provided, then the client will use LDAPv3.

`-w, --bindPassword` ***bindPassword***
Use the bind password when authenticating to the directory server. This option can be used for simple authentication as well as password-based SASL mechanisms. This

option must not be used in conjunction with `--bindPasswordFile`. To prompt for the password, type `-w -`.

SASL is not supported for a proxy server instance.

`-W, --keyStorePassword` ***keyStorePassword***
Use the password needed to access the certificates in the client keystore. This option is only required if `--keyStorePath` is used. This option must not be used in conjunction with `--keyStorePasswordFile`.

`-X, --trustAll`
Trust any certificate that the directory server might present during SSL or StartTLS negotiation. This option can be used for convenience and testing purposes, but for security reasons a trust store should be used to determine whether the client should accept the server certificate.

`-Z, --useSSL`
Use Secure Sockets Layer when communicating with the directory server. If SSL is to be used, then the `--port` option should be used to specify the server's secure port.

### A.4.1.6 Command Input/Output Options

`--noPropertiesFile`
Indicate that a properties file will not be used to get the default command-line options.

`--propertiesFilePath` ***propertiesFilePath***
Specify the path to the properties file that contains the default command-line options.

`-v, --verbose`
Run in verbose mode, displaying process and diagnostic information on standard output.

### A.4.1.7 General Options

`-?, -H, --help`
Display command-line usage information for the command and exit without making any attempt to run the command.

`-V, --version`
Display the version information for the directory server.

### A.4.1.8 Examples

The following examples show how to use the `ldapcompare` command.

***Example A–119 Comparing an Entity for Group Membership***

The following command specifies the host name (`-h`) that is connected to port `1389` (`-p`) and verifies if an employee (`uid=scarter`) is a member of a group (`cn=Accounting Managers`).

```
$ ldapcompare -h hostname -p 1389 \
"uniquemember:uid=scarter,ou=People,dc=example,dc=com" \
"cn=Accounting Managers,ou=groups,dc=example,dc=com"

Comparing type uniquemember with value uid=scarter,ou=People,dc=example,dc=com
in entry cn=Accounting Managers,ou=groups,dc=example,dc=com
Compare operation returned true for entry
```

```
cn=Accounting Managers,ou=groups,dc=example,dc=com
```

***Example A–120 Comparing an Attribute Value to an Entry***

The following command specifies the hostname (`-h`) that is connected to port `1389` (`-p`) and verifies if an attribute (`ou=Accounting`) is present in an entity's (`cn=Sam Carter`) record.

```
$ ldapcompare -h hostname -p 1389 "ou:Accounting" \
"uid=scarter,ou=People,dc=example,dc=com"

Comparing type ou with value Accounting in entry
uid=scarter,ou=People,dc=example,dc=com
Compare operation returned true for entry uid=scarter,ou=People,dc=example,dc=com
```

***Example A–121 Using `ldapcompare` with Server Authentication***

The following command uses server authentication, specifies the host name (`-h`), SSL port (`-p`), base DN (`-b`), the bind DN (`-D`), the bind password (`-w`), trust store file path (`-P`), and checks if the attribute is present in the entry. For Windows platforms, use the path where your trust store file resides (for example, `-P \temp\certs\cert.db`).

```
$ ldapcompare -h hostname -p 1636 -D "cn=Directory Manager" \
-j pwd-file -P /home/kwinters/certs/cert.db \
'givenname:Sam' "uid=scarter,ou=People,dc=example,dc=com"

Comparing type givenname with value Sam in entry
uid=scarter,ou=People,dc=example,dc=com
Compare operation returned true for entry uid=scarter,ou=People,dc=example,dc=com
```

***Example A–122 Using `ldapcompare` with Client Authentication***

The following command uses client authentication with the compare. The command uses SSL (`-Z`) with the SSL port (`-p`), specifies the trust store file path (`-P`), the certificate nickname (`-N`), the keystore file path (`-K`), the keystore password (`-W`) and checks if the entity's given name `givenname=Sam` is present in the entry. For Windows platforms, use the path where your trust store file resides (for example, `-P \temp\certs\cert.db`) and where the path where your keystore file resides (`-K \temp\security\key.db`).

```
$ ldapcompare -h hostname -p 1636 -Z \
-P /home/kwinters/security/cert.db -N "kwcert" \
-K /home/kwinters/security/key.db -W KeyPassword \
'givenname:Sam' "uid=scarter,ou=People,dc=example,dc=com"

Comparing type givenname with value Sam in entry
uid=scarter,ou=People,dc=example,dc=com
Compare operation returned true for entry uid=scarter,ou=People,dc=example,dc=com
```

### A.4.1.9 Exit Codes

An exit code of 6 indicates that the comparison is successful. An exit code of 5 indicates that the comparison is unsuccessful. Any other exit code indicates that an error occurred during processing.

### A.4.1.10 Using a CLI Properties File

The directory server supports the use of a *properties file* that passes in any default option values used with the `ldapcompare` command. The properties file is convenient when working in different configuration environments, especially in scripted or embedded applications. For more information, see Section A.1.2, "Using a Properties File With Server Commands."

The following options can be stored in a properties file:

- `assertionFilter`
- `bindDN`
- `bindPassword`
- `bindPasswordFile`
- `certNickname`
- `continueOnError`
- `control`
- `dry-run`
- `filename`
- `hostname`
- `keyStorePassword`
- `keyStorePasswordFile`
- `keyStorePath`
- `ldapVersion`
- `port`
- `saslOption`
- `trustAll`
- `trustStorePassword`
- `trustStorePasswordFile`
- `trustStorePath`
- `useSASLExternal`
- `useSSL`
- `useStartTLS`
- `verbose`

Entries in the properties file have the following format:

```
toolname.propertyname=propertyvalue
```

For example:

```
ldapcompare.ldapport=12345
```

### A.4.1.11 Location

- UNIX and Linux: *INSTANCE_DIR*`/OUD/bin/ldapcompare`

- Windows: *INSTANCE_DIR*\OUD\bat\ldapcompare.bat

#### A.4.1.12 Related Commands

- Section A.4.2, "ldapdelete"
- Section A.4.3, "ldapmodify"
- Section A.4.4, "ldappasswordmodify"
- Section A.4.5, "ldapsearch"

## A.4.2 ldapdelete

The `ldapdelete` command issues LDAP delete requests to the directory server in order to remove entries.

### A.4.2.1 Synopsis

`ldapdelete` [*option*] [*DN*]

### A.4.2.2 Description

The `ldapdelete` command issues LDAP delete requests to the directory server in order to remove entries. Unless the `--filename` option is given, an entry DN must be given as the only trailing option to specify which entry should be removed.

### A.4.2.3 Before You Begin

Many UNIX or Linux operating systems provide an installed version of common LDAP client commands, such as `ldapsearch`, `ldapmodify`, and `ldapdelete` in the `/usr/bin` directory. You can check if a version is on your system by entering the command: `which ldapdelete`. If the command returns a value (seen below), you will need to update your `$PATH` to the *INSTANCE_DIR*/OUD/bin directory or create an alias to the directory server instance.

```
$ which ldapdelete (UNIX/Linux)
/usr/bin/ldapdelete
```

### A.4.2.4 Options

The `ldapdelete` command accepts an option in either its short form (for example, `-D` *bindDN*) or its long form equivalent (for example, `--bindDN` *bindDN*).

### A.4.2.5 Command Options

`-c, --continueOnError`
Continue processing even if an error occurs. This operation applies when multiple entry DNs have been given either as trailing options or in a file specified with the `--filename` option. If an error occurs while processing a compare request, then the client will continue with the next entry DN if the `--continueOnError` option has been provided, or it will exit with an error if that option was not provided.

`-f, --filename` ***filename***
Specify the path to a file that contains one or more filters to use when processing the search operation. If there are multiple entry DNs, then the file should be structured with one DN per line. If this option is used, then do not add any trailing options. The DN of the entry to remove should be the only trailing option.

`-J, --control` ***controloid*[:*criticality*[:*value*|::*b64value*|:<*fileurl*]]**
Perform a search with the specified control in search requests sent to the directory server. This option makes it possible to include arbitrary request controls that the client cannot directly support. The value for this option must be in the form:

*oid*[:*criticality*[:*value*|::*b64value*|:<*fileurl*]]

The elements of this value include:

- **oid.** Use the OID for the control. For certain types of controls, a text name may be used instead of the numeric OID (for search operations, this includes `managedsait` for the manage DSA IT control). This element is required. Human-readable names can be used in place of the OID to reference controls that do not require values using the -J or `control` option. These OID names are the following:

  `accountusable` or `accountusability` — Use in place of the Account Usability Request Control OID:1.3.6.1.4.1.42.2.27.9.5.8 (no value).

  `authzid` or `authorizationidentity` — Use in place of the Authorization Identity Request Control OID: 2.16.840.1.113730.3.4.16 (no value).

  `effectiverights` — Use in place of the Get Effective Rights Control OID: 1.3.6.1.4.1.42.2.27.9.5.2 (value = authorization ID).

  `managedsait` — Use in place of the Manage DSA IT Control OID: 2.16.840.1.113730.3.4.2 (no value).

  `noop` or `no-op` — Use in place of the LDAP No-op Control OID: 1.3.6.1.4.1.4203.1.10.2 (no value).

  `pwpolicy` or `password policy` — Use in place of the Password Policy Request Control OID: 1.3.6.1.4.1.42.2.27.8.5.1 (no value).

  `subtreedelete` or `treedelete` — Use in place of the Subtree Delete Request Control OID: 1.2.840.113556.1.4.805 (no value).

- **criticality.** If `true`, the control should be marked critical (meaning that the directory server should not process the operation unless it can meet the requirements of this control). If `false`, the control should not be marked critical. If this subcommand is not provided, then the control is not marked critical.

- **value.** Specifies the value for the control. This form should only be used if the value can be expressed as a string. It must not be used in conjunction with either the `::`*b64value* or `:<` *fileurl* forms. If none of these subcommands is present, then the control will not have a value.

- **b64value.** Specifies the value for the control in base64-encoded form. This subcommand must not be used in conjunction with either the `:`*value* or `:<` *fileurl* forms. If none of these subcommands is present, then the control will not have a value.

- **fileurl.** Specifies a URL that references a file from which the value of the control should be taken. It must not be used in conjunction with either the `:`*value* or `::`*b64value* forms. If none of these subcommands is present, then the control will not have a value.

  For example, the value `1.3.6.1.4.1.42.2.27.9.5.2:true:dn:uid=dmiller,ou=people,dc=example,dc=com` will include a critical control with an OID of `1.3.6.1.4.1.42.2.27.9.5.2`, marked as critical (true), and with a string value for the authorization ID `dn:uid=dmiller,ou=people,dc=example,dc=com`. Or, you can use the OID

names:
`effectiverights:true:dn:uid=dmiller,ou=people,dc=example,dc=com`.

`-n, --dry-run`
Run in `no-op` mode. That is, report what should happen but do not actually perform any searches or communicate with the server in any way.

`-x, --deleteSubtree`
Delete the specified entry and all entries below it.

### A.4.2.6 LDAP Connection Options

`-D, --bindDN` ***bindDN***
Use the bind DN to authenticate to the directory server. This option is used when performing simple authentication and is not required if SASL authentication is to be used. The default value for this option is `cn=Directory Manager`.

`-h, --hostname` ***address***
Contact the directory server on the specified host name or IP address. If it is not provided, then a default address of `localhost` will be used.

`-j, --bindPasswordFile` ***bindPasswordFile***
Use the bind password in the specified file when authenticating to the directory server. The option is used for simple authentication, as well as for password-based SASL mechanisms such as CRAM-MD5, DIGEST-MD5, and PLAIN. It is not required if no authentication is to be performed. This option must not be used in conjunction with `--bindPassword`.

SASL is not supported for a proxy server instance.

`-K, --keyStorePath` ***keyStorePath***
Use the client keystore certificate in the specified path for secure communication when using the SSL or the StartTLS extended operation. This option should only be necessary if the client needs to present a certificate to the directory server, for example, when using SASL EXTERNAL authentication.

SASL is not supported for a proxy server instance.

`-N, --certNickName` ***certNickName***
Use the specified certificate for certificate-based client authentication.

`-o, --saslOption` ***name*** = ***value***
Use the specified option when performing SASL authentication. Multiple SASL options can be provided by using this option multiple times, once for each option. See Section 20.6, "Using SASL Authentication" for more information.

SASL is not supported for a proxy server instance.

`-p, --port` ***port***
Contact the directory server at the specified port. If this option is not provided, then a default port of `389` will be used.

`-P, --trustStorePath` ***trustStorePath***
Use the client trust store certificate in the specified path for secure communication when using the SSL or the StartTLS extended operation. This option is not needed if `--trustAll` is used, although a trust store should be used when working in a production environment.

`-q, --useStartTLS`
Use the StartTLS Extended Operation when communicating with the directory server. This option must not be used in conjunction with `--useSSL`.

`-r, --useSASLExternal`
Use the SASL EXTERNAL mechanism for authentication, which attempts to identify the client by using an SSL certificate that it presents to the directory server. If this option is used, then the `--keyStorePath` option must also be provided to specify the path to the client keystore and either the `--useSSL` or the `--useStartTLS` option must be used to establish a secure communication channel with the server.

SASL is not supported for a proxy server instance.

`--trustStorePassword` ***trustStorePassword***
Use the password needed to access the certificates in the client trust store. This option is only required if `--trustStorePath` is used and the specified trust store requires a password in order to access its contents (which most trust stores do not require). This option must not be used in conjunction with `--trustStorePasswordFile`.

`-u, --keyStorePasswordFile` ***keyStorePasswordFile***
Use the password in the specified file to access the certificates in the client keystore. This option is only required if `--keyStorePath` is used. This option must not be used in conjunction with `--keyStorePassword`.

`-U, --trustStorePasswordFile` ***trustStorePasswordFile***
Use the password in the specified file to access the certificates in the client trust store. This option is only required if `--trustStorePath` is used and the specified trust store requires a password in order to access its contents (most trust stores do not require this). This option must not be used in conjunction with `--trustStorePassword`.

`-V, --ldapVersion` ***version***
Set the LDAP protocol version that the client should use when communicating with the directory server. The value must be either `2` (for LDAPv2 communication) or `3` (for LDAPv3). If this option is not provided, then the client will use LDAPv3.

`-w, --bindPassword` ***bindPassword***
Use the bind password when authenticating to the directory server. This option can be used for simple authentication as well as password-based SASL mechanisms. This option must not be used in conjunction with `--bindPasswordFile`. To prompt for the password, type `-w -`.

SASL is not supported for a proxy server instance.

`-W, --keyStorePassword` ***keyStorePassword***
Use the password needed to access the certificates in the client keystore. This option is only required if `--keyStorePath` is used. This option must not be used in conjunction with `--keyStorePasswordFile`.

`-X, --trustAll`
Trust any certificate that the directory server might present during SSL or StartTLS negotiation. This option can be used for convenience and testing purposes, but for security reasons a trust store should be used to determine whether the client should accept the server certificate.

`-Z, --useSSL`
Use Secure Sockets Layer when communicating with the directory server. If SSL is to be used, then the `--port` option should be used to specify the server's secure port.

### A.4.2.7 Command Input/Output Options

`--noPropertiesFile`
Indicate that a properties file will not be used to get the default command-line options.

`--propertiesFilePath` ***propertiesFilePath***
Specify the path to the properties file that contains the default command-line options.

`-v, --verbose`
Run in verbose mode, displaying process and diagnostic information on standard output.

### A.4.2.8 General Options

`-?, -H, --help`
Display command-line usage information for the command and exit without making any attempt to run the command.

`-V, --version`
Display the version information for the directory server.

### A.4.2.9 Examples

The following examples show how to use the `ldapdelete` command.

#### *Example A–123 Deleting an Entry from the Command Line*

The following command specifies the host name (`-h`), the port (`-p`), the bind DN (`-D`), the bind password (`-w`), and deletes a single entry:

```
$ ldapdelete -h hostname -p 1389 -D "cn=Directory Manager" -j pwd-file \
"uid=mgarza,ou=People,dc=example,dc=com"
```

#### *Example A–124 Deleting Multiple Entries by Using a DN File*

The following file contains a list of DN's for deletion. The file must list each DN on a separate line.

```
uid=mgarza,ou=People,dc=example,dc=com
uid=wsmith,ou=People,dc=example,dc=com
uid=jarrow,ou=People,dc=example,dc=com
uid=mbean,ou=People,dc=example,dc=com
```

The following command specifies the host name (`-h`), the port (`-p`), the bind DN (`-D`), and the bind password (`-w`), and reads the entries in a file for deletion. If an error occurs, the command continues (`-c`) to the next search item. For Windows platforms, use the path where the deletion file resides (for example, `-f \temp\delete.ldif`):

```
$ ldapdelete -h hostname -p 1389 -D "cn=Directory Manager" -j pwd-file \
-c -f /usr/local/delete.ldif
```

#### *Example A–125 Deleting Entries by Using Server Authentication*

The following command uses server authentication to delete an entry. The command specifies the host name (`-h`), SSL port (`-p`), bind DN (`-D`), the bind password (`-w`), trust store file path (`-P`), and LDIF file (`-f`) that contains the deletes. If an error occurs, the command continues (`-c`) to the next search item. For Windows platforms, use the path where the deletion file resides (for example, `-f \temp\delete.ldif`) and the

file where the trust store password resides (for example, `-P \temp\certs\cert.db`):

```
$ ldapdelete -h hostname -p 1636 -c -f /usr/local/delete.ldif \
-D "cn=Directory Manager" -j pwd-file \
-P /home/kwinters/certs/cert.db
```

***Example A–126 Deleting Entries by Using Client Authentication***

The following command uses client authentication to perform a delete option. The command uses SSL (`-Z`) with the SSL port (`-p`), specifies the trust store file path (`-P`), the certificate nickname (`-N`), the keystore file path (`-K`), the keystore password (`-W`) and the LDIF file (`-f`) that contains the deletions. If an error occurs, the command continues (`-c`) to the next search item. For Windows platforms, use the path where the deletion file resides (for example, `-f \temp\delete.ldif`), the file where the trust store password resides (for example, `-P \temp\certs\cert.db`), and the file where the keystore password resides (for example, `-K \temp\security\key.db`).

```
$ ldapdelete -h hostname -p 1636 -c -f /usr/local/delete.ldif \
-Z -P /home/kwinters/security/cert.db -N "kwcert" \
-K /home/kwinters/security/key.db -W keypassword
```

### A.4.2.10 Exit Codes

An exit code of 0 indicates that the operation completed successfully. A nonzero exit code indicates that an error occurred during processing.

### A.4.2.11 Using a CLI Properties File

The directory server supports the use of a properties file that passes in any default option values used with the `ldapdelete` command. The properties file is convenient when working in different configuration environments, especially in scripted or embedded applications. See Section A.1.2, "Using a Properties File With Server Commands" for more information.

The following options can be stored in a properties file:

- `bindDN`
- `bindPassword`
- `bindPasswordFile`
- `certNickname`
- `continueOnError`
- `control`
- `deleteSubtree`
- `dry-run`
- `filename`
- `hostname`
- `keyStorePassword`
- `keyStorePasswordFile`
- `keyStorePath`
- `ldapVersion`

- port
- saslOption

  SASL is not supported for a proxy server instance
- trustAll
- trustStorePassword
- trustStorePasswordFile
- trustStorePath
- useSASLExternal

  SASL is not supported for a proxy server instance.
- useSSL
- useStartTLS
- verbose

Entries in the properties file have the following format:

```
toolname.propertyname=propertyvalue
```

For example:

```
ldapdelete.ldapport=12345
```

#### A.4.2.12 Location

- UNIX and Linux: *INSTANCE_DIR*/OUD/bin/ldapdelete
- Windows: *INSTANCE_DIR*\OUD\bat\ldapdelete.bat

#### A.4.2.13 Related Commands

- Section A.4.1, "ldapcompare"
- Section A.4.3, "ldapmodify"
- Section A.4.4, "ldappasswordmodify"
- Section A.4.5, "ldapsearch"

## A.4.3 ldapmodify

The ldapmodify command modifies directory entries.

#### A.4.3.1 Synopsis

ldapmodify [*options*] [*filter*] [*attributes*]

#### A.4.3.2 Description

The ldapmodify command can be used to perform LDAP modify, add, delete, and modify DN operations in the directory server. The operations to perform in the directory server should be specified in LDIF change format, as described in RFC 2849 (http://www.ietf.org/rfc/rfc2849.txt). This change syntax uses the changetype keyword to indicate the type of change.

An add change record is straightforward, because it is a complete entry in LDIF form with a changetype value of add. For example:

```
dn: uid=john.doe,ou=People,dc=example,dc=com
changetype: add
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
uid: john.doe
givenName: John
sn: Doe
cn: John Doe
mail: john.doe@example.com
userPassword: password
```

A `delete` change record is even simpler than an `add` change record. The `add` record consists of a line with the entry DN followed by another line with a `changetype` of `delete`. For example:

```
dn: uid=john.doe,ou=People,dc=example,dc=com
changetype: delete
```

The `modify` change record is the most complex operation, because of the number of variants. The `modify` change records all start with the entry DN followed by a `changetype` of `modify`. The next line consists of either `add`, `delete`, or `replace` followed by an attribute name indicating what modification will be and to which attribute. The change record may optionally be followed by one or more lines containing the attribute name followed by a value to use for the modification (that is, a value to add to that attribute, remove from that attribute, or use to replace the existing set of values). Multiple attribute changes can be made to an entry in the same `modify` operation by separating changes with a line containing only a dash, starting the next line with a new `add`, `delete`, or `replace` tag followed by a colon and the next attribute name, and then setting of values for that attribute. For example:

```
dn: uid=john.doe,ou=People,dc=example,dc=com
changetype: modify
replace: description
description: This is the new description for John Doe
-
add: mailAlternateAddress
mailAlternateAddress: jdoe@example.com
```

Modify DN change records should always contain the `newRDN` and `deleteOldRDN` elements and can optionally contain the `newSuperior` component to specify a new parent for the target entry. For example:

```
dn: uid=john.doe,ou=People,dc=example,dc=com
changetype: moddn
newRDN: uid=jdoe
deleteOldRDN: 1
```

If no arguments are provided to the `ldapmodify` command, it attempts to interact with a Directory Server instance using an unauthenticated connection using the loopback address on port 389, and information about the changes to request will be read from standard input. This is unlikely to succeed, as it will almost certainly be necessary to at least provide arguments that will be used to specify how to authenticate to the server.

### A.4.3.3 Before You Begin

Many UNIX and Linux operating systems provide an installed version of common LDAP client commands, such as `ldapsearch`, `ldapmodify`, and `ldapdelete` in the

/usr/bin directory. You can check if a version is on your system by entering the command: which ldapmodify. If the command returns a value (seen below), you will need to update your $PATH to *INSTANCE_DIR*/OUD/bin or create an alias to the directory server instance.

```
$ which ldapmodify (Unix/Linux)
/usr/bin/ldapmodify
```

### A.4.3.4 Options

The ldapmodify command accepts an option in either its short form (for example, -D *bindDN*) or its long form equivalent (for example, --bindDN *bindDN*).

### A.4.3.5 Command Options

-a, --defaultAdd
Add entries. Treat records with no changetype element as an add request. This option can be used to add entries from a standard LDIF file that does not contain information in the LDIF change format.

--assertionFilter ***filter***
Perform a search using the LDAP assertion control (as defined in RFC 4528 (http://www.ietf.org/rfc/rfc4528.txt)) to indicate that the operation should only be processed if the assertion contained in the provided filter is true.

-c, --continueOnError
Continue processing even if an error occurs. Use this option when using multiple search filters in a file --filename. If an error occurs during processing, the directory server will continue processing the next search filter. Otherwise the command will exit before all searches have been completed.

-f, --filename ***filename***
Read modifications from the specified file containing one or more filters to use during the modify operation. The records in the LDIF file should be in the LDIF change format (that is, including the changetype element). If the LDIF file only contains entries that should be added to the directory server, then the file can be used with the --defaultAdd option even if the entries do not have a changetype element. The provided file can contain multiple changes as long as there is at least one blank line between change records.

If this option is not provided, then the ldapmodify command will attempt to read change information from standard input. This makes it possible to have the change records either provided interactively by the target user on the command line or piped into the command from some other source.

-J, --control ***controloid*[:*criticality*[:*value*|::*b64value*|:<*fileurl*]]**
Perform a search with the specified control in search requests sent to the directory server. This option makes it possible to include arbitrary request controls that the client cannot directly support. The value for this option must be in the form:

*oid*[:*criticality*[:*value*|::*b64value*|:<*fileurl*]]

The elements of this value include:

- **oid.** Use the OID for the control. For certain types of controls, a text name may be used instead of the numeric OID (for search operations, this includes managedsait for the manage DSA IT control). This element is required. Human-readable names can be used in place of the OID to reference controls that

do not require values using the -J or `control` option. These OID names are the following:

`accountusable` or `accountusability` — Use in place of the Account Usability Request Control OID: 1.3.6.1.4.1.42.2.27.9.5.8 (no value).

`authzid` or `authorizationidentity` — Use in place of the Authorization Identity Request Control OID: 2.16.840.1.113730.3.4.16 (no value).

`effectiverights` — Use in place of the Get Effective Rights Control OID: 1.3.6.1.4.1.42.2.27.9.5.2 (value = authorization ID).

`managedsait` — Use in place of the Manage DSA IT Control OID: 2.16.840.1.113730.3.4.2 (no value).

`noop` or `no-op` — Use in place of the LDAP No-op Control OID: 1.3.6.1.4.1.4203.1.10.2 (no value).

`pwpolicy` or `password policy` — Use in place of the Password Policy Request Control OID: 1.3.6.1.4.1.42.2.27.8.5.1 (no value).

`subtreedelete` or `treedelete` — Use in place of the Subtree Delete Request Control OID: 1.2.840.113556.1.4.805 (no value).

- **criticality.** If `true`, the control should be marked critical (meaning that the directory server should not process the operation unless it can meet the requirements of this control). If `false`, the control should not be marked critical. If this subcommand is not provided, then the control is not marked critical.
- **value.** Specifies the value for the control. This form should only be used if the value can be expressed as a string. It must not be used in conjunction with either the `::`*b64value* or `:<` *fileurl* forms. If none of these subcommands is present, then the control will not have a value.
- **b64value.** Specifies the value for the control in base64-encoded form. This subcommand must not be used in conjunction with either the `:`*value* or `:<` *fileurl* forms. If none of these subcommands is present, then the control will not have a value.
- **fileurl.** Specifies a URL that references a file from which the value of the control should be taken. It must not be used in conjunction with either the `:`*value* or `::`*b64value* forms. If none of these subcommands is present, then the control will not have a value.

  For example, the value `1.3.6.1.4.1.42.2.27.9.5.2:true:dn:uid=dmiller,ou=people,dc=example,dc=com` will include a critical control with an OID of `1.3.6.1.4.1.42.2.27.9.5.2`, marked as critical (true), and with a string value for the authorization ID `dn:uid=dmiller,ou=people,dc=example,dc=com`. Or, you can use the OID names: `effectiverights:true:dn:uid=dmiller,ou=people,dc=example,dc=com`.

`-n, --dry-run`
Run in `no-op` mode. That is, report what should happen but do not actually perform any searches or communicate with the server in any way.

`--postReadAttributes` ***attrList***
Use the LDAP ReadEntry Post-read Control (as defined in RFC 4527 (`http://www.ietf.org/rfc/rfc4527.txt`)) to indicate that the directory server should return a copy of the target entry as it was immediately after the update. This is

only applicable for add, modify, and modify DN operations. The value for this option should be a comma-separated list of the attributes to include in the representation of the pre-read entry. The same conventions apply to this list as for the list of attributes to return in the `ldapsearch` command (that is, it is possible to use `*` for all user attributes, `+` for all operational attributes, `@`*ocname* for all attributes in the specified objectclass, and so on). If no attributes are specified (signified with empty quotes), then all user attributes will be returned.

`--preReadAttributes` ***attrList***
Use the LDAP ReadEntry Pre-read Control (as defined in RFC 4527 (`http://www.ietf.org/rfc/rfc4527.txt`)) to indicate that the directory server should return a copy of the target entry as it was immediately before the update. This is only applicable for delete, modify, and modify DN operations. The value for this option should be a comma-separated list of the attributes to include in the representation of the pre-read entry. The same conventions apply to this list as for the list of attributes to return in the `ldapsearch` command (that is, it is possible to use `*` for all user attributes, `+` for all operational attributes, `@`*ocname* for all attributes in the specified objectclass, and so on). If no attributes are specified (signified with empty quotes), then all user attributes will be returned.

`-Y, --proxyAs` ***authzID***
Use the Proxied Authorization Control to specify the identity of the user for whom the operations should be performed. This will use version 2 of the Proxied Authorization Control as defined in RFC 4370 (`http://www.ietf.org/rfc/rfc4370.txt`). The value of the option should be an authorization ID in the form `dn:` followed by the DN of the target user (for example, `dn:uid=john.doe,ou=People,dc=example,dc=com`), or `u:` followed by the user name (for example, `u:john.doe`). If this option is not provided, then proxied authorization will not be used.

### A.4.3.6 LDAP Connection Options

`-D, --bindDN` ***bindDN***
Use the bind DN to authenticate to the directory server. This option is used when performing simple authentication. The default value for this option is `cn=Directory Manager`. It is not required when using SASL authentication or if no authentication is to be performed.

`-E, --reportAuthzID`
Use the authorization identity request control (as defined in RFC 3829 (`http://www.ietf.org/rfc/rfc3829.txt`)) in the bind request so that the directory server returns the corresponding authorization ID to the client when authentication has completed. (The line containing the authorization ID will be prefixed with a `#` character, making it a comment if the output is to be interpreted as an LDIF.)

`-h, --hostname` ***address***
Contact the directory server on the specified host name or IP address. If it is not provided, then a default address of `localhost` will be used.

`-j, --bindPasswordFile` ***bindPasswordFile***
Use the bind password in the specified file when authenticating to the directory server. The option is used for simple authentication, as well as for password-based SASL mechanisms such as CRAM-MD5, DIGEST-MD5, and PLAIN. It is not required if no authentication is to be performed. This option must not be used in conjunction with `--bindPassword`.

SASL is not supported for a proxy server instance.

`-K, --keyStorePath` ***keyStorePath***
Use the client keystore certificate in the specified path for secure communication when using the SSL or the StartTLS extended operation. This option should only be necessary if the client needs to present a certificate to the directory server, for example, when using SASL EXTERNAL authentication.

SASL is not supported for a proxy server instance.

`-N, --certNickName` ***certNickName***
Use the specified certificate for certificate-based client authentication.

`-o, --saslOption` ***name*** = ***value***
Use the specified option when performing SASL authentication. Multiple SASL options can be provided by using this option multiple times, once for each option. For information about using SASL authentication in clients, see Section 20.7, "Configuring SASL Authentication."

SASL is not supported for a proxy server instance.

`-p, --port` ***port***
Contact the directory server at the specified port. If this option is not provided, then a default port of `389` will be used.

`-P, --trustStorePath` ***trustStorePath***
Use the client trust store certificate in the specified path for secure communication when using the SSL or the StartTLS extended operation. This option is not needed if `--trustAll` is used, although a trust store should be used when working in a production environment.

`-q, --useStartTLS`
Use the StartTLS extended operation when communicating with the directory server. This option must not be used in conjunction with `--useSSL`.

`-r, --useSASLExternal`
Use the SASL EXTERNAL mechanism for authentication, which attempts to identify the client by using an SSL certificate that it presents to the directory server. If this option is used, then the `--keyStorePath` option must also be provided to specify the path to the client keystore and either the `--useSSL` or the `--useStartTLS` option must be used to establish a secure communication channel with the server.

SASL is not supported for a proxy server instance.

`--trustStorePassword` ***trustStorePassword***
Use the password needed to access the certificates in the client trust store. This option is only required if `--trustStorePath` is used and the specified trust store requires a password in order to access its contents (which most trust stores do not require). This option must not be used in conjunction with `--trustStorePasswordFile`.

`-u, --keyStorePasswordFile` ***keyStorePasswordFile***
Use the password in the specified file to access the certificates in the client keystore. This option is only required if `--keyStorePath` is used. This option must not be used in conjunction with `--keyStorePassword`.

`-U, --trustStorePasswordFile` ***trustStorePasswordFile***
Use the password in the specified file to access the certificates in the client trust store. This option is only required if `--trustStorePath` is used and the specified trust

store requires a password in order to access its contents (most trust stores do not require this). This option must not be used in conjunction with `--trustStorePassword`.

`-V, --ldapVersion` ***version***
Set the LDAP protocol version that the client should use when communicating with the directory server. The value must be either `2` (for LDAPv2 communication) or `3` (for LDAPv3). If this option is not provided, then the client will use LDAPv3.

`-w, --bindPassword` ***bindPassword***
Use the bind password when authenticating to the directory server. This option can be used for simple authentication as well as password-based SASL mechanisms. This option must not be used in conjunction with `--bindPasswordFile`. To prompt for the password, type `-w -`.

SASL is not supported for a proxy server instance.

`-W, --keyStorePassword` ***keyStorePassword***
Use the password needed to access the certificates in the client keystore. This option is only required if `--keyStorePath` is used. This option must not be used in conjunction with `--keyStorePasswordFile`.

`-X, --trustAll`
Trust any certificate that the directory server might present during SSL or StartTLS negotiation. This option can be used for convenience and testing purposes, but for security reasons a trust store should be used to determine whether the client should accept the server certificate.

`-Z, --useSSL`
Use SSL when communicating with the directory server. If SSL is to be used, then the `--port` option should be used to specify the server's secure port.

### A.4.3.7 Command Input/Output Options

`--noPropertiesFile`
Indicate that a properties file will not be used to get the default command-line options.

`--propertiesFilePath` ***propertiesFilePath***
Specify the path to the properties file that contains the default command-line options.

`-v, --verbose`
Run in verbose mode, displaying process and diagnostic information on standard output.

### A.4.3.8 General Options

`-?, -H, --help`
Display command-line usage information for the command and exit without making any attempt to run the command.

`-V, --version`
Display the version information for the directory server.

### A.4.3.9 Examples

The following examples show how to use the `ldapmodify` command.

#### *Example A–127 Adding an Entry*

The following LDIF file contains an entry for an employee:

```
dn: uid=Marcia Garza,ou=People,dc=example,dc=com
cn: Marcia Garza
sn: Garza
givenName: Marcia
objectClass: person
objectClass: inetOrgPerson
objectClass: top
objectClass: organizationalPerson
ou: Accounting
ou: People
```

The following command specifies the host name (`-h`), port (`-p`), bind DN (`-D`), bind password (`-w`), reads the modifications from the file (`-f`) and adds the entry (`-a`) to the database. For Windows platforms, specify the path to your LDIF file (for example, `-f \temp\add_entry.ldif`).

```
$ ldapmodify -h hostname -p 1389 -D "cn=Directory Manager" -j pwd-file \
-a -f /usr/local/add_entry.ldif
```

#### *Example A–128 Adding an Attribute to an Entry*

The following LDIF file modifies an entry by adding a `telephonenumber` attribute:

```
dn: uid=Marcia Garza,ou=People,dc=example,dc=com
changetype: modify
add: telephonenumber
telephonenumber: +1 408 555 8283
```

The following command specifies the host name (`-h`), port (`-p`), bind DN (`-D`), bind password (`-w`), reads the modifications from the file (`-f`) and adds an attribute to the entry. For Windows platforms, specify the path to your LDIF file (for example,

`-f \temp\add_attribute.ldif`).

```
$ ldapmodify -h hostname -p 1389 -D "cn=Directory Manager" -j pwd-file \
-f /usr/local/add_attribute.ldif
```

#### *Example A–129 Modifying the Value of an Attribute*

The following LDIF file modifies the value of the `telephonenumber` attribute:

```
dn: uid=Marcia Garza,ou=People,dc=example,dc=com
changetype: modify
replace: telephonenumber
telephonenumber: +1 408 555 6456
```

The following command specifies the hostname (`-h`), port (`-p`), bind DN (`-D`), bind password (`-w`), reads the modifications from the file (`-f`) and modifies the attribute's value. For Windows-platforms, specify the path to your LDIF file (for example, `-f \temp\modify_attribute.ldif`).

```
$ ldapmodify -h hostname -p 1389 -D "cn=Directory Manager" -j pwd-file \
-f /usr/local/modify_attribute.ldif
```

#### *Example A–130 Modifying Multiple Attributes*

The following LDIF file contains multiple modifications to an entry:

```
dn: uid=Marcia Garza,ou=People,dc=example,dc=com
changetype: modify
replace: telephonenumber
telephonenumber: +1 408 555 6465
-
add: facsimiletelephonenumber
facsimiletelephonenumber: +1 408 222 4444
-
add: l
l: Sunnyvale
```

The following command specifies the host name (`-h`), port (`-p`), bind DN (`-D`), bind password (`-w`), reads the modifications from the file (`-f`) and processes the changes to the database. For Windows platforms, specify the path to your LDIF file (for example,`-f \temp\mod_attribute.ldif`):

```
$ ldapmodify -h hostname -p 1389 -D "cn=Directory Manager" -j pwd-file \
-f /usr/local/mod_attribute.ldif
```

#### *Example A–131 Deleting an Attribute from the Command Line*

The following command specifies the host name (`-h`), port (`-p`), bind DN (`-D`), bind password (`-w`), and deletes the `facsimiletelephonenumber` attribute for an entry. Because the command is run from the command line, enter the `dn`, `changetype`, modification operation, and then press Control-D (UNIX, Linux) or Control-Z (Windows) to process it:

```
$ ldapmodify -h hostname -p 1389 -D "cn=Directory Manager" -j pwd-file
dn: uid=Marcia Garza,ou=People,dc=example,dc=com
changetype: modify
delete: facsimiletelephonenumber
(Press Control-D for Unix, Linux)
(Press Control-Z for Windows)
```

#### *Example A–132 Deleting an Entry from the Command Line*

The following command specifies the hostname (`-h`), port (`-p`), bind DN (`-D`), bind password (`-w`), and deletes the entry. Because the command is run from the command line, enter the `dn`, `changetype`, and then press Control-D (UNIX, Linux) or Control-Z (Windows) to process it:

```
$ ldapmodify -h hostname -p 1389 -D "cn=Directory Manager" -j pwd-file
dn: uid=Marcia Garza,ou=People,dc=example,dc=com
changetype: delete
(Press Control-D for Unix, Linux)
(Press Control-Z for Windows)
```

#### *Example A–133 Using `ldapmodify` with Server Authentication*

The following command uses the `-P` SSL option to perform a modify with server authentication. The command specifies the host name (`-h`), SSL port (`-p`), base DN (`-b`), the bind DN (`-D`), the bind password (`-w`), trust store file path (`-P`), and LDIF file (`-f`) that contains the changes. For Windows platforms, specify the paths for the

modification file (for example, `-f \temp\myldif.ldif`) and trust store file (for example, `-P \temp\certs\cert.db`):

```
$ ldapmodify -h hostname -p 1636 -f /home/local/myldif.ldif \
-D "cn=Directory Manager" -j pwd-file \
-P /home/scarter/certs/cert.db
```

***Example A–134 Using `ldapmodify` with Client Authentication***

The following command uses the `-P` SSL option to perform a modify using client authentication. The command uses SSL (`-Z`) with the SSL port (`-p`) and specifies the trust store file path (`-P`), the certificate nickname (`-N`), the keystore file path (`-K`), the keystore password (`-W`) and the LDIF file (`-f`) that contains the changes. For Windows platforms, specify the paths for the modification file (for example, `-f \temp\myldif.ldif`), trust store file (for example, `-P \certs\cert.db`), and the keystore file (for example, `-K \security\key.db`):

```
$ ldapmodify -h hostname -p 1636 -f /home/local/myldif.ldif \
-Z -P /home/scarter/security/cert.db -N "sccert" \
-K /home/scarter/security/key.db -W keypassword
```

### A.4.3.10 Exit Codes

An exit code of 0 indicates that the operation completed successfully. A nonzero exit code indicates that an error occurred during processing.

### A.4.3.11 Using a CLI Properties File

The directory server supports the use of a properties file that passes in any default option values used with the `ldapmodify` command. The properties file is convenient when working in different configuration environments, especially in scripted or embedded applications. See Section A.1.2, "Using a Properties File With Server Commands" for more information.

The following options can be stored in a properties file:

- `assertionFilter`
- `bindDN`
- `bindPassword`
- `bindPasswordFile`
- `certNickname`
- `continueOnError`
- `control`
- `dry-run`
- `filename`
- `hostname`
- `keyStorePassword`
- `keyStorePasswordFile`
- `keyStorePath`
- `ldapVersion`

- `port`
- `postReadAttributes`
- `preReadAttributes`
- `proxyAs`
- `reportAuthzID`
- `saslOption`

  SASL is not supported for a proxy server instance.
- `trustAll`
- `trustStorePassword`
- `trustStorePasswordFile`
- `trustStorePath`
- `useSASLExternal`

  SASL is not supported for a proxy server instance.
- `useSSL`
- `useStartTLS`
- `verbose`

Entries in the properties file have the following format:

```
toolname.propertyname=propertyvalue
```

For example:

```
ldapmodify.ldapport=12345
```

#### A.4.3.12 Location

- UNIX and Linux: *INSTANCE_DIR*/OUD/bin/ldapmodify
- Windows: *INSTANCE_DIR*\OUD\bat\ldapmodify.bat

#### A.4.3.13 Related Commands

- Section A.4.1, "ldapcompare"
- Section A.4.2, "ldapdelete"
- Section A.4.4, "ldappasswordmodify"
- Section A.4.5, "ldapsearch"

### A.4.4 ldappasswordmodify

The `ldappasswordmodify` command modifies LDAP passwords.

#### A.4.4.1 Synopsis

`ldappasswordmodify` *options*

### A.4.4.2 Description

The `ldappasswordmodify` command can be used to change or reset user passwords with the LDAP password modify extended operation as defined in RFC 3062 (`http://www.ietf.org/rfc/rfc3062.txt`).

Using this mechanism for changing user passwords offers a number of benefits over a simple LDAP modify operation targeted at the password attribute, including the following:

- Changing one's own password. The command allows a user to change his own password even after it has expired, provided that this capability is allowed in that user's password policy.
- Supplying clear-text password. The command provides a mechanism for supplying the clear-text version of the current password for further validation of the user's identity.
- Using authorization ID. When changing a user's password, the user can be specified by using an authorization ID (prefixed by `dn:` or `u:`) in addition to a full DN.
- Generating passwords. If a new password is not provided, then the server can generate one for the user, provided that this capability is allowed in that user's password policy.

### A.4.4.3 Options

The `ldappasswordmodify` command accepts an option in either its short form (for example, `-D` *bindDN*) or its long form equivalent (for example, `--bindDN` *bindDN*).

### A.4.4.4 Command Options

`-a, --authzID` ***authzID***
Specify an authorization ID for the user whose password is to be changed. The authorization ID can be in the form `dn:` followed by the DN of the target user, or `u:` followed by the user name of the target user. If this option is not provided, then no authorization ID will be included in the request and the password for the authenticated user will be changed. This option must not be used in conjunction with the `--provideDNForAuthzID` option.

`-A, --provideDNForAuthzID`
Indicate that the bind DN should be used as the authorization ID for the password modify operation. This option must not be used in conjunction with the `--authzID` option.

`-c, --currentPassword` ***currentPassword***
Specify the current password for the user. It must not be used in conjunction with `--currentPasswordFile`. The user's current password must be provided in cases in which no authentication is performed, for example, if a user is trying to change his password after it has already expired. The password might also be required by the server based on the password policy configuration even if a bind password was provided.

`-C, --currentPasswordFile` ***currentPasswordFile***
Read the current password from the specified file. It must not be used in conjunction with `--currentPassword`. The user's current password must be provided in cases in which no authentication is performed, for example, if a user is trying to change his password after it has already expired. The password might also be required by the

server based on the password policy configuration even if a bind password was provided.

`-J, --control` ***controloid*[:*criticality*[:*value*|::*b64value*|:<*fileurl*]]**
Perform a search with the specified control in search requests sent to the directory server. This option makes it possible to include arbitrary request controls that the client cannot directly support. The value for this option must be in the form:

*oid*[:*criticality*[:*value*|::*b64value*|:<*fileurl*]]

The elements of this value include:

- **oid.** Use the OID for the control. For certain types of controls, a text name may be used instead of the numeric OID (for search operations, this includes `managedsait` for the manage DSA IT control). This element is required. Human-readable names can be used in place of the OID to reference controls that do not require values using the -J or `control` option. These OID names are the following:

  `accountusable` or `accountusability` — Use in place of the Account Usability Request Control OID: 1.3.6.1.4.1.42.2.27.9.5.8 (no value).

  `authzid` or `authorizationidentity` — Use in place of the Authorization Identity Request Control OID: 2.16.840.1.113730.3.4.16 (no value).

  `effectiverights` — Use in place of the Get Effective Rights Control OID: 1.3.6.1.4.1.42.2.27.9.5.2 (value = authorization ID).

  `managedsait` — Use in place of the Manage DSA IT Control OID: 2.16.840.1.113730.3.4.2 (no value).

  `noop` or `no-op` — Use in place of the LDAP No-op Control OID: 1.3.6.1.4.1.4203.1.10.2 (no value).

  `pwpolicy` or `password policy` — Use in place of the Password Policy Request Control OID: 1.3.6.1.4.1.42.2.27.8.5.1 (no value).

  `subtreedelete` or `treedelete` — Use in place of the Subtree Delete Request Control OID: 1.2.840.113556.1.4.805 (no value).

- **criticality.** If `true`, the control should be marked critical (meaning that the directory server should not process the operation unless it can meet the requirements of this control). If `false`, the control should not be marked critical. If this subcommand is not provided, then the control is not marked critical.

- **value.** Specifies the value for the control. This form should only be used if the value can be expressed as a string. It must not be used in conjunction with either the `::`*b64value* or `:<` *fileurl* forms. If none of these subcommands is present, then the control will not have a value.

- **b64value.** Specifies the value for the control in base64-encoded form. This subcommand must not be used in conjunction with either the `:`*value* or `:<` *fileurl* forms. If none of these subcommands is present, then the control will not have a value.

- **fileurl.** Specifies a URL that references a file from which the value of the control should be taken. It must not be used in conjunction with either the `:`*value* or `::`*b64value* forms. If none of these subcommands is present, then the control will not have a value.

For example, the value `1.3.6.1.4.1.42.2.27.9.5.2:true:dn:uid=dmiller,ou=people,dc=example,dc=com` will include a critical control with an OID of

1.3.6.1.4.1.42.2.27.9.5.2, marked as critical (true), and with a string value for the authorization ID dn:uid=dmiller,ou=people,dc=example,dc=com. Or, you can use the OID names:
effectiverights:true:dn:uid=dmiller,ou=people,dc=example,dc=com.

-n, --newPassword ***newPassword***
Specify the new password that should be assigned to the target user. This option must not be used in conjunction with --newPasswordFile. If neither of these options is provided, then the server will automatically generate a new password for the user, provided that a password generator is configured in the user's password policy.

-N, --newPasswordFile ***newPasswordFile***
Read the new password from the specified file that should be assigned to the target user. This option must not be used in conjunction with --newPassword. If neither of these options is provided, then the server will automatically generate a new password for the user, provided that a password generator is configured in the user's password policy.

### A.4.4.5 LDAP Connection Options

--certNickname ***nickname***
Use the certificate for certificate-based client authentication.

-D, --bindDN ***bindDN***
Use the DN when binding to the directory server through simple authentication. If this option is not provided, then the --authzID option must be used to specify the authorization ID for the target user, and either the --currentPassword or --currentPasswordFile option must be provided to specify the current password for the user. (This mode of use will be required for users to change their passwords after the passwords have expired.)

-h, --hostname ***address***
Contact the directory server on the specified host name or IP address. If it is not provided, then a default address of localhost will be used.

-j, --bindPasswordFile ***bindPasswordFile***
Use the bind password in the specified file when authenticating to the directory server. The option is used for simple authentication, as well as for password-based SASL mechanisms such as CRAM-MD5, DIGEST-MD5, and PLAIN. It is not required if no authentication is to be performed. This option must not be used in conjunction with --bindPassword.

SASL is not supported for a proxy server instance.

-K, --keyStorePath ***keyStorePath***
Use the client keystore certificate in the specified path for secure communication when using the SSL or the StartTLS extended operation. This option should only be necessary if the client needs to present a certificate to the directory server, for example, when using SASL EXTERNAL authentication.

SASL is not supported for a proxy server instance.

-o, --saslOption ***name***=***value***
Use the specified option when performing SASL authentication. Multiple SASL options can be provided by using this option multiple times, once for each option. See Section 20.6, "Using SASL Authentication" for more information.

`-p, --port` ***port***
Contact the directory server at the specified port. If this option is not provided, then a default port of `389` will be used.

`-P, --trustStorePath` ***trustStorePath***
Use the client trust store certificate in the specified path for secure communication when using the SSL or the StartTLS extended operation. This option is not needed if `--trustAll` is used, although a trust store should be used when working in a production environment.

`-q, --useStartTLS`
Use the StartTLS extended operation when communicating with the directory server. This option must not be used in conjunction with `--useSSL`.

`--trustStorePassword` ***trustStorePassword***
Use the password needed to access the certificates in the client trust store. This option is only required if `--trustStorePath` is used and the specified trust store requires a password in order to access its contents (which most trust stores do not require). This option must not be used in conjunction with `--trustStorePasswordFile`.

`-u, --keyStorePasswordFile` ***keyStorePasswordFile***
Use the password in the specified file to access the certificates in the client keystore. This option is only required if `--keyStorePath` is used. This option must not be used in conjunction with `--keyStorePassword`.

`-U, --trustStorePasswordFile` ***trustStorePasswordFile***
Use the password in the specified file to access the certificates in the client trust store. This option is only required if `--trustStorePath` is used and the specified trust store requires a password in order to access its contents (most trust stores do not require this). This option must not be used in conjunction with `--trustStorePassword`.

`-w, --bindPassword` ***bindPassword***
Use the bind password when authenticating to the directory server. This option can be used for simple authentication as well as password-based SASL mechanisms. This option must not be used in conjunction with `--bindPasswordFile`. To prompt for the password, type `-w -`.

SASL is not supported for a proxy server instance.

`-W, --keyStorePassword` ***keyStorePassword***
Use the password needed to access the certificates in the client keystore. This option is only required if `--keyStorePath` is used. This option must not be used in conjunction with `--keyStorePasswordFile`.

`-X, --trustAll`
Trust any certificate that the directory server might present during SSL or StartTLS negotiation. This option can be used for convenience and testing purposes, but for security reasons a trust store should be used to determine whether the client should accept the server certificate.

`-Z, --useSSL`
Use the Secure Sockets Layer when communicating with the directory server. If SSL is to be used, then the `--port` option should be used to specify the server's secure port.

### A.4.4.6 Command Input/Output Options

`--noPropertiesFile`
Indicate that a properties file will not be used to get the default command-line options.

`--propertiesFilePath` ***propertiesFilePath***
Specify the path to the properties file that contains the default command-line options.

### A.4.4.7 General Options

`-?, -H, --help`
Display command-line usage information for the command and exit without making any attempt to run the command.

`-V, --version`
Display the version information for the directory server.

### A.4.4.8 Examples

The following examples show how to use the `ldappasswordmodify` command.

#### *Example A–135 Modifying Your User Password*

The following command connects to the host (`-h`) using port `1389` (`-p`), specifies the authorization ID `uid=abergin` (`-a`) of an administrator, specifies the user's current password file (`-C`), and changes it with a new one specified in a new password file (`-N`). For Windows platforms, use the file paths where your current and new passwords exist, respectively. For example, use `-C \temp\currentPasswordFile` and `-N \temp\newPasswordFile`.

```
$ ldappasswordmodify -h hostname -p 1389 \
-a "dn:uid=abergin,ou=People,dc=example,dc=com" \
-C /tmp/currentPasswordFile -N /tmp/newPasswordFile

The LDAP password modify operation was successful
```

#### *Example A–136 Modifying and Generating a Password for Another User*

The following command connects to the host (`-h`) using port `1389` (`-p`), specifies the bind DN (`-D`), specifies the bind password file (`-j`), and modifies and generates a password for another user (`-a`) connecting over simple authentication. For Windows platforms, specify the file where the bind password file resides, for example, `-j \temp\bindPasswordFile`.

```
$ ldappasswordmodify -h hostname -p 1389 \
-D "cn=Directory Manager" -j /tmp/bindPasswordFile \
-a "dn:uid=abergin,ou=People,dc=example,dc=com"

The LDAP password modify operation was successful
Generated Password:  blb44hjm
```

#### *Example A–137 Modifying a Password for Another User*

The following command connects to the host (`-h`) using port `1389` (`-p`), specifies the bind DN (`-D`), specifies the bind password file (`-j`), and modifies the password with a new one (`-N`) for another user (`-a`) connecting over simple authentication. For Windows platforms, specify the bind password file (for example, `-j`

\temp\bindPasswordFile) and the new password file (for example, -N \temp\newPassword).

```
$ ldappasswordmodify -h hostname -p 1389 \
-D "cn=Directory Manager" -j /tmp/bindPasswordFile \
-a "dn:uid=abergin,ou=People,dc=example,dc=com" -N /tmp/newPassword

The LDAP password modify operation was successful
```

#### A.4.4.9 Exit Codes

An exit code of 0 indicates that the operation completed successfully. A nonzero exit code indicates that an error occurred during processing.

#### A.4.4.10 Using a CLI Properties File

The directory server supports the use of a properties file that passes in any default option values used with the `ldappasswordmodify` command. The properties file is convenient when working in different configuration environments, especially in scripted or embedded applications. See Section A.1.2, "Using a Properties File With Server Commands" for more information.

The following options can be stored in a properties file:

- `authzID`
- `bindDN`
- `bindPassword`
- `bindPasswordFile`
- `currentPassword`
- `currentPasswordFile`
- `control`
- `hostname`
- `keyStorePassword`
- `keyStorePasswordFile`
- `keyStorePath`
- `newPassword`
- `newPasswordFile`
- `port`
- `provideDNForAuthzID`
- `trustAll`
- `trustStorePassword`
- `trustStorePasswordFile`
- `trustStorePath`
- `useSSL`
- `useStartTLS`

Entries in the properties file have the following format:

```
toolname.propertyname=propertyvalue
```

For example:

```
ldappasswordmodify.ldapport=12345
```

#### A.4.4.11 Location

- UNIX and Linux: *INSTANCE_DIR*`/OUD/bin/ldappasswordmodify`
- Windows: *INSTANCE_DIR*`\OUD\bat\ldappasswordmodify.bat`

#### A.4.4.12 Related Commands

- Section A.4.1, "ldapcompare"
- Section A.4.2, "ldapdelete"
- Section A.4.3, "ldapmodify"
- Section A.4.5, "ldapsearch"

### A.4.5 ldapsearch

The `ldapsearch` command searches directory server entries.

#### A.4.5.1 Synopsis

`ldapsearch` [*options*] [*filter*] [*attributes*]

#### A.4.5.2 Description

The `ldapsearch` command can be used to enter a search request to the directory server. The command opens a connection to the directory server, binds to it, and returns all entries that meet the search filter and scope requirements starting from the specified base DN. It can also be used to test other components of the directory server, such as authentication, control, and secure communication mechanisms.

If the `--filename` option is used to specify a file containing one or more search filters, then the search filter should not be included as an option. All trailing options will be interpreted as requested attributes.

If an entry has non-ASCII characters for its name and attributes, such as *sn*, *givenName*, *uid*, and *title*, the non-ASCII characters returned by running the `ldapsearch` command are suppressed while printing. You have to run the `base64` command to decode the Base64-encoded string.

If no specific attributes are requested, then all user attributes (that is, all non-operational attributes) are returned. If one or more attribute names are listed, then only those attributes are included in the entries that are returned.

#### A.4.5.3 Before You Begin

Many UNIX and Linux operating systems provide an installed version of common LDAP client commands, such as `ldapsearch`, `ldapmodify`, and `ldapdelete` in the `/usr/bin` directory. You can check if a version is on your system by entering the command: `which ldapsearch`. If the command returns a value (seen below), you will need to update your `$PATH` to directory server installation directory or create an alias to the directory server instance.

```
$ which ldapsearch (Unix/Linux)
/usr/bin/ldapsearch
```

### A.4.5.4 Options

The `ldapsearch` command accepts an option in either its short form (for example, `-b` *baseDN*) or its long form equivalent (for example, `--baseDN` *baseDN*).

### A.4.5.5 Command Options

`-a, --dereferencePolicy dereferencePolicy`
Specify the dereference alias policy during a search. *Dereference alias* allows you to set an entry to point to another object. If this option is not provided, then a default of `never` will be used. Possible values are the following:

- `always` — Dereference aliases both when finding the base DN and when searching below it.
- `find` — Dereference alias when finding the base DN.
- `never` — Never dereference aliases (default).
- `search` — Dereference aliases when searching below the base DN but not when finding the base DN.

`-A, --typesOnly`
Perform a search to include attribute names in matching entries but not the attribute values. If this option is not provided, then both attribute names and values will be included in the matching entries.

`--assertionFilter filter`
Perform a search using the LDAP assertion control (as defined in RFC 4528 (http://www.ietf.org/rfc/rfc4528.txt)) to indicate that the operation should only be processed if the assertion contained in the provided filter is true.

`-b, --baseDN baseDN`
Specify the base DN to use for the search operation. If a file containing multiple filters is provided using the `--filename` option, then this base DN will be used for all of the searches. This is a required option. If a base DN with a null value (`""`) is specified, the server returns the root DSE entry.

`-c, --continueOnError`
Continue processing even if an error occurs. Use this option when you use multiple search filters in a file (`--filename`). If an error occurs during processing, the server will continue processing the next search filter. Otherwise the command will exit before all searches have been completed.

`-C, --persistentSearch`
`ps[:changetype[:changesonly[:entrychangecontrols]]]`
Use the persistent search control (as defined in draft-ietf-ldapext-psearch.txt (https://opends.dev.java.net/public/standards/draft-ietf-ldapext-psearch.txt)) in the search request to obtain information about changes that are made to entries that match the provided search criteria. The value for this option must be in the form:

```
ps[:changetype[:changesonly [:entrychangecontrols]]]
```

The elements of this value include:

- `ps` — Required operator.

- `changetype` — Indicates the types of changes for which the client wants to receive notification. It can be any of `add`, `del`, `mod`, or `moddn`, or it can be `all` to register for all change types, or it can be a comma-separated list to register for multiple specific change types. If this element is not provided, then it will default to including `all` change types.
- `changesonly` — If `true`, the client is only notified of changes that occur to matching entries after the search is registered. If `false`, the directory server sends all existing entries in the directory server that match the provided search criteria. If this element is not provided, then it will default to only returning entries for updates that occurred since the search was registered.
- `entrychangecontrols` — If `true`, the directory server includes the entry change notification control in entries sent to the client as a result of changes. If `false`, the entry change notification control is not included. If this element is not provided, then it will default to including the entry change notification controls.

For example, the value `ps:add,del:true:true` returns only entries matching the search criteria that have been added or deleted since the time that the persistent search was registered, and those entries will include entry change notification controls.

`--countEntries`
Display the total number of matching entries returned by the directory server. If the `--filename` option is used to specify the path to a file containing multiple search filters, the total number of matching entries for all searches is displayed.

`-e, --getEffectiveRightsAttribute attribute`
Return the effective rights on the specified attribute. This option can be used to specify attributes that would not normally appear in the search results for the entry. For example, use this option to determine if a user has permission to add an attribute that does not currently exist in the entry. The `-e` option requires the `--getEffectiveRightsAuthzid` or `-g` option.

`-f, --filename filename`
Specify the path to a file that contains one or more filters to use when processing the search operation. If the file contains multiple filters, the file should be structured with one filter per line. The searches will be performed using the same connection to the directory server in the order that they appear in the filter file. If this option is used, any trailing options will be treated as separate attributes. Otherwise the first trailing option must be the search filter.

`-g, --getEffectiveRightsAuthzid authzid`
Display the effective rights of the user binding with the given *authzid*. This option can be used with the `-e` option but cannot be used with the `-J` option.

`-G, --virtualListView before:after:index:count|before:after:value`
Retrieve the virtual list view displaying a portion of the total search results. Use one of two patterns to specify the size of the virtual list view:

- `before:after:index:count` — Return the target entry and the specified number of entries *before* the target entry and *after* the target entry. The target entry depends on the `index` and the `count` options. The `count` option can take the following values:

  **count=0.** The target entry is the entry at the specified *index* position, starting from 1 and relative to the entire list of sorted results.

  **count=1.** The target entry is the first entry in the list of sorted results.

**count>1.** The target entry is the first entry in the portion of the list represented by the fraction *index*/*count*. To target the last result in the list, use an *index* option greater than the *count* option.

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.

- `before:after:value` — Return the target entry and specified number of entries before and after the target entry. The target entry is the first entry in the sorted results whose sort attribute is greater than or equal to the specified value.

  For example, `-G 5:10:johnson -S sn` returns 16 entries in alphabetical order from the surname attribute: `5` less than `johnson`, the entry equal to or following `johnson`, and the `10` entries after `johnson`.

`-J, --control controloid[:criticality[:value|::b64value|:<filePath]]`

Perform a search with the specified control in search requests sent to the directory server. This option makes it possible to include arbitrary request controls that the client cannot directly support. The value for this option must be in the form:

`oid[:criticality[:value|::b64value|:<filePath]]`

The elements of this value include:

- **oid.** Use the OID for the control. For certain types of controls, a text name may be used instead of the numeric OID (for search operations, this includes `managedsait` for the manage DSA IT control). This element is required. Human-readable names can be used in place of the OID to reference controls that do not require values using the -J or `control` option. These OID names are the following:

  `accountusable` or `accountusability` — Use in place of the Account Usability Request Control OID: 1.3.6.1.4.1.42.2.27.9.5.8 (no value).

  `authzid` or `authorizationidentity` — Use in place of the Authorization Identity Request Control OID: 2.16.840.1.113730.3.4.16 (no value).

  `effectiverights` — Use in place of the Get Effective Rights Control OID: 1.3.6.1.4.1.42.2.27.9.5.2 (value = authorization ID).

  `managedsait` — Use in place of the Manage DSA IT Control OID: 2.16.840.1.113730.3.4.2 (no value).

  `noop` or `no-op` — Use in place of the LDAP No-op Control OID: 1.3.6.1.4.1.4203.1.10.2 (no value).

  `pwpolicy` or `password policy` — Use in place of the Password Policy Request Control OID: 1.3.6.1.4.1.42.2.27.8.5.1 (no value).

  `subtreedelete` or `treedelete` — Use in place of the Subtree Delete Request Control OID: 1.2.840.113556.1.4.805 (no value).

- **criticality.** If `true`, the control should be marked critical (meaning that the directory server should not process the operation unless it can meet the requirements of this control). If `false`, the control should not be marked critical. If this subcommand is not provided, then the control is not marked critical.

- **value.** Specifies the value for the control. This form should only be used if the value can be expressed as a string. It must not be used in conjunction with either the `::`*b64value* or `:<` *fileurl* forms. If none of these subcommands is present, then the control will not have a value.

- **b64value.** Specifies the value for the control in base64-encoded form. This subcommand must not be used in conjunction with either the `:`*value* or `:<` *fileurl* forms. If none of these subcommands is present, then the control will not have a value.
- **fileurl.** Specifies a URL that references a file from which the value of the control should be taken. It must not be used in conjunction with either the `:`*value* or `::`*b64value* forms. If none of these subcommands is present, then the control will not have a value.

For example, the value `1.3.6.1.4.1.42.2.27.9.5.2:true:dn:uid=dmiller,ou=people,dc=example,dc=com` will include a critical control with an OID of `1.3.6.1.4.1.42.2.27.9.5.2`, marked as critical (true), and with a string value for the authorization ID `dn:uid=dmiller,ou=people,dc=example,dc=com`. Or, you can use the OID names: `effectiverights:true:dn:uid=dmiller,ou=people,dc=example,dc=com`.

`-l, --timeLimit numSeconds`
Set the maximum length of time, in seconds, that the directory server should spend processing any search request. If this option is not provided, no time limit is requested by the client. Note that the directory server can enforce a lower time limit than the one that is requested by the client.

`--matchedValuesFilter filter`
Use the LDAP matched values control (as defined in RFC 3876 (http://www.ietf.org/rfc/rfc3876.txt)) to indicate that only attribute values matching the specified filter should be included in the search results. This option can be provided multiple times to specify multiple matched values filters.

`-n, --dry-run`
Run in `no-op` mode. That is, report what should happen but do not actually perform any searches or communicate with the server in any way.

`-s, --searchScope scope`
Set the scope for the search operation. The scope value must be one of the following:

- `base` — Search only the entry specified by the `--baseDN` or `-b` option.
- `one` — Search only the entry specified by the `--baseDN` or `-b` option and its immediate children.
- `sub` or `subordinate` — Search the subtree whose base is the entry specified by the `--baseDN` or `-b` option. This is the default option when the `--searchScope` is not provided.

`-S, --sortOrder sortOrder`
Sort the results before returning them to the client. The sort order is a comma-delimited list of sort keys, where each sort key consists of the following elements:

- `+/- (plus or minus sign)` — Indicates that the sort should be in ascending (`+`) or descending (`-`) order. If this element is omitted, then the sort will be in ascending order.
- `attribute name` — The name of the attribute to use when sorting the data. This element must always be provided.
- `name` or `OID Matching Rule` — An optional colon followed by the name or OID of the matching rule to use to perform the sort. If this element is not

provided, then the default ordering matching rule for the specified attribute type will be used. For example, the sort order string `sn,givenName` sorts entries in ascending order first by `sn` and then by `givenName`. Alternately, the value `--modifyTimestamp` will cause the results to be sorted with the most recent values first.

`--simplePageSize numEntries`
Use the Simple Paged Results control with the given page size.

`--subEntries`
Use the subentries control to specify that subentries are visible, and normal entries are not.

`-Y, --proxyAsauthzID`
Use the Proxied Authorization Control to specify the identity of the user for whom the operations should be performed. This will use version 2 of the Proxied Authorization Control as defined in RFC 4370 (http://www.ietf.org/rfc/rfc4370.txt). The value of the option should be an authorization ID in the form `dn:` followed by the DN of the target user (for example, `dn:uid=john.doe,ou=People,dc=example,dc=com`), or `u:` followed by the user name (for example, `u:john.doe`). If this option is not provided, proxied authorization is not used.

`-z, --sizeLimit numEntries`
Set the maximum number of matching entries that the directory server should return to the client. If this option is not provided, then there will be no maximum requested by the client. Note that the directory server can enforce a lower size limit than the one requested by the client.

### A.4.5.6 LDAP Connection Options

`-D, --bindDN bindDN`
Use the bind DN to authenticate to the directory server. This option is used when performing simple authentication. The default value for this option is `cn=Directory Manager`. It is not required when using SASL authentication or if no authentication is to be performed.

`-E, --reportAuthzID`
Use the authorization identity request control (as defined in RFC 3829 (http://www.ietf.org/rfc/rfc3829.txt)) in the bind request so that the directory server returns the corresponding authorization ID to the client when authentication has completed. (The line containing the authorization ID will be prefixed with a `#` character, making it a comment if the output is to be interpreted as an LDIF.)

`-h, --hostname address`
Contact the directory server on the specified host name or IP address. If it is not provided, then a default address of `localhost` will be used.

`-j, --bindPasswordFile bindPasswordFile`
Use the bind password in the specified file when authenticating to the directory server. The option is used for simple authentication, as well as for password-based SASL mechanisms such as CRAM-MD5, DIGEST-MD5, and PLAIN. It is not required if no authentication is to be performed. This option must not be used in conjunction with `--bindPassword`.

SASL is not supported for a proxy server instance.

`-K, --keyStorePath keyStorePath`
Use the client keystore certificate in the specified path for secure communication when using the SSL or the StartTLS extended operation. This option should only be necessary if the client needs to present a certificate to the directory server, for example, when using SASL EXTERNAL authentication.

SASL is not supported for a proxy server instance.

`-N, --certNickName certNickName`
Use the specified certificate for certificate-based client authentication.

`-o, --saslOption name=value`
Use the specified option when performing SASL authentication. Multiple SASL options can be provided by using this option multiple times, once for each option. See Section 20.7, "Configuring SASL Authentication" for more information on using SASL authentication in clients.

SASL is not supported for a proxy server instance.

`-p, --port port`
Contact the directory server at the specified port. If this option is not provided, then a default port of `389` will be used.

`-P, --trustStorePath trustStorePath`
Use the client trust store certificate in the specified path for secure communication when using the SSL or the StartTLS extended operation. This option is not needed if `--trustAll` is used, although a trust store should be used when working in a production environment.

`-q, --useStartTLS`
Use the StartTLS Extended Operation extended operation when communicating with the directory server. This option must not be used in conjunction with `--useSSL`.

`-r, --useSASLExternal`
Use the SASL EXTERNAL mechanism for authentication, which attempts to identify the client by using an SSL certificate that it presents to the directory server. If this option is used, then the `--keyStorePath` option must also be provided to specify the path to the client keystore and either the `--useSSL` or the `--useStartTLS` option must be used to establish a secure communication channel with the server.

SASL is not supported for a proxy server instance.

`--trustStorePassword trustStorePassword`
Use the password needed to access the certificates in the client trust store. This option is only required if `--trustStorePath` is used and the specified trust store requires a password in order to access its contents (which most trust stores do not require). This option must not be used in conjunction with `--trustStorePasswordFile`.

`-u, --keyStorePasswordFile keyStorePasswordFile`
Use the password in the specified file to access the certificates in the client keystore. This option is only required if `--keyStorePath` is used. This option must not be used in conjunction with `--keyStorePassword`.

`--usePasswordPolicyControl`
Use the Password Policy Request Control in the bind request so that the directory server returns the corresponding result control in the bind response. This can be used to obtain information about any warnings or errors with regard to the state of the client's account.

`-U, --trustStorePasswordFile trustStorePasswordFile`
Use the password in the specified file to access the certificates in the client trust store. This option is only required if `--trustStorePath` is used and the specified trust store requires a password in order to access its contents (most trust stores do not require this). This option must not be used in conjunction with `--trustStorePassword`.

`-V, --ldapVersion version`
Set the LDAP protocol version that the client should use when communicating with the directory server. The value must be either `2` (for LDAPv2 communication) or `3` (for LDAPv3). If this option is not provided, then the client will use LDAPv3.

`-w, --bindPassword bindPassword`
Use the bind password when authenticating to the directory server. This option can be used for simple authentication as well as password-based SASL mechanisms. This option must not be used in conjunction with `--bindPasswordFile`. To prompt for the password, type `-w -`.

SASL is not supported for a proxy server instance.

`-W, --keyStorePassword keyStorePassword`
Use the password needed to access the certificates in the client keystore. This option is only required if `--keyStorePath` is used. This option must not be used in conjunction with `--keyStorePasswordFile`.

`-X, --trustAll`
Trust any certificate that the directory server might present during SSL or StartTLS negotiation. This option can be used for convenience and testing purposes, but for security reasons a trust store should be used to determine whether the client should accept the server certificate.

`-Z, --useSSL`
Use SSL when communicating with the directory server. If SSL is to be used, then the `--port` option should be used to specify the server's secure port.

### A.4.5.7 Command Input/Output Options

`--noPropertiesFile`
Indicate that a properties file will not be used to get the default command-line options.

`--propertiesFilePath propertiesFilePath`
Specify the path to the properties file that contains the default command-line options.

`-T, --dontWrap`
Do not wrap long lines when displaying matching entries. If this option is not provided, then long lines will be wrapped (in a manner compatible with the LDIF specification) to fit on an 80-column terminal.

`-v, --verbose`
Run in verbose mode, displaying process and diagnostic information on standard output.

### A.4.5.8 General Options

`-?, -H, --help`
Display command-line usage information for the command and exit without making any attempt to run the command.

`-V, --version`
Display the version information for the directory server.

### A.4.5.9 Examples

The following examples show how to use the `ldapsearch` command. For additional examples, see Section 17.4, "Searching Directory Data."

#### *Example A–138 Returning All Entries*

The following command returns all entries on the directory server. The command connects to the default port `1389` (`-p`) on the host (`-h`), specifies the base DN as `example.com` (`-b`), and returns all entries by using the search filter `(objectclass=*)`. Because the scope (`-s`) is not specified, the scope is set to the default value of `sub`, the full subtree of the base DN. Because no attributes are specified, the command returns all attributes and values.

```
$ ldapsearch -h hostname -p 1389 -b dc=example,dc=com "(objectclass=*)"

dn: dc=example,dc=com
objectClass: domain
objectClass: top
dc: example

dn: ou=Groups,dc=example,dc=com
objectClass: organizationalunit
objectClass: top
ou: Groups

dn: cn=Directory Administrators,ou=Groups,dc=example,dc=com
objectClass: groupofuniquenames
objectClass: top
ou: Groups
cn: Directory Administrators
uniquemember: uid=kvaughan, ou=People, dc=example,dc=com
uniquemember: uid=rdaugherty, ou=People, dc=example,dc=com
uniquemember: uid=hmiller, ou=People, dc=example,dc=com
```

#### *Example A–139 Returning Attributes Names but No Values*

The following command returns the attribute names (`-A`) but no values. The command connects to the default port `1389` (`-p`) on the host (`-h`), specifies the base DN as `dc=example,dc=com` (`-b`), matches all entries by using the search filter `objectclass=*`, and returns three (`-z`) entries. Using the `-A` option is a convenient way to check if an attribute is present in the database.

```
$ ldapsearch -h hostname -p 1389 -b dc=example,dc=com -A -z 3 "(objectclass=*)"

dn: dc=example,dc=com
objectClass
dc

dn: ou=Groups,dc=example,dc=com
objectClass
ou

dn: cn=Directory Administrators,ou=Groups,dc=example,dc=com
objectClass
ou
```

```
cn
uniquemember
```

#### *Example A–140 Returning Specific Attribute Values*

The following command returns a specific attribute and its value. The command connects to the port `1389` (`-p`) on the host (`-h`), specifies the base DN as `dc=example,dc=com` (`-b`), matches all entries by using the search filter `cn=Sam Carter`, and returns the value of the attribute, `telephonenumber`.

```
$ ldapsearch -h hostname -p 1389 -b dc=example,dc=com "(cn=Sam Carter)"
telephoneNumber

dn: uid=scarter,ou=People,dc=example,dc=com
telephonenumber: +1 408 555 4798
```

#### *Example A–141 Returning the Root DSE*

The root DSE is a special entry that provides information about the directory server's name, version, naming contexts, and supported features. You specify the root DSE by using a base DN with a null value (for example, `-b ""`) from which the directory server searches below all public naming contexts by default. You can override the null base DN default by specifying specific sets of base DNs with the `subordinate-base-dn` property by using the `dsconfig` command. The following example connects to the default port `1389` (`-p`) on the host (`-h`), specifies the root DSE as an empty base entry (`-b`), specifies the scope of the search to `base` (`-s`), matches all entries by using the search filter `objectclass=*`, and returns the directory server's root DSE information for supported controls:

```
$ ldapsearch -h hostname -p 1389 -b "" -s base "(objectclass=*)" supportedControl

dn:
supportedControl: 1.2.826.0.1.3344810.2.3
supportedControl: 1.2.840.113556.1.4.319
supportedControl: 1.2.840.113556.1.4.473
supportedControl: 1.2.840.113556.1.4.805
...
```

#### *Example A–142 Searching by Using Server Authentication*

The following command uses the SSL option to run a search with server authentication. The command specifies the host name (`-h`), SSL port `1636` (`-p`), base DN (`-b`), the bind DN (`-D`), the bind password (`-w`), trust store file path (`-P`), and the entity's given name. For Windows platforms, specify the paths for trust store file (for example, `-P \certs\cert.db`).

```
$ ldapsearch -h hostname -p 1636 -b "dc=example,dc=com" \
-D "uid=scarter,ou=people,dc=example,dc=com" -w bindPassword \
-P /home/scarter/certs/cert.db "(givenname=Sam)"
```

#### *Example A–143 Searching by Using Client Authentication*

The following command uses the SSL option to perform a search by using client authentication. The command uses SSL (`-Z`) with the SSL port (`-p`) and specifies the trust store file path (`-P`), the certificate nickname (`-N`), the keystore file path (`-K`), the keystore password (`-W`) and the entity's given name `(givenname=Sam)`. For

Windows platforms, specify the paths for the trust store file (for example, `-P \certs\cert.db`), and the keystore file (for example, `-K \security\key.db`):

```
$ ldapsearch -h hostname -p 1636 -b "dc=example,dc=com" \
-Z -P /home/scarter/security/cert.db -N "sccert" \
-K /home/scarter/security/key.db -W KeyPassword \
"(givenname=Sam)"
```

#### *Example A–144 Returning the Effective Rights of a User*

The following command returns the effective rights granted to a user, in addition to the user's attribute entries. Only a directory administrator can access this information for another user. The command specifies the host name (`-h`), port `1389` (`-p`), bindDN (`-D`), bindDN password (`-w`), base DN (`-b`), control spec option that includes the OID name `effectiverights` (alternately, you can enter the OID equivalent: `1.3.6.1.4.1.42.2.27.9.5.2`), search filter `objectclass=*`, and the `aclRights` attribute.

```
$ ldapsearch -h hostname -p 1389 -D "cn=Directory Manager" -j pwd-file \
-b dc=example,dc=com -J "1.3.6.1.4.1.42.2.27.9.5.2"  "(objectclass=*)" \
aclRights

dn: dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0

dn: ou=Groups, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0

dn: ou=People, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0

dn: cn=Accounting Managers,ou=groups,dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0

dn: cn=HR Managers,ou=groups,dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0

dn: uid=bjensen,ou=People, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0

dn: uid=cfuente, ou=People, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:1,proxy:0
```

#### *Example A–145 Returning the Schema*

The following command searches the `cn=schema` entry for the object classes and attributes defined on the directory instance. The command connects to the port `1389` (`-p`) on the host (`-h`), sets the scope of the search to `base` (`-s`), matches all entries by using the search filter `(objectclass=\*)` and returns the objectClass definitions in the schema entry, `cn=schema`. You can also use the + symbol to view the schema. Place it after the search filter.

```
$ ldapsearch -h hostname -p 1389 -b cn=schema -s base "(objectclass=*)"
objectClasses

dn: cn=schema
objectClasses: ( 2.5.6.0 NAME 'top' ABSTRACT MUST objectClass X-ORIGIN 'RFC 4512
' )
objectClasses: ( 2.5.6.1 NAME 'alias' SUP top STRUCTURAL MUST aliasedObjectName
```

```
X-ORIGIN 'RFC 4512' )
objectClasses: ( 2.5.6.2 NAME 'country' SUP top STRUCTURAL MUST c MAY ( searchGu
ide $ description ) X-ORIGIN 'RFC 4519' )
objectClasses: ( 2.5.6.3 NAME 'locality' SUP top STRUCTURAL MAY ( street $ seeAl
so $ searchGuide $ st $ l $ description ) X-ORIGIN 'RFC 4519' )
...
```

#### *Example A–146 Performing a Persistent Search*

The `ldapsearch` command provides an option to run a persistent search (`-C`) that keeps the connection open and displays the entries that matching the scope and filter whenever any changes (`add`, `delete`, `mod`, or `all`) occur. The command connects to the port `1389` (`-p`), sets the scope of the search to `base` (`-s`), and matches all entries by using the search filter `(objectclass=\*)`. You can quit out of the search by pressing `Control-C`.

```
$ ldapsearch -b dc=example,dc=com -p 1389 -D "cn=Directory Manager" \
-j pwd-file -C ps:add:true:true "(objectclass=*)"
```

#### *Example A–147 Viewing ACI Attributes*

The following command displays the access control instruction (ACI) attributes from the specified base DN. The command connects to the port `1389` (`-p`), sets the scope of the search to `base` (`-s`), matches all entries using the search filter `(objectclass=\*)` and specifies the `aci` attribute.

```
$ ldapsearch -p 1389 -D "cn=Directory Manager" -j pwd-file -b dc=example,dc=com \
-s base "(objectclass=*)" aci

dn: dc=example,dc=com
aci: (target ="ldap:///dc=example,dc=com")(targetattr h3.="userPassword")(version
3.0;acl "Anonymous read-search access";allow (read, search, compare)(userdn = "
ldap:///anyone");)
aci: (target="ldap:///dc=example,dc=com") (targetattr = "*")(version 3.0; acl "a
llow all Admin group"; allow(all) groupdn = "ldap:///cn=Directory Administrator
s,ou=Groups,dc=example,dc=com";)
```

#### *Example A–148 Viewing Monitoring Information*

The following command searches the cn=monitor entry for information on the activity on the directory server. The command specifies the host name (`-h`), port (`-p`), base DN (`-b`) for `cn=monitor`, authenticates using the bind DN (`-D`) and bind password (`-w`) and specifies the filter `(objectclass=\*)`.

```
$ ldapsearch --useSSL -X -h hostname -p 4444 -b cn=monitor -D "cn=Directory
Manager" \
-j pwd-file "(objectclass=*)"

dn: cn=monitor
objectClass: top
objectClass: extensibleObject
objectClass: ds-monitor-entry
currentTime: 20070803161832Z
startTime: 20070803132044Z
productName: Oracle Unified Directory
...
```

***Example A–149 Searching by Using a Properties File***

The directory server supports the use of a *properties file* that passes in any default option values used with the `ldapsearch` command. The properties file is convenient when working in different configuration environments, especially in scripted or embedded applications. See Section A.1.2, "Using a Properties File With Server Commands" for more information.

The following options can be stored in a properties file:

- `assertionFilter`
- `bindDN`
- `bindPassword`
- `bindPasswordFile`
- `certNickname`
- `continueOnError`
- `control`
- `countEntries`
- `dereferencePolicy`
- `dry-run`
- `dontWrap`
- `filename`
- `getEffectiveRightsAttribute`
- `getEffectiveRightsAuthzid`
- `hostname`
- `keyStorePassword`
- `keyStorePasswordFile`
- `keyStorePath`
- `ldapVersion`
- `matchedValuesFilter`
- `persistentSearch`
- `port`
- `proxyAs`
- `reportAuthzID`
- `saslOption`

  SASL is not supported for a proxy server instance.
- `searchScope`
- `simplePageSize`
- `sizeLimit`
- `sortOrder`
- `timeLimit`

- trustAll
- trustStorePassword
- trustStorePasswordFile
- trustStorePath
- typesOnly
- usePasswordPolicyControl
- useSASLExternal

  SASL is not supported for a proxy server instance.
- useSSL
- useStartTLS
- verbose
- virtualListView

### A.4.5.10 To Search by Using a Properties File

1. Create a properties file in any text editor. Here, save the file as `tools.properties`.

```
hostname=host
port=1389
bindDN=cn=Directory Manager
bindPassword=password
baseDN=dc=example,dc=com
searchScope=sub
sortOrder=givenName
virtualListView=0:2:1:0
```

2. Use `ldapsearch` with the `--propertiesFilePath` option. `$ldapsearch --propertiesFilePath tools.properties "(objectclass=*)"`

### A.4.5.11 Search Attributes

A number of special search attributes can also be used for various purposes, including the following:

*This symbol indicates that all user attributes should be included in the entries returned by the directory server.

```
$ ldapsearch -h hostname -p 1389 -b dc=example,dc=com "(objectclass=*)" *
```

+This symbol indicates that all operational attributes are to be included in the entries returned by the directory server. By default, no operational attributes will be returned. Note that even if this is specified, there might be some operational attributes that are not returned automatically for some reason for example, if an expensive computation is required to construct the value). On some systems, you might need to escape the + symbol by enclosing it in quotation marks, `"+"` or by using a backslash, `\+`.

```
$ ldapsearch -h hostname -p 1389 -b dc=example,dc=com "(objectclass=*)" "+"
```

`1.1`This indicates that no attribute values should be included in the matching entries. On some systems, you might need to escape the `1.1` character by enclosing it in quotation marks, `"1.1"`, or by using a backslash, `\1.1`.

```
$ ldapsearch -h hostname -p 1389 -b dc=example,dc=com "(objectclass=*)" "1.1"
```

`@_objectclass`_This indicates that all attributes associated with the specified object class should be included in the entries returned by the server. For example, `@person` indicates that the server should include all attributes associated with the `person` object class.

```
$ ldapsearch -h hostname -p 1389 -b dc=example,dc=com "(objectclass=*)" @person
```

#### A.4.5.12 Exit Codes

An exit code of 0 indicates that the operation completed successfully. A nonzero exit code indicates that an error occurred during processing.

#### A.4.5.13 Location

- UNIX and Linux: *INSTANCE_DIR*/OUD/bin/ldapsearch
- Windows: *INSTANCE_DIR*\OUD\bat\ldapsearch.bat

#### A.4.5.14 Related Commands

- Section A.4.1, "ldapcompare"
- Section A.4.2, "ldapdelete"
- Section A.4.3, "ldapmodify"
- Section A.4.4, "ldappasswordmodify"

# B

# Supported Controls and Operations

The Oracle Unified Directory supports a number of standard LDAP controls and extended operations. The following sections list these controls and extended operations.

- Section B.1, "Supported LDAP Controls"
- Section B.2, "Supported Extended Operations"

For information about using the LDAP controls, see Section 17.5.3, "Searching Using Controls".

## B.1 Supported LDAP Controls

A supported control is a mechanism for identifying the request control supported by the Oracle Unified Directory. The object identifier of these controls are listed in the `supportedControl` attribute of the server's root DSE.

Table B–1 lists the controls supported by the directory server.

If you have installed a proxy instance, refer to Table B–2, which lists the controls supported by the proxy as well as by the remote LDAP servers.

***Table B–1 LDAP Controls Supported by the Directory Server***

| OID | LDAP Control | RFC or draft |
|---|---|---|
| 1.2.826.0.1.3344810.2.3 | Matched Values Control | RFC3876 |
| 1.2.840.113556.1.4.319 | Page Results Control | RFC2696 |
| 1.2.840.113556.1.4.473 | Server-side Sort Control | RFC2891 |
| 1.2.840.113556.1.4.805 | Subtree Delete Control | Draft |
| 1.3.6.1.1.12 | Assertion Control | RFC4528 |
| 1.3.6.1.1.13.1 | LDAP Pre-read Control | RFC4527 |
| 1.3.6.1.1.13.2 | LDAP Post-read Control | RFC4527 |
| 1.3.6.1.4.1.26027.1.5.2 | Replication Repair Control | |
| 1.3.6.1.4.1.4203.1.10.2 | LDAP No-Op Control | Draft |
| 1.3.6.1.4.1.42.2.27.8.5.1 | Password Policy Control | Draft |
| 1.3.6.1.4.1.42.2.27.9.5.2 | Get Effective Rights Control | |
| 1.3.6.1.4.1.42.2.27.9.5.8 | Account Usability Control | |
| 1.3.6.1.4.1.42.2.27.9.5.9 | CSN (Change Number Control) | |

*Table B–1 (Cont.) LDAP Controls Supported by the Directory Server*

| OID | LDAP Control | RFC or draft |
|---|---|---|
| 1.3.6.1.4.1.4203.1.10.1 | LDAP Subentry Request Control | RFC3672 |
| 2.16.840.1.113730.3.4.4 | Password Expired Control | |
| 2.16.840.1.113730.3.4.5 | Password Expiration Warning Control | |
| 2.16.840.1.113730.3.4.12 | Proxy Authorization v1 Control | Draft |
| 2.16.840.1.113730.3.4.18 | Proxy Authorization v2 Control | RFC4370 |
| 2.16.840.1.113730.3.4.16 | Authorization Identity Request Control | RFC3829 |
| 2.16.840.1.113730.3.4.17 | Real Attributes Only Control | |
| 2.16.840.1.113730.3.4.19 | Virtual Attributes Only Control | |
| 2.16.840.1.113730.3.4.2 | ManageDsaIT | RFC3296 |
| 2.16.840.1.113730.3.4.3 | Persistent Search Control | Draft |
| 2.16.840.1.113730.3.4.9 | Virtual List View Control | Draft |
| 2.16.840.1.113894.1.8.21 | OID Search Count Request Control | |

*Table B–2 LDAP Controls Supported by the Proxy*

| OID | LDAP Control | RFC or draft | Supported by Proxy Workflow Element | Supported by Distribution Algorithm | Supported by remote ODSEE | Supported by remote Oracle Unified Directory server | Notes |
|---|---|---|---|---|---|---|---|
| 1.2.826.0.1.3344810.2.3 | Matched Values Control | RFC3876 | Yes | Yes | No | Yes | |
| 1.2.840.113556.1.4.319 | Page Results Control | RFC2696 | Yes | No | No | Yes | |
| 1.2.840.113556.1.4.473 | Server-side Sort Control | RFC2891 | Yes | No | Yes | Yes | Supported if all targeted entries are on the same remote LDAP server, and that remote LDAP server supports server-side LDAP control. |

*Table B–2 (Cont.) LDAP Controls Supported by the Proxy*

| OID | LDAP Control | RFC or draft | Supported by Proxy Workflow Element | Supported by Distribution Algorithm | Supported by remote ODSEE | Supported by remote Oracle Unified Directory directory server | Notes |
|---|---|---|---|---|---|---|---|
| 1.2.840.113556.1.4.805 | Subtree Delete Control | Draft | Yes | No | No | Yes | Supported if all targeted entries are on the same remote LDAP server, and that remote LDAP server supports subtree delete LDAP control. Not supported by the distribution algorithm because targeted entries can span multiple remote LDAP servers. |
| 1.3.6.1.1.12 | Assertion Control | RFC4528 | Yes | Yes | No | Yes | Supported if the remote LDAP server that hosts the targeted entry also supports assertion control. Therefore not supported in proxy configurations where all remote LDAP servers run Oracle Directory Server Enterprise Edition. |
| 1.3.6.1.1.13.1 | LDAP Pre-read Control | RFC4527 | Yes | Yes | Complies sufficiently for the proxy to work | Yes | Supported if the remote LDAP servers that host the targeted entries also support LDAP pre-read control.<br>Required for the global index catalog. In Oracle Unified Directory directory servers, this control must be enabled. |

***Table B–2 (Cont.) LDAP Controls Supported by the Proxy***

| OID | LDAP Control | RFC or draft | Supported by Proxy Workflow Element | Supported by Distribution Algorithm | Supported by remote ODSEE | Supported by remote Oracle Unified Directory directory server | Notes |
|---|---|---|---|---|---|---|---|
| 1.3.6.1.1.13.2 | LDAP Post-read Control | RFC4527 | Yes | Yes | No | Yes | Supported if the remote LDAP servers that hosts the targeted entries also support LDAP post-read control. Therefore not supported in proxy configurations where all remote LDAP servers run Oracle Directory Server Enterprise Edition.<br>In Oracle Unified Directory directory servers, this control must be enabled. |
| 1.3.6.1.4.1.26027.1.5.2 | Replication Repair Control | | No | No | No | Yes | Not supported by the proxy. To repair data inconsistency across remote LDAP servers, bypass the proxy and send the control directly to the remote LDAP servers running Oracle Unified Directory. For remote LDAP servers running Oracle Directory Server Enterprise Edition, refer to the `dsrepair` command in the Oracle Directory Server Enterprise Edition documentation. |
| 1.3.6.1.4.1.4203.1.10.2 | LDAP No-Op Control | Draft | Yes | Yes | No | Yes | Supported if the remote LDAP servers that host the targeted entries also support the LDAP no-op control. Therefore not supported in proxy configurations where all remote LDAP servers run Oracle Directory Server Enterprise Edition. |

*Table B–2 (Cont.) LDAP Controls Supported by the Proxy*

| OID | LDAP Control | RFC or draft | Supported by Proxy Workflow Element | Supported by Distribution Algorithm | Supported by remote ODSEE | Supported by remote Oracle Unified Directory directory server | Notes |
|---|---|---|---|---|---|---|---|
| 1.3.6.1.4.1.42.2.27.8.5.1 | Password Policy Control | Draft | Yes | Yes | Yes | Yes | |
| 1.3.6.1.4.1.42.2.27.9.5.2 | Get Effective Rights Control | | Yes | Yes | Yes | Yes | If this control is to be used by a configuration of the proxy where remote LDAP servers run Oracle Unified Directory, then the aclRights and aclRightsInfo controls need to be authorized in Oracle Unified Directory, if you have sufficient credentials. |
| 1.3.6.1.4.1.42.2.27.9.5.8 | Account Usability Control | | Yes | Yes | Yes | Yes | |
| 1.3.6.1.4.1.4203.1.10.1 | LDAP Subentry Request Control | RFC3672 | Yes | Yes | No | Yes | Supported if the remote LDAP servers that host the targeted entries also support the LDAP sub-entry control. |
| 2.16.840.1.113730.3.4.12 | Proxy Authorization v1 Control | Draft | Yes | Yes | Yes | Yes | Supported if the remote LDAP servers that host the targeted entries also support the proxy-authorization v1 control. If the proxy is configured in this control mode, the remote LDAP server must also support the get effective rights control. |
| 2.16.840.1.113730.3.4.18 | Proxy Authorization v2 Control | RFC4370 | Yes | Yes | Yes | Yes | Supported if the remote LDAP servers that host the targeted entries also support the proxy-authorization v2 control. If the proxy is configured in this control mode, the remote LDAP server must also support the get effective rights control. |

***Table B–2 (Cont.) LDAP Controls Supported by the Proxy***

| OID | LDAP Control | RFC or draft | Supported by Proxy Workflow Element | Supported by Distribution Algorithm | Supported by remote ODSEE | Supported by remote Oracle Unified Directory directory server | Notes |
|---|---|---|---|---|---|---|---|
| 2.16.840.1.113730.3.4.16 | Authorization Identity Request Control | RFC3829 | Yes | Yes | Yes | Yes | Supported if the remote LDAP server that hosts the target entry also supports the authorization identity request control. |
| 2.16.840.1.113730.3.4.17 | Real Attributes Only Control | | Yes | Yes | Yes | Yes | Supported if the remote LDAP servers that host the targeted entries also support the real attributes only control. |
| 2.16.840.1.113730.3.4.19 | Virtual Attributes Only Control | | Yes | Yes | Yes | Yes | Supported if the remote LDAP servers that host the targeted entries also support the virtual attributes only request control. |
| 2.16.840.1.113730.3.4.2 | ManageDsaIT | RFC3296 | Yes | Yes | Yes | Yes | |
| 2.16.840.1.113730.3.4.3 | Persistent Search Control | Draft | Yes | Yes | Yes | Yes | Supported if the remote LDAP servers that host the targeted entries also support the persistent search control. |
| 2.16.840.1.113730.3.4.9 | Virtual List View Control | Draft | Yes | No | Yes | Yes | Supported if all of the targeted entries are located on the same remote LDAP server, and that server supports virtual list view control. |
| 1.3.6.1.4.1.42.2.27.9.5.9 | CSN (Change Number Control) | | Yes | Yes | Yes | Yes | Dedicated to replication, appropriate for modifyRequest, delRequest, and modDNRequest LDAP messages. Required for the global index catalog. |

## B.2 Supported Extended Operations

A supported extension is a mechanism for identifying the extended operation supported by the Oracle Unified Directory. The object identifier of these extended operations are listed in the `supportedExtension` attribute of the server's root DSE.

The supported extensions for the Oracle Unified Directory include:

**1.3.6.1.1.8**
The cancel extended operation

**1.3.6.1.4.1.1466.20037**
The StartTLS extended operation

**1.3.6.1.4.1.26027.1.6.1**
The Password Policy State extended operation

**1.3.6.1.4.1.26027.1.6.2**
The Get Connection ID extended operation

**1.3.6.1.4.1.26027.1.6.3**
The Get Symmetric Key extended operation

**1.3.6.1.4.1.4203.1.11.1**
The Password Modify extended operation

**1.3.6.1.4.1.4203.1.11.3**
The "Who Am I?" extended operation

# C

# Standards and Specifications Supported by Oracle Unified Directory

This section describes the different standards and specifications supported by Oracle Unified Directory and contains the following topics:

- Section C.1, "RFCs Supported by Oracle Unified Directory"
- Section C.2, "Internet Drafts Supported by Oracle Unified Directory"
- Section C.3, "Other Specifications Supported by Oracle Unified Directory"

## C.1 RFCs Supported by Oracle Unified Directory

Table C–1 contains a list of the RFCs currently supported by Oracle Unified Directory. Oracle Unified Directory is continuously being updated to ensure that it conforms to the newer protocols.

***Table C–1 Supported RFCs***

| Number | Description |
|---|---|
| RFC 1274 | The COSINE and Internet X.500 Schema |
| RFC 1321 | The MD5 Message-Digest Algorithm |
| RFC 1777 | Lightweight Directory Access Protocol (v2) |
| RFC 1778 | The String Representation of Standard Attribute Syntaxes |
| RFC 1779 | A String Representation of Distinguished Names |
| RFC 2079 | Definition of an X.500 Attribute Type and an Object Class to Hold Uniform Resource Identifiers (URIs) |
| RFC 2222 | Simple Authentication and Security Layer (SASL) |
| RFC 2246 | The TLS Protocol |
| RFC 2246 | The TLS Protocol Version 1.0 |
| RFC 2247 | Using Domains in LDAP/X.500 Distinguished Names |
| RFC 2251 | Lightweight Directory Access Protocol (v3) |
| RFC 2252 | Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions |
| RFC 2254 | The String Representation of LDAP Search Filters |
| RFC 2255 | The LDAP URL Format |
| RFC 2256 | A Summary of the X.500(96) User Schema for use with LDAPv3 |

***Table C–1 (Cont.) Supported RFCs***

| Number | Description |
|---|---|
| RFC 2307 | An Approach for Using LDAP as a Network Information Service |
| RFC 2377 | Naming Plan for Internet Directory-Enabled Applications |
| RFC 2605 | Directory Server Monitoring MIB |
| RFC 2649 | An LDAP Control and Schema for Holding Operation Signatures |
| RFC 2696 | LDAP Control Extension for Simple Paged Results Manipulation |
| RFC 2713 | Schema for Representing Java(tm) Objects in an LDAP Directory |
| RFC 2714 | Schema for Representing CORBA Object References in an LDAP Directory |
| RFC 2739 | Calendar Attributes for vCard and LDAP |
| RFC 2788 | Network Services Monitoring MIB |
| RFC 2798 | Definition of the inetOrgPerson LDAP Object Class |
| RFC 2829 | Authentication Methods for LDAP |
| RFC 2830 | Lightweight Directory Access Protocol (v3): Extension for Transport Layer Security |
| RFC 2831 | Using Digest Authentication as a SASL Mechanism |
| RFC 2849 | The LDAP Data Interchange Format (LDIF) - Technical Specification |
| RFC 2891 | LDAP Control Extension for Server Side Sorting of Search Results |
| RFC 2926 | Conversion of LDAP Schemas to and from SLP Templates |
| RFC 3045 | Storing Vendor Information in the LDAP root DSE |
| RFC 3062 | LDAP Password Modify Extended Operation |
| RFC 3112 | LDAP Authentication Password Schema |
| RFC 3174 | US Secure Hash Algorithm 1 (SHA1) |
| RFC 3296 | Named Subordinate References in Lightweight Directory Access Protocol (LDAP) Directories |
| RFC 3377 | Lightweight Directory Access Protocol (v3) |
| RFC 3377 | Lightweight Directory Access Protocol (v3): Technical Specification |
| RFC 3383 | Internet Assigned Numbers Authority (IANA) Considerations for the Lightweight Directory Access Protocol (LDAP) |
| RFC 3454 | Preparation of Internationalized Strings ("stringprep") |
| RFC 3546 | Transport Layer Security (TLS) Extensions |
| RFC 3671 | Collective Attributes in the Lightweight Directory Access Protocol (LDAP) |
| RFC 3672 | Subentries in the Lightweight Directory Access Protocol (LDAP) |

***Table C–1 (Cont.) Supported RFCs***

| Number | Description |
|---|---|
| RFC 3673 | Lightweight Directory Access Protocol version 3 (LDAPv3): All Operational Attributes |
| RFC 3674 | Feature Discovery in Lightweight Directory Access Protocol (LDAP) |
| RFC 3698 | Lightweight Directory Access Protocol (LDAP): Additional Matching Rules |
| RFC 3771 | Lightweight Directory Access Protocol (LDAP) Intermediate Response Message |
| RFC 3829 | Lightweight Directory Access Protocol (LDAP) Authorization Identity Request and Response Controls |
| RFC 3866 | Language Tags and Ranges in the Lightweight Directory Access Protocol (LDAP) |
| RFC 3876 | Returning Matched Values with the Lightweight Directory Access Protocol version 3 (LDAPv3) |
| RFC 3909 | Lightweight Directory Access Protocol (LDAP) Cancel Operation |
| RFC 4346 | The Transport Layer Security (TLS) Protocol Version 1.1 |
| RFC 4370 | Lightweight Directory Access Protocol (LDAP) Proxied Authorization Control |
| RFC 4403 | Lightweight Directory Access Protocol (LDAP) Schema for Universal Description, Discovery, and Integration version 3 (UDDIv3) |
| RFC 4422 | Simple Authentication and Security Layer (SASL) |
| RFC 4505 | Anonymous Simple Authentication and Security Layer (SASL) Mechanism |
| RFC 4510 | Lightweight Directory Access Protocol (LDAP): Technical Specification Road Map |
| RFC 4511 | Lightweight Directory Access Protocol (LDAP): The Protocol |
| RFC 4512 | Lightweight Directory Access Protocol (LDAP): Directory Information Models |
| RFC 4513 | Lightweight Directory Access Protocol (LDAP): Authentication Methods and Security Mechanisms |
| RFC 4514 | Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names |
| RFC 4515 | Lightweight Directory Access Protocol (LDAP): String Representation of Search Filters |
| RFC 4516 | Lightweight Directory Access Protocol (LDAP): Uniform Resource Locator |
| RFC 4517 | Lightweight Directory Access Protocol (LDAP): Syntaxes and Matching Rules |
| RFC 4518 | Lightweight Directory Access Protocol (LDAP): Internationalized String Preparation |
| RFC 4519 | Lightweight Directory Access Protocol (LDAP): Schema for User Applications |

*Table C–1 (Cont.) Supported RFCs*

| Number | Description |
|---|---|
| RFC 4520 | Internet Assigned Numbers Authority (IANA) Considerations for the Lightweight Directory Access Protocol (LDAP) |
| RFC 4522 | Lightweight Directory Access Protocol (LDAP): The Binary Encoding Option |
| RFC 4524 | COSINE LDAP/X.500 Schema |
| RFC 4525 | Lightweight Directory Access Protocol (LDAP) Modify-Increment Extension |
| RFC 4526 | Lightweight Directory Access Protocol (LDAP) Absolute True and False Filters |
| RFC 4527 | Lightweight Directory Access Protocol (LDAP) Read Entry Controls |
| RFC 4528 | Lightweight Directory Access Protocol (LDAP) Assertion Control |
| RFC 4529 | Requesting Attributes by Object Class in the Lightweight Directory Access Protocol (LDAP) |
| RFC 4530 | Lightweight Directory Access Protocol (LDAP) entryUUID Operational Attribute |
| RFC 4532 | Lightweight Directory Access Protocol (LDAP) "Who am I?" Operation |
| RFC 4616 | The PLAIN Simple Authentication and Security Layer (SASL) Mechanism |
| RFC 4634 | US Secure Hash Algorithms (SHA and HMAC-SHA) |
| RFC 4752 | The Kerberos V5 ("GSSAPI") SASL Mechanism |
| RFC 5020 | The Lightweight Directory Access Protocol (LDAP) entryDN Operational Attribute |

## C.2 Internet Drafts Supported by Oracle Unified Directory

Table C–2 contains a list of Internet drafts supported by Oracle Unified Directory.

*Table C–2 Internet Drafts Supported by Oracle Unified Directory*

| Document | Description |
|---|---|
| draft-armijo-ldap-treedelete | Tree Delete Control |
| draft-behera-ldap-password-policy | Password Policy for LDAP Directories |
| draft-furuseth-ldap-untypedobject | Structural object class 'untypedObject' for LDAP/X.500 |
| draft-good-ldap-changelog | Definition of an Object Class to Hold LDAP Change Records |
| draft-haripriya-dynamicgroup | LDAP: Dynamic Groups for LDAPv3 |
| draft-howard-namedobject | A Structural Object Class for Arbitrary Auxiliary Object Classes |
| draft-howard-rfc2307bis | An Approach for Using LDAP as a Network Information Service |
| draft-ietf-boreham-numsubordinates | numSubordinates LDAP Operational Attribute |

*Table C–2 (Cont.) Internet Drafts Supported by Oracle Unified Directory*

| Document | Description |
|---|---|
| draft-ietf-ldapext-ldapv3-dupent | LDAP Control for a Duplicate Entry Representation of Search Results |
| draft-ietf-ldapext-ldapv3-vlv | LDAP Extensions for Scrolling View Browsing of Search Results |
| draft-ietf-ldapext-psearch | Persistent Search: A Simple LDAP Change Notification Mechanism |
| draft-ietf-ldup-subentry | LDAP Subentry Schema |
| draft-ietf-sasl-crammd5 | The CRAM-MD5 SASL Mechanism |
| draft-ietf-sasl-rfc2831bis | Using Digest Authentication as a SASL Mechanism |
| draft-poitou-ldap-schema-update | LDAP Schema Update Procedures |
| draft-sermersheim-ldap-subordinate-scope | Subordinate Subtree Search Scope for LDAP |
| draft-vchu-ldap-pwd-policy | Password Policy for LDAP Directories |
| draft-wahl-ldap-adminaddr | LDAP Administrator Address Attribute |
| draft-weltman-ldapv3-proxy | LDAP Proxied Authorization Control |
| draft-zeilenga-ldap-noop | The LDAP No-Op Control |
| draft-zeilenga-ldap-entrydn | The LDAP entryDN Operational Attribute |

## C.3 Other Specifications Supported by Oracle Unified Directory

Table C–3 contains a list of documents and standards supported by Oracle Unified Directory.

*Table C–3 Other Specifications Supported by Oracle Unified Directory*

| Number | Description |
|---|---|
| DSMLv2.doc | OASIS Directory Services Markup Language v2.0 Documentation |
| DSMLv2.xsd | OASIS Directory Services Markup Language v2.0 Standard |
| FIPS 180-1 | Secure Hash Standard (SHA-1) |
| FIPS 180-2 | Secure Hash Standard (SHS) (FIPS PUB 180-2) |

# D

# Glossary of terms for Oracle Unified Directory

This glossary defines the vocabulary that is used to describe LDAP and directory services and includes terms that are specific to Oracle Unified Directory.

## D.1 A

### D.1.1 abandon operation

The LDAP abandon operation can be used to request that the server stop processing on an outstanding request. The abandon request protocol op is as follows:

```
AbandonRequest ::= [APPLICATION 16] MessageID
```

The message ID provided in the request is the message ID of the operation to abandon.

The abandon operation does not have a response, so there is no way for clients to know whether the abandon operation was successful. Similarly, if an operation was abandoned, then no response will be provided for it, so the client may wait indefinitely for a response that will never be sent. Both of these issues are addressed by the cancel extended operation.

Bind, unbind, abandon, and StartTLS extended operations cannot be abandoned.

### D.1.2 abstract object class

An abstract object class is one that cannot be used directly in an entry but must be subclassed by either a structural object class or auxiliary object class. The subclasses will inherit any required and/or optional attribute type defined by the abstract class.

One of the most notable abstract object classes defined in LDAP is the `top` object class, which is the root class for virtually all other object classes defined in the server schema.

### D.1.3 Abstract Syntax Notation One

Abstract Syntax Notation One (ASN.1) is a mechanism for encoding data in a binary form. It uses a TLV structure, in which each element has a type, length, and value. The type component is a data type that indicates what kind of information is stored in the element and indicates how the value should be encoded. The length component specifies the number of bytes in the value, and the value is the actual data held by the element.

Examples of ASN.1 elements include the following:

**Null**
Null elements do not hold any value. They are generally used as placeholders when an element is required but no value is needed.

**Octet string**
Octet string elements hold a set of zero or more octets (bytes) of data. It can be used for holding string or binary data.

**Boolean**
Boolean elements hold values that represent either `true` or `false`.

**Integer**
Integer elements hold values that represent integer values.

**Enumerated**
Enumerated elements hold values that represent integer values where each value has a specific meaning.

**Sequence**
Sequence elements are containers that hold zero or more other ASN.1 elements in a manner where the order of the elements is significant.

**Set**
Set elements are containers that hold zero or more other ASN.1 elements in a manner where the order of the elements is not significant.

Note that ASN.1 is a general framework for binary encoding, but doesn't actually define how the data should be encoded. That is handled by an encoding rule, and there are a number of different kinds of ASN.1 encoding rules. LDAP uses the Basic Encoding Rules encoding, but other types include Distinguished Encoding Rules (DER), Canonical Encoding Rules (CER), and Packed Encoding Rules (PER).

## D.1.4 access control

Access control provides a mechanism for restricting who can get access to various kinds of information in the Directory Server. The access control provider can be used to control a number of things, including:

- Whether or not a client can retrieve an entry from the server.
- Which attributes within the entry the client is allowed to retrieve.
- Which values of an attribute the client is allowed to retrieve.
- The ways in which the client is able to manipulate data in the directory.

A number of things can be taken into account when making access control decisions, including:

- The DN as whom the user is authenticated.
- The method by which the client authenticated to the server.
- Any groups in which that user is a member.
- The contents of the authenticated user's entry.
- The contents of the target entry.
- The address of the client system.

- Whether or not the communication between the client and server is secure.
- The time of day and/or day of week of the attempt.

See Chapter 22, "Controlling Access To Data" for details on the access control syntax.

In addition to the access control subsystem, the directory server also provides a privilege that can be used to control what a user will be allowed to do. One of the privileges available is the `bypass-acl` privilege, which can be used to allow that client to bypass any restrictions that the access control subsystem would otherwise enforce.

## D.1.5 access control instruction (ACI)

See access control rule.

## D.1.6 access control rule

An access control rule (also called an access control instruction, or ACI), is a rule which may be used to grant or deny a user or set of users access to perform some kind of operation in the server. The Directory Server access control policy comprises the complete set of access control rules defined in the server.

See Chapter 22, "Controlling Access To Data" for more information about the syntax used for access control rules and the operations that can be allowed or denied using them.

## D.1.7 access log

The Directory Server access log provides a mechanism for keeping track of every operation processed by the server, including every request received and response returned. It may also be used to obtain information about the internal operations performed within the server.

The directory server provides an extensible framework for implementing access loggers (as well as error log and debug log loggers). The default access control log implementation writes information to a log file with two records per operation. The first record reflects the request received from the client and the second provides information about the result of the operation processing.

All messages will include a common set of elements including:

- The time that the message was logged.
- The type of operation being processed.
- The connection ID of the client connection that requested the operation.
- The operation ID of the operation on that client connection.
- The message ID of the message used to request the operation.

For abandon operation, request log messages include the message ID of the operation to abandon. There is no response to an abandon operation, but the server will nevertheless log a result message indicating whether the abandon was successful and the processing time in milliseconds.

For add operation, request log messages include the distinguished name of the entry to add. The response log message may include the result code, diagnostic message, matched DN, the authorization ID for the operation, and the processing time in milliseconds.

For bind operation, request log messages include the authentication type (either `SIMPLE` or `SASL` followed by the mechanism name) and the bind DN. The response log message may include the result code, diagnostic message, matched DN, authentication ID, authorization ID, and processing time in milliseconds.

For compare operation, request log messages include the target entry DN and the attribute type. The response log message may include the result code, diagnostic message, matched DN, authorization ID, and the processing time in milliseconds.

For delete operation, request log messages include the target entry DN. The response log message may include the result code, diagnostic message, matched DN, authorization ID, and the processing time in milliseconds.

For extended operation, request log messages include the object identifier for the extended request. The response log message may include the OID of the extended response, the result code, diagnostic message, matched DN, and the processing time in milliseconds.

For modify operation, request log messages include the target entry DN. The response log message may include the result code, diagnostic message, matched DN, authorization ID, and the processing time in milliseconds.

For modify DN operation, request log messages include the target entry DN, the new RDN, a flag indicating whether to delete the old RDN values, and the new superior DN. The response log message may include the result code, diagnostic message, matched DN, authorization ID, and the processing time in milliseconds.

For search operation, request log messages include the search base DN, search scope, LDAP search filter, and search attributes. The response log message may include the result code, number of entries returned, diagnostic message, matched DN, authorization ID, and the processing time in milliseconds.

For unbind operation, the request message will simply indicate that an unbind request has been received. There is no response to an unbind request, and no result log message.

### D.1.8 account expiration

Account expiration is a component of the Directory Server password policy that may be used to indicate that an account is no longer able to be used beyond a given date. This feature may be useful for creating temporary user accounts (for example, for use by contractors, interns, or other temporary workers) that will expire after a specified date.

Account expiration may be enabled by adding the `ds-pwp-account-expiration-time` operational attribute to the target user's entry. The value for this attribute should be a time stamp in generalized time format that specifies the time that the account should expire. Once the account expiration time has passed, the user will no longer be allowed to authenticate to the server.

### D.1.9 account lockout

Account lockout is a component of the Directory Server password policy that may be used to lock user accounts after too many failed bind attempts. Once an account has been locked, that user will not be allowed to authenticate. The lockout may be temporary (automatically ending after a specified period of time) or permanent (remaining in effect until an administrator resets the user's password).

## D.1.10 account status notification

An account status notification is a mechanism that can be used to provide indication that a user account has changed in a manner that is significant with regard to the server's password policy.

The types of account status notifications available for use in the server include:

- When the user's account has been account lockout
- When the user's account has been account lockout
- When the user's account has been unlocked by an administrator
- When the user's account has been manually disabled or re-enabled by an administrator
- When the user's account expiration
- When the user's password expiration or is about to expire
- When the user's password has been password reset
- When the user's password has been changed by the end user

The directory server provides an extensible framework for handling account status notifications. The default handler writes messages to the server's error log, but the framework can be used to send email messages or take other actions that may be desired.

## D.1.11 account usability control

The account usability control provides a pair of request and response controls that can be used to determine whether a user account may be used for authenticating to the server.

The request control has an OID of 1.3.6.1.4.1.42.2.27.9.5.8 and does not include a value. It should only be included in search operation messages.

The corresponding response control has an OID of 1.3.6.1.4.1.42.2.27.9.5.8 (the same as the request control), and it will be included in any search result entry messages for a search request that includes the account usability request control.

The value for the account usability response control is encoded as follows:

```
ACCOUNT_USABLE_RESPONSE ::= CHOICE {
     is_available           [0] INTEGER, -- Seconds before expiration --
     is_not_available       [1] MORE_INFO }

     MORE_INFO ::= SEQUENCE {
     inactive               [0] BOOLEAN DEFAULT FALSE,
     reset                  [1] BOOLEAN DEFAULT FALSE,
     expired                [2] BOOLEAN DEFAULT_FALSE,
     remaining_grace        [3] INTEGER OPTIONAL,
     seconds_before_unlock  [4] INTEGER OPTIONAL }
```

If the user account is available, then the control will include the number of seconds until the user's password expires, or -1 if password expiration is not enabled. If the user's account is not available, then the control will provide the reason it is unavailable.

For an example of using this control in a search request, see Searching Using the Account Usability Request Control.

## D.1.12 ACID

ACID is an acronym that stands for Atomicity, Consistency, Isolation, and Durability. This term is standard database terminology that refers to the characteristics that can be achieved using the transaction nature of the database. These elements include:

**Atomicity**
Each transaction performed in the database is atomic. That is, it either completely succeeds or completely fails. It never partially succeeds such that some changes that are part of the transaction are applied while others are not.

**Consistency**
The database is always in a consistent state such that the integrity of its contents will be preserved. It should not be possible for a successful or failed transaction to leave the database in an inconsistent state.

**Isolation**
The operations performed as part of a transaction will be isolated from other operations performed in the database at the same time. If one transaction is used to make a number of changes to database contents, then it should not be possible for another transactional operation to see the effects of those changes until they have been committed.

**Durability**
Any transaction that the database has reported as complete and committed successfully is guaranteed to be on persistent storage. Even if the directory server, or the underlying JVM, operating system, or hardware should fail the instant after the notification of the successful commit, then that change will not be lost.

The Berkeley DB Java Edition used as the data store for the primary back end provides full support for ACID compliance, although it also provides methods for relaxing its compliance to these constraints if desirable for performance reasons. The directory server exposes some of this flexibility, particularly with regard to configuring how durable the changes will be (for example, it is possible to configure the server so that changes are not immediately flushed to disk, which may allow better write performance but could cause the loss of one or more changes in the event of a hardware or software failure).

## D.1.13 add operation

The LDAP add operation can be used to create an entry in the Directory Server. The add request protocol op is defined as follows:

```
AddRequest ::= [APPLICATION 8] SEQUENCE {
     entry           LDAPDN,
     attributes      AttributeList }
```

The elements included in this request include the distinguished name of the entry to add and the set of attributes to include in that entry.

The response to an LDAP add operation is an LDAP result element, defined as follows:

```
AddResponse::= [APPLICATION 9] LDAPResult
```

## D.1.14 alias

An alias is a special type of entry that references another entry in the server, much like a symbolic link in a UNIX file system. It should include the `alias` object class and the

`aliasedObjectName` attribute with a value equal to the DN of the entry that it references.

Aliases are primarily used for search operation. In particular, the search request includes an element that specifies the dereference policy that should be used when aliases are encountered. The allowed dereference policy values include:

**neverDerefAliases**
The server should never dereference alias entries.

**derefInSearching**
The server should dereference any alias entries that it finds in the possible set of search result entries, but if the search base DN specifies an alias entry it will not be de referenced.

**derefFindingBaseObj**
The server should dereference the search base entry if it is an alias, but it will not dereference any aliases within the possible set of search result entries.

**derefAlways**
The server should dereference any aliases encountered, whether in the search base entry or in the possible set of search result entries.

Note that aliases are an optional part of the LDAPv3 protocol, and the directory server does not currently support them.

## D.1.15 AND search filter

An AND search filter is a type of LDAP search filter that is intended to serve as a container that holds zero or more other search filters. In order for an entry to match an AND filter, it must match all of the filters contained in that AND filter.

AND filters may be represented as a string by enclosing the entire filter in parentheses and placing an ampersand just after the opening parenthesis. For example, a filter of `(&(objectClass=person)(uid=john.doe))` represents an AND search filter that embeds the `(objectClass=person)` and `(uid=john.doe)` equality filters.

An AND filter that does not contain any embedded filters is called an LDAP true filter. The string representation for an LDAP true filter is an ampersand `(&)`, and LDAP true filters will always match any target entry.

## D.1.16 anonymous bind

An anonymous bind is a type of bind operation using simple authentication with a zero-length bind DN and a zero-length password. It may be used to destroy any previous authentication performed on a connection and return it to an unauthenticated state.

Note that there is an ANONYMOUS SASL mechanism that has the same effect, but in general the term "anonymous bind" refers to the simple bind operation with no DN and password.

## D.1.17 ANONYMOUS SASL mechanism

The ANONYMOUS SASL mechanism is a type of Simple Authentication and Security Layer authentication mechanism. It is different from other SASL mechanisms in that it is used to create an unauthenticated session, and will destroy any previous authentication that may have been performed on the connection.

The ANONYMOUS SASL mechanism provides the ability to include trace information in the request that may be included in the server's access log. This trace information can provide information about the client performing the bind, although because no authentication is performed the validity of the trace information cannot be guaranteed.

### D.1.18 approximate index

An approximate index is a type of index that is used to efficiently identify which entries are approximately equal to a given assertion value. An approximate index can be maintained only for attributes that have a corresponding approximate matching rule. That matching rule are used to normalized value to use as index keys, and the value for that key is the ID list containing the entry ID of the entries with values that are approximately equal to that normalized value.

### D.1.19 approximate search filter

An approximate search filter is a type of LDAP search filter that can be used to identify entries that contain a value for a given attribute that is approximately equal to a given assertion value. The server will use an approximate matching rule to make the determination.

The string representation of an LDAP approximate filter comprises an opening parenthesis followed by the attribute name, a tilde, an equal sign, the attribute value, and the closing parenthesis. For example, an equality filter of `(givenName~=John` will match any entry in which the `givenName` attribute contains a value that is approximately equal to `John`.

### D.1.20 ASN.1

See Abstract Syntax Notation One.

### D.1.21 assertion value

An assertion value is the value of an attribute value assertion. The assertion value is provided to a matching rule in order to make a determination about the attribute value of a specified attribute.

### D.1.22 attribute

An attribute is a named set of values. An attribute has an attribute description, which contains the name of that attribute (which links it to an attribute type) and an optional set of attribute option, and a collection of one or more values.

An entry contains a collection of attributes. It is possible for an entry to have multiple attributes with the same attribute type but different sets of options.

### D.1.23 attribute description

An attribute description is used to identify a given attribute in an entry. An attribute description contains a name or OID that ties it to an attribute type and zero or more attribute option. If the attribute description contains any attribute options, then they are separated from the attribute name/OID by a semicolon, and a semicolon is also used to separate individual attribute options if there is more than one option in the attribute description.

## D.1.24 attribute option

An attribute option is a kind of tag that provides additional information about the way that an attribute should be interpreted. An attribute description consists of the attribute name or object identifier followed by zero or more attribute options. If there are attribute options, then they are separated from the attribute name and from each other using semicolons. For example, in the attribute description `userCertificate;binary`, the attribute name is `userCertificate` and the attribute option is `binary`.

Attribute options can be used for several purposes, including providing information about how the server should treat that attribute (for example, the binary encoding option as described in RFC 4522 (`http://www.ietf.org/rfc/rfc4522.txt`)) They may also be provided for the benefit of clients in some form (for example, the language tag options as described in RFC 3866 (`http://www.ietf.org/rfc/rfc3866.txt`), which make it possible to provide an attribute value in different languages).

## D.1.25 attribute syntax

An attribute syntax is a schema element that defines a kind of data type that is used to dictate the kind of information that may be stored in an attribute value. Any attempt to store an attribute value that violates the syntax for the associated attribute type should be rejected.

Common attribute syntaxes include:

**Binary**
Can hold any kind of data, whether textual or not, that should be compared on a byte-for-byte basis. Note that the binary syntax has been deprecated in favor of the octet string syntax.

**Boolean**
Can hold values of either `TRUE` or `FALSE`.

**Directory String**
Can hold any kind of string value (technically, binary values are allowed as well, but directory string values are typically strings).

**Distinguished Name**
Can hold values that are valid distinguished name.

**Generalized Time**
Can hold values that contain time stamps of varying precision (anywhere from an hour to a fraction of a second) including time zone information. For example, the value `20070525222745Z` represents a time stamp of May 25, 2007 at 10:27:45 PM in the UTC time zone.

**IA5 String**
Can hold values that contain ASCII strings (that is, use of non-ASCII characters is not allowed).

**Integer**
Can hold integer values. Positive, negative, and zero values are allowed.

**Octet String**
Can hold any kind of data that should be compared on a byte-for-byte basis.

**Postal Address**
Can hold a multi-line address, in which the lines of the address should be separated by dollar signs.

**Printable String**
Can hold a string containing any combination of printable characters. Printable characters include all uppercase and lowercase ASCII letters, the numeric digits, the space character, and the symbols `' ( ) + , - . = / : ?`.

**Telephone Number**
Can hold telephone number values.

The set of attribute syntaxes defined in the server may be determined by retrieving the `ldapSyntaxes` attribute of the subschema subentry. For more information about attribute syntaxes, see Understanding Attribute Syntaxes.

## D.1.26 attribute type

An attribute type is a schema element that correlates an object identifier and a set of names with an attribute syntax and a set of matching rule.

The components of an attribute type definition include:

- An OID used to uniquely identify the attribute type.
- A set of zero or more names that can be used to more easily reference the attribute type.
- An optional equality matching rule that specifies how equality matching should be performed on values of that attribute. If no equality matching rule is specified, then the default equality rule for the associated attribute syntax will be used. If the associated syntax doesn't have a default equality matching rule, then equality operations will not be allowed for that attribute.
- An optional ordering matching rule that specifies how ordering operations should be performed on values of that attribute. If no ordering matching rule is specified, then the default ordering rule for the associated attribute syntax will be used. If the associated syntax doesn't have a default ordering matching rule, then ordering operations will not be allowed for that attribute.
- An optional substring matching rule that specifies how substring matching should be performed on values of that attribute. If no substring matching rule is specified, then the default substring rule for the associated attribute syntax will be used. If the associated syntax doesn't have a default substring matching rule, then substring operations will not be allowed for that attribute.
- An optional syntax OID that specifies the syntax for values of the attribute. If no syntax is specified, then it will default to the directory string syntax.
- A flag that indicates whether the attribute is allowed to have multiple values.
- An optional attribute usage string indicating the context in which the attribute is to be used.
- An optional flag that indicates whether the attribute can be modified by external clients.

The set of attribute types defined in the server may be determined by retrieving the `attributeTypes` attribute of the subschema subentry. For more information about attribute types, see Understanding Attribute Types.

## D.1.27 attribute usage

An attribute type attribute usage defines the contexts in which it may be used. There are four types of attribute usage:

**userApplications**
This should be used for all attribute types that are intended for use in holding user-defined data.

**directoryOperation**
This should be used for attribute types that are used for behind-the-scenes processing within the server.

**distributedOperation**
This should be used for attribute types that store operational data that need to be distributed (that is, replication) throughout the directory environment.

**dSAOperation**
This should be used for attribute types that store operational data that should be stored only in one server and should not be replicated throughout the directory environment.

Attributes with a usage of `userApplications` are known as user attribute. Attributes with a usage of `directoryOperation`, `distributedOperation`, or `dSAOperation` are known as operational attribute.

## D.1.28 attribute value

An attribute value describes an element of actual data held by an attribute. An attribute may have multiple values, if allowed by the associated attribute type. The way that the server should interact with the values of that attribute is governed by that attribute's attribute syntax and matching rule.

## D.1.29 attribute value assertion

An attribute value assertion (AVA) is a combination of an attribute description and an attribute value. The assertion value is used in conjunction with a matching rule in order to make the determination. If the matching rule is an equality matching rule, then it will be used to determine whether the attribute contains a given value. If it is an ordering matching rule, then the AVA will be used to determine whether the attribute contains a value that is greater than or equal to, or less than or equal to, the assertion value. If it is an approximate matching rule, then the AVA will be used to determine whether the attribute contains a value that is approximately equal to the assertion value. Substring matching is more complex and uses a substring assertion rather than a simple assertion value.

Attribute Value assertions are used in LDAP compare operation, as well as equality search filter, greater than or equal to search filter, less than or equal to search filter, and approximate search filter search filters.

## D.1.30 audit log

The audit log is a special type of access log that is used to log information about all changes that are made in the server. It provides a log of those changes in LDAP Data Interchange Format form so that administrators can see exactly what changes were made. This information can be used for diagnostic purposes when investigating a problem, to help better understand the kinds of changes that an application might

make in the directory, or to help collect information about changes for replay to an alternate repository.

The name "audit log" is a legacy term referring to its use in the Netscape Directory Server. It should not be confused with a log that could be used for security auditing, as it only records changes to directory data and does not keep track of things like successful or failed authentication attempts. However, in many cases, the combination of the content from the traditional access log and the audit log can be used to obtain this kind of information. If desired, an administrator could also provide a custom access logging implementation to keep track of any kind of desired information.

## D.1.31 authentication

Authentication is the process whereby a client identifies itself to the directory server and provides proof of its identity. In LDAP, this is performed through the use of a bind operation.

The authentication process has two phases:

#### Identification

The client identifies itself to the server in some way. In simple authentication, the DN provided in the bind request is used for this purpose. In Simple Authentication and Security Layer authentication, the identity of the client is obtained through some other means (for example, using a certificate, a Kerberos principal, or some other kind of identifier).

#### Verification of Identity

The client must provide sufficient proof that it is who it has identified itself to be. In simple authentication, this is done through the password. In SASL authentication, this verification is obtained in a manner specific to the associated mechanism (it may be a password, or it may be a certificate or some other form of proof).

Some authentication mechanisms may be considered stronger than others. For example, simple authentication may be considered less trustworthy if the client has a password that is easy to guess or obtain through some other means, whereas authentication using a certificate or Kerberos credentials might be considered much stronger and harder to forge. The directory server's access control implementation may be configured to take the client's authentication mechanism into account when determining whether a requested operation will be allowed.

## D.1.32 authentication ID

An authentication ID is an identifier that is used by a client to identify itself to the Directory Server for certain kinds of Simple Authentication and Security Layer mechanisms (for example, CRAM-MD5 SASL mechanism, DIGEST-MD5 SASL mechanism, and PLAIN SASL mechanism). It can be used to allow a client to identify itself with a username (or other friendly identifier) rather than a distinguished name.

In most cases, an authentication ID should be specified in one of the following forms:

- The string `dn:` followed by the distinguished name of the target user (or just the string `dn:` if the authentication identity should be that of the anonymous user).
- The string `u:` followed by a username used to identify the user. An identity mapper will be used to map the provided username to the corresponding user entry.

## D.1.33 authentication password syntax

The authentication password syntax defines a standard method for encoding a user password for storage in the server, ideally in a manner that makes it difficult or impossible to determine the clear-text value of that password.

The authentication password syntax is described in RFC 3112 (`http://www.ietf.org/rfc/rfc3112.txt`), which defines the `authPassword` attribute type and a corresponding `authPasswordObject` auxiliary object class that will allow the use of that attribute.

The basic form of a password encoded using the authentication password syntax is:

*scheme* `$`*authInfo* `$` *authValue*

where *scheme* is the name of the scheme used to encode the value, *authInfo* is some kind of modifier (for example, a salt) used in the encoding process, and *authValue* is the encoded password information. For example, the value `SHA1$RzqH67DY3uQ=$atAcDs1eS+IJwPy7V4UDXEoBrDI=` is encoded using the authentication password syntax The scheme is `SHA1`, the authInfo element is `RzqH67DY3uQ=`, and the authValue element is `atAcDs1eS+IJwPy7V4UDXEoBrDI=`.

The authentication password schemes supported by the directory server include the following:

**MD5**
Uses the MD5 message digest.

**SHA1**
Uses the SHA-1 variant of the Secure Hash Algorithm.

**SHA256**
Uses the 256-bit SHA-2 variant of the Secure Hash Algorithm.

**SHA384**
Uses the 384-bit SHA-2 variant of the Secure Hash Algorithm.

**SHA512**
Uses the 512-bit SHA-2 variant of the Secure Hash Algorithm.

## D.1.34 authorization

Authorization is the process of determining whether a user will be allowed to perform a requested operation. A number of server components may be involved in the authorization process, including:

- The access control handler.
- The privilege subsystem.
- The password policy.
- Custom plug-in installed in the server.

## D.1.35 authorization ID

An authorization ID is an identifier that is used by a client to indicate that one or more operations should be performed under the authority of an alternate identity. This alternate authorization identity can last for a single operation (when used in conjunction with the proxied authorization control) or for the entire duration of an authentication session (when used in conjunction with an appropriate SASL

mechanism, like DIGEST-MD5 SASL mechanism, GSSAPI SASL mechanism, or PLAIN SASL mechanism).

In most cases, an authorization ID should be specified in one of the following forms:

- The string `dn:` followed by the distinguished name of the target user (or just the string `dn:` if the authorization identity should be that of the anonymous user).
- The string `u:` followed by a username used to identify the user. An identity mapper maps the provided username to the corresponding user entry.

The ability for a client to use an alternate authorization identity is controlled by the `proxied-auth` privilege. In some cases, additional access control rights may also be required.

### D.1.36 authorization identity control

The authorization identity controls are a pair of request and response controls defined in RFC 3829 (`http://www.ietf.org/rfc/rfc3829.txt`) that can be used in conjunction with an bind operation to allow the client to learn the authorization identity for the client connection.

The authorization identity request control has an object identifier of `2.16.840.1.113730.3.4.16` and does not have a value. The authorization identity response control has an OID of `2.16.840.1.113730.3.4.15` and the value of that control should be a string representing the authorization identify for that connection (or an empty string if the authorization identity is that of the anonymous user). The response control should only be included in the response if the authentication was successful.

Note that the authorization identity controls are only allowed for use in conjunction with the LDAP bind operation, and therefore cannot be used after the client has authenticated. The "Who Am I?" extended operation can be used to obtain the authorization identity at any time after the bind has completed.

For an example of using this control in a search request, see Searching Using the Authorization Identity Request Control.

### D.1.37 auxiliary object class

An auxiliary object class is one that does not define the core type of an entry, but defines additional characteristics of that entry. An entry can contain zero or more auxiliary object classes. The set of auxiliary classes allowed for use in an entry may be controlled by a DIT content rule associated with that entry's structural object class.

### D.1.38 AVA

See attribute value assertion.

## D.2 B

### D.2.1 back end

A Directory Server back end provides a repository for storing data and a set of logic for interacting with that data. A back end will typically contain some kind of database and may maintain a set of index that allows the back end to quickly locate entries for various operations. All back ends will have the following qualities:

- A back end ID, which uniquely identifies that back end among all other back ends in the server.
- A set of one or more base distinguished name that indicate the data that the back end holds.
- A writability mode, which indicates whether the back end will accept write operations.

The logic provided by the back end includes:

- A method for determining whether a given entry exists, based on its DN
- A method for retrieving an entry, based on its DN
- A method of adding a new entry to the database (as part of processing an LDAP add operation)
- A method for removing an existing entry from the database (as part of processing an LDAP delete operation)
- A method for replacing an entry in the database (as part of processing an LDAP modify operation)
- A method for renaming an entry in the database (as part of processing an LDAP modify DN operation)
- A method for processing an LDAP search operation
- A method for exporting the contents of the database in LDAP Data Interchange Format form
- A method for importing data in LDAP Data Interchange Format form into the database
- A method for performing a backup of the data
- A method for performing a restore of a previous backup

## D.2.2 backup

A backup is a transportable representation of the data in a Directory Server back end. Each back end is responsible for controlling whether or not it is possible to back up its contents, and ensuring that the backup information is suitable to be restore at a later time. Note that the term "back up" is a verb (the action of backing up the contents of the back end) and "backup" is a noun (what you get when you perform a backup).

There are a number of reasons that a back end may not provide a backup mechanism. Some reasons include:

- The back end only contains temporary, point-in-time information that doesn't make sense to archive or attempt to restore at a later time (for example, the root DSE or the monitor back ends).
- The back end stores its information in a remote repository that is not directly available to be archived. In cases like this, the external repository will likely have its own backup and restore mechanism.

The primary back end used by the directory server is one that uses the Berkeley DB Java Edition as its underlying database and that back end provides complete backup and restore capabilities. The backup mechanism is also very portable and can be transported across different platforms and different filesystem locations, and it is suitable for use as a binary copy mechanism.

## D.2.3 base64 encoding

Base64 encoding is a way of representing binary data in a text-only form. It is commonly used in LDAP Data Interchange Format for values containing non-ASCII characters, or for values that could otherwise be ambiguous (for example, values that begin or end with spaces). It is also frequently used to encode certificate contents or the output of message digests like MD5 or Secure Hash Algorithm. The base64 encoding is described in section 5.2 of RFC 1341 (`http://www.ietf.org/rfc/rfc1341.txt`).

The basic principle of base64 encoding is that it defines a 64-character alphabet containing the following characters in the given order:

```
ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/
```

Each of those characters is assigned a numeric value between 0 and 63 based on its position in the list (that is, `A` is 0, `B` is 1, `C` is 2,... `+` is 62, and `/` is 63). A value is broken up into six-bit segments, and each of those six bits is converted into a numeric value between 0 and 63 and replaced with the specified character from the alphabet given above. This means that every three bytes of a binary value is converted into four characters from the base64 alphabet. If the length of the binary value is not a multiple of three bytes, then it is zero-padded and either one or two equal signs are appended to the base64-encoded value.

## D.2.4 Basic Encoding Rules

The Basic Encoding Rules (BER) are a set of Abstract Syntax Notation One encoding rules that define a specific way in which information may be encoded in a binary form. It is used as the underlying mechanism for encoding message.

### D.2.4.1 Basic Encoding Rules Overview

Many network protocols are text-based, which has the advantages of being relatively easy to understand if you examine the network traffic, and in many cases you can even interact with the target server by telnetting to it and typing in the appropriate commands. However, there are disadvantages as well, including that they are generally more verbose and less efficient to parse than they need to be. On the other hand, other protocols use a binary encoding that is more compact and more efficient. LDAP falls into this category, and uses the ASN.1 (abstract syntax notation one) mechanism, and more specifically the BER (basic encoding rules) flavor of ASN.1. There are a number of other encoding rules (such as DER, PER, and CER) that fall under the ASN.1 umbrella, but LDAP uses BER.

This section discusses the subset of BER that is used by LDAP in particular and does not address other cases.

BER elements use a TLV structure, where TLV stands for "type", "length", and "value". That is, each BER element has one or more bytes (in LDAP, typically only a single byte) that indicates the data type for the element, one or more bytes that indicate the length of the value, and the encoded value itself (where the form of the encoded value depends on the data type), which can be zero or more bytes, as described in the following sections:

- The BER type
- The BER Length
- The BER Value

### D.2.4.2 The BER Type

The BER type indicates the data type for the value of the element. The BER specification provides several different data types, but the most commonly used by LDAP include `OCTET STRING` (which can be either a text string or just some binary data), `INTEGER`, `BOOLEAN`, `NULL`, `ENUMERATED` (like an integer, but where each value has a special meaning), `SEQUENCE` (an ordered collection of other elements, similar to an array), and `SET` (the same as a sequence, except that the order doesn't matter). There is also a `CHOICE` element, but it typically allows one of a few different kinds of elements.

The BER type is typically only a single byte, and this byte has data encoded in it. The two most significant bits (the two leftmost bits, because BER uses big endian/network ordering) are used to indicate the class for the element, using these possible class values:

**00**
The universal class. Most BER elements have a universal type, so any element with a universal type specifies what kind of data it holds. Examples of universal types include 0x01 (`BOOLEAN`), 0x02 (`INTEGER`), 0x04 (`OCTET STRING`), 0x05 (`NULL`), 0x0A (`ENUMERATED`), 0x30 (`SEQUENCE`), and 0x31 (`SET`). The binary encodings for all of those type values have the leftmost two bits set to zero.

**01**
The application-specific class. This class allows an application to define its own types that are consistent throughout that application. In this context, LDAP is considered an application. For example, when 0x42 appears in LDAP, it indicates an unbind request protocol op, because RFC 2251 section 4.3 (`https://tools.ietf.org/html/rfc2251#section-4.3`) states that the unbind request protocol op has a type of `[APPLICATION 2]`.

**10**
The context-specific class. This class indicates that the type is specific to a particular usage within a given application. The same type can be re-used in different contexts in the same application as long as there is enough other information to determine which context is applicable in a given situation. For example, in the context of the credentials in a bind request protocol op, the context-specific type 0x80 is used to hold the bind password, but in the context of an extended operation it would be used to hold the request OID.

**11**
The private class, not typically used in LDAP.

The next bit (the third from the left) is the primitive/constructed bit. If it is set to zero (off), then the element is considered primitive, and the value is encoded in accordance with the rules of that data type. If it is set to one (on), then it means that the value is constructed from zero or more other ASN.1 elements that are concatenated together in their encoded forms. For example, for the universal `SEQUENCE` type of 0x30, the binary encoding is `00110000` and the primitive/constructed bit is set to one indicating that the value of the sequence is constructed from zero or more encoded elements.

The final five bits of the BER type byte specify the value of that type, and they are treated as a simple integer value (where `00000` is zero, `00001` is one, `00010` is two, `00011` is three, and so on). The only special value is `11111`, which means that the type value is larger than can fit in the five bits allowed, and so multiple bytes are required. This value is not used in LDAP.

### D.2.4.3 The BER Length

The second component in the TLV structure of a BER element is the length. This specifies the size in bytes of the encoded value. For the most part, this uses a straightforward binary encoding of the integer value (for example, if the encoded value is five bytes long, then it is encoded as 00000101 binary, or 0x05 hex), but if the value is longer than 127 bytes then it is necessary to use multiple bytes to encode the length. In that case, the first byte has the leftmost bit set to one and the remaining seven bits are used to specify the number of bytes required to encode the full length. For example, if there are 500 bytes in the length (hex 0x01F4), then the encoded length will actually consist of three bytes: `82 01 F4`.

Note that there is an alternate form for encoding the length called the indefinite form. In this mechanism, only a part of the length is given at a time, similar to the chunked encoding that is available in HTTP 1.1. However, this form is not used in LDAP, as specified in RFC 2251 section 5.1 (https://tools.ietf.org/html/rfc2251#section-5.1).

### D.2.4.4 The BER Value

The BER element contains the actual data of the element. Because BER is a binary encoding, the encodings can take advantage of that to represent the data in a compact form. As such, each data type has its own encoded form:

`NULL`
The `NULL` element never has a value, and therefore the length is always zero.

`OCTET STRING`
The value of this element is encoded as a concatenation of the raw bytes of the data being represented. For example, to represent the string `Hello`, the encoded value would be `48 65 6C 6C 6F`. The value can have a length of zero bytes.

`BOOLEAN`
The value of this element is always a single byte. If all the bits in that byte are set to zero (0x00), then the value is `FALSE`. If one or more of the bytes is set to one, then the value is `TRUE`. As a result, there are 255 different ways to encode a `BOOLEAN` value of `TRUE`, but in practice it is generally encoded as 0xFF (that is, all the bits are set to one).

`INTEGER`
The value of this element is encoded as a binary integer in two's complement form. Although BER itself does not place a limit on the magnitude of the values that can be encoded, many software implementations have a cap of four or eight bytes (that is, 32-bit or 64-bit integer values), and LDAP generally uses a maximum of 4 bytes (which allows encoding values within the plus or minus 2 billion range). There is always at least one byte in the value.

`ENUMERATED`
The value of this element is encoded in exactly the same way as the value of an `INTEGER` element.

`SEQUENCE`
The value of this element is simply a concatenation of the encoded BER elements contained in the sequence. For example, to encode a sequence with two octet string elements encoding the text `Hello` and `there`, the encoded sequence value is `04 05 48 65 6C 6C 6F 04 05 74 68 65 72 65`. A sequence value can be zero bytes if there are no elements in the sequence.

SET
The value of this element is encoded in exactly the same way as the value of a SEQUENCE element.

### D.2.4.5 BER Encoding Examples

The example above for encoding a SEQUENCE value had two complete BER elements concatenated together: the OCTET STRING representations of the strings Hello and there:

```
04 05 48 65 6C 6C 6F
04 05 74 68 65 72 65
```

In both of these cases, the first byte is the type (0x04, which is the universal primitive OCTET STRING type), and the second is the length (0x05, indicating that there are five bytes in the value). The remaining five bytes are the encoded representations of the strings Hello and there.

The following example encodes the integer value 3 using a context-specific type value of 5 instead of the universal INTEGER type:

```
85 01 03
```

The next example encodes an LDAP bind request protocol op as defined in RFC 2251 section 4.2 (https://tools.ietf.org/html/rfc2251#section-4.2). A simplified BNF representation of this element is as follows:

```
BindRequest ::= [APPLICATION 0] SEQUENCE {
     version                    INTEGER (1 .. 127),
     name                       OCTET STRING,
     authentication             CHOICE {
          simple                [0] OCTET STRING,
          sasl                  [3] SEQUENCE {
               mechanism        OCTET STRING,
               credentials      OCTET STRING OPTIONAL } } }
```

This example encodes a bind request using simple authentication for the user cn=test with a password of password. The complete encoding for this bind request protocol op is:

```
60 16 02 01 03 04 07 63 6E 3D 74 65 73 74 80 08 70 61 73 73 77 6F 72 64
```

In analysis, that string of bytes contains the following information:

- The first byte is 0x60 and it is the BER type for the bind request protocol op. It comes from the [APPLICATION 0] SEQUENCE portion of the definition. Because it is application-specific, then the class bytes are 01, and because it is a SEQUENCE, it is constructed. Put that together with a type value of zero, the binary representation is 01100000, which is 0x60 hex.
- The second byte is 0x16, which indicates the length of the bind request sequence. 0x16 hex is 22 decimal, and the number of bytes after the 0x16 is 22.
- The next three bytes are 02 01 03, which is a universal INTEGER value of 3. It corresponds to the version component of the bind request sequence, and it indicates that this is an LDAPv3 bind request.
- The next nine bytes are 04 07 63 6E 3D 74 65 73 74, which is a universal OCTET STRING containing the text cn=test. It corresponds to the "name" component of the bind request sequence.

- The last component is `80 08 70 61 73 73 77 6F 72 64`, which is an element with a type of `context-specific primitive 0` and a length of eight bytes. As specified in the definition of the bind request protocol op, `context-specific` maps to the simple authentication type and that it should be treated as an `OCTET STRING`, and those eight bytes in the value do represent the encoded string `password`.

### D.2.5 BER

See Basic Encoding Rules.

### D.2.6 Berkeley DB Java Edition

The Berkeley DB Java Edition (also referred to as "Berkeley DB JE", "BDBJE", or "JE") is a pure Java database designed by Sleepycat Software, which was purchased by the Oracle Corporation. It provides a highly-scalable, high-performance, transactional B-Tree database, with support for full ACID semantics and it is used as the primary database for storing user data.

The directory server provides a back end that uses the Berkeley DB Java Edition for storing its information. This back end is often called the "JE Backend" or simply "JEB". It uses a Berkeley DB Java Edition environment that consists of multiple individual databases. The id2entry database provides a mechanism for mapping entry ID values to entry contents. Other databases serve as index that can be used to quickly find entry contents for processing various types of operations.

### D.2.7 binary copy

Binary copy refers to the process of performing a backup of a Directory Server back end of one server instance and restore that back end into another instance of the server. This can provide a fast disaster recovery mechanism and can also be used as a replica initialization mechanism.

Not all Directory Server back ends necessarily support the use of binary copy, and those that do may not support it in all circumstances. The primary back end type used by the directory server is based on the use of the Berkeley DB Java Edition, and it does support the use of the binary copy mechanism, including across different operating systems and CPU architectures, and with different filesystem locations. However, it does require that both servers have the same set of base distinguished name and the same types of index defined.

### D.2.8 bind operation

The LDAP bind operation can be used to authenticate to the Directory Server. There are two basic types of bind operations:

- A simple bind operation, which uses simple authentication involving a bind DN and password to authenticate to the server.
- A SASL bind operation, which uses the Simple Authentication and Security Layer to authenticate the client, which can use a variety of types of credentials based on the selected SASL mechanism.

The bind request protocol op is defined as follows:

```
BindRequest ::= [APPLICATION 0] SEQUENCE {
     version                 INTEGER (1 ..  127),
     name                    LDAPDN,
     authentication          AuthenticationChoice }
```

```
AuthenticationChoice ::= CHOICE {
     simple                  [0] OCTET STRING,
                                 -- 1 and 2 reserved
     sasl                    [3] SaslCredentials,
     ...  }

SaslCredentials ::= SEQUENCE {
     mechanism               LDAPString,
     credentials             OCTET STRING OPTIONAL }
```

The elements of the request include:

- The LDAP protocol version. Allowed values are 2 and 3, although LDAPv2 has been classified as a historical protocol and is no longer recommended for use.
- The bind DN. This is always used for simple authentication (although it may be a zero-length string for anonymous simple authentication), and is generally not used for SASL authentication.
- The credentials. The type of credentials provided vary based on the authentication type.
  - For simple authentication, the credentials should be the password for the target bind DN, or an empty string for anonymous simple authentication.
  - For SASL authentication, the credentials should include the name of the SASL mechanism to use, and may optionally include encoded credential information appropriate for the SASL mechanism.

The response to an LDAP bind operation is defined as follows:

```
BindResponse ::= [APPLICATION 1] SEQUENCE {
     COMPONENTS OF LDAPResult,
     serverSaslCreds    [7] OCTET STRING OPTIONAL }
```

This indicates that the bind response will include the elements in the LDAP result object and may also include a set of server SASL credentials if appropriate for the authentication type.

# D.3 C

## D.3.1 cancel extended operation

The LDAP Cancel extended operation is an extended operation that provides a function similar to the core LDAP abandon operation in that it can be used to request that the server stop processing on an operation in progress. The primary advantages of the Cancel extended operation over the abandon operation are that both the cancel request and the operation being canceled are guaranteed to get a response, whereas there is no response for the abandon request and there may not be a response for the operation being abandoned.

The Cancel extended operation is defined in RFC 3909 (http://www.ietf.org/rfc/rfc3909.txt). The value of the Cancel Request extended operation is encoded as follows:

```
cancelRequestValue ::= SEQUENCE {
     cancelID        MessageID
                     -- MessageID is as defined in [RFC2251]
```

```
}
```

### D.3.2 CDDL

See Common Development and Distribution License.

### D.3.3 certificate

A certificate is an element of public key cryptography that may be used to perform asymmetric encryption. In particular, a certificate consists of a pair of keys (called the "public key" and the "private key", respectively) that are linked so that any data encrypted using the public key can be decrypted using the private key. With many public key algorithms, like RSA, the reverse is also true so that any data encrypted with the private key can be decrypted using the public key.

The term certificate has different meanings, based on the context in which it is used. In many cases, it refers to only the public key (in particular, whenever the server presents its certificate to the client, or if a client presents its certificate to the server, then only the public key is included). However, in other cases, it does include the private key (i.e., the server will require the use of the private key to establish a secure communication channel with the client, and the client will need access to its private key in order to send its own certificate to the server).

Certificates have two primary uses in the directory server. The first is for providing a secure communication mechanism, generally through the use of Secure Sockets Layer or StartTLS extended operation. In this case, the negotiation process involves the client encrypting information using the server's public key so that only the server can decrypt it using its public key and that information will not be exposed to any third party that might be able to observe the communication. Certificates may also be used for data signing, in which case the server will encrypt information using its private key, and clients will know that the data is legitimately from the server if it can be decrypted using the server's public key.

### D.3.4 certificate mapper

A certificate mapper provides the logic required to identify a user in the Directory Server that corresponds to a provided client certificate. The mapping may use any of the information contained in the certificate, although many certificate mappers are based primarily on the certificate's subject (the name of the certificate, which comprises a number of attribute-value pairs and looks very much like an LDAP distinguished name.

For more information about the certificate mappers available for use in the directory server, see the Certificate Mapper Configuration (`http://www.opends.org/promoted-builds/latest/OpenDS/build/docgen/configuration_guide/certificate-mapper.html`).

### D.3.5 chaining

Chaining provides a mechanism for making data in a remote Directory Server instance appear as if it is part of the local server. That is, chaining is used to present a part of the directory information tree using data from another server. Any request that the server receives for data in a chained portion of the DIT will be transparently forwarded to the server that actually contains the request.

### D.3.6 changelog

A changelog is a special kind of database that is used to keep track of the changes that occur in a Directory Server instance. There are two different kinds of changelogs:

- A replication changelog stores change information in a format needed for replication.
- An LDAP-accessible changelog that represents its data in the format specified in draft-good-ldap-changelog that allows clients to learn about the changes that have occurred in the directory environment.

### D.3.7 cn=Directory Manager

See directory manager.

### D.3.8 collective attribute

A collective attribute is a special type of virtual attribute that is defined in RFC 3671 (http://www.ietf.org/rfc/rfc3671.txt). Collective attributes enable you to define values that are assigned to attributes based on an entry's membership in a subentry.

### D.3.9 Common Development and Distribution License

The Common Development and Distribution License (CDDL) is an OSI-approved (http://www.opensource.org/) open source license which is used by the OpenDS project, on which Oracle Unified Directory.

The CDDL is a file-based license, which means that any changes to files contained in the project need to remain licensed under the CDDL. New files, however, may be licensed under any license chosen by the author (including closed-source licenses). The CDDL is based on the Mozilla Public License (MPL) and includes a patent grant clause so that any technology covered by patents will be granted to other projects using the code.

The CDDL license contents may be found at http://www.opensource.org/licenses/cddl1.php (http://www.opensource.org/licenses/cddl1.php).

### D.3.10 compare operation

The LDAP compare operation can be used to determine whether a specified entry contains a given attribute value. The compare request protocol op is defined as follows:

```
CompareRequest ::= [APPLICATION 14] SEQUENCE {
     entry           LDAPDN,
     ava             AttributeValueAssertion }

AttributeValueAssertion ::= SEQUENCE {
     attributeDesc   AttributeDescription,
     assertionValue  AssertionValue }
```

The elements of the request include the following:

- The DN of the entry in which the comparison is to be made.
- The name of the attribute in which the comparison is to be made.
- The assertion value to try to find in the specified attribute.

The response to an LDAP compare operation is an LDAP result element as defined below:

```
CompareResponse ::= [APPLICATION 15] LDAPResult
```

## D.3.11 connection handler

A connection handler is a component of the Directory Server that is responsible for accepting connections from clients, reading and parsing requests submitted by the clients, ensuring that they are processed by the server, and sending the corresponding responses back to the client. The connection handler manages all communication with the client and therefore needs to implement support for the associated protocol.

The directory server currently provides connection handlers capable of communicating using Lightweight Directory Access Protocol and Java Management Extensions, as well as a special connection handler for internal use that may be used to allow components of the server (like plug-in and other kinds of extensions) to perform operations. The server also provides an extensible connection handler API that may be used to implement support for additional network protocols.

## D.3.12 connection ID

A connection ID is a unique integer identifier that is assigned to each connection maintained within the Directory Server. It is used primarily for logging purposes, so that it is possible to correlate the various operations performed on a given connection.

The connection ID counter starts at zero for the first connection received by the server and increments by one for each additional connection. The counter is reset whenever the server is restarted.

Internal connections, which are used for processing internal operations, are assigned negative values to distinguish them from connections from external clients.

## D.3.13 control

An LDAP control is an element that may be included in an message. If it is included in a request message, it can be used to provide additional information about the way that the operation should be processed. If it is included in the response message, it can be used to provide additional information about the way the operation was processed.

Examples of LDAP controls include:

- account usability control - This is a pair of request and response controls that indicate whether an account is able to authenticate to the server.
- authorization identity control - This is a pair of request and response controls that may be used to determine the authorization identity for a user as part of a bind operation.
- entry change notification control - This is a control that is included in search result entry messages performed as part of a persistent search to indicate how an entry has been updated.
- get effective rights control - This is a request control that may be used to obtain information about what rights a user has for accessing a given entry.
- LDAP assertion control - This is a request control that may be used to ensure that an operation is only processed if the target entry matches a given assertion filter.

- LDAP no-op control - This is a request control that may be used to ensure that a write operation does not actually change any information in the server but attempts to determine whether the operation would otherwise be successful.
- LDAP post-read control - This is a pair of request and response controls that may be used to retrieve an entry as it appeared immediately after performing an add, modify, or modify DN operation.
- LDAP pre-read control - This is a pair of request and response controls that may be used to retrieve an entry as it appeared immediately before performing a delete, modify, or modify DN operation.
- manage DSA IT control - This is a request control that may be used to request that the server treat smart referrals as regular entries rather than as referrals.
- matched values control - This is a request control that may be used to request that entries returned from a search operation only include values matching a given filter.
- persistent search control - This is a request control that may be used to receive notification whenever an entry matching a given set of criteria is updated in the server.
- proxied authorization control - This is a request control that may be used to request that an operation be performed under the authorization of another user.
- server-side sort control - This is a request control that may be used to request that the server sort the results before returning them to the client.
- simple paged results control - This is a request control that may be used to request that the server retrieve only a subset of the results, and when used repeatedly can allow the client to page through the result set.
- virtual list view control - This is a pair of request and response controls that may be used to retrieve an arbitrary page of search results from the server.

An LDAP control is defined as follows:

```
Control ::= SEQUENCE {
     controlType             LDAPOID,
.... criticality             BOOLEAN DEFAULT FALSE,
.... controlValue            OCTET STRING OPTIONAL }
```

A control includes these elements:

- An object identifier that specifies the type of control.
- A criticality, which indicates whether the control should be considered a critical part of the operation (that is, if the server cannot process the control, the operation should fail).
- An optional value, which can be used to provide additional information about the way the control should be processed.

## D.3.14 CRAM-MD5 SASL mechanism

The CRAM-MD5 Simple Authentication and Security Layer mechanism provides a way for clients to authentication to the Directory Server with a username and password in a manner that does not expose the clear-text password, so it is significantly safer than simple authentication or the PLAIN SASL mechanism when the connection between the client and the server is not secure.

The CRAM-MD5 SASL mechanism is described in the draft-ietf-sasl-crammd5-10 (http://tools.ietf.org/html/draft-ietf-sasl-crammd5-10) Internet Draft. The process is as follows:

1. The client sends an message to the server with a bind request protocol op type using an authentication type of SASL with a mechanism name of CRAM-MD5 and no credentials.
2. The server sends a bind response message back to the client with a result code of 14 (SASL bind in progress) and a server SASL credentials element including randomly-generated data (the challenge).
3. The client responds with a second SASL bind request message to the server with a mechanism name of CRAM-M5, and this time provides SASL credentials containing the authentication ID used to identify the user and an MD5 digest that is computed by combining the server-provided challenge with the clear-text password.
4. The server uses the authentication ID to identify the user, and then retrieves the clear-text password for that user (if the clear-text password cannot be obtained, then authentication will fail) and uses it to determine whether the provided digest is valid. The server will then send an appropriate response to the client (usually with a result of either success or invalid credentials) indicating whether the authentication was successful.

The CRAM-MD5 SASL mechanism is very similar to DIGEST-MD5 SASL mechanism, but it is somewhat weaker because CRAM-MD5 only includes random data from the server whereas DIGEST-MD5 includes random data from both the client and the server. DIGEST-MD5 also provides a provision for ensuring connection integrity and/or confidentiality, which CRAM-MD5 does not offer.

### D.3.15 crypt algorithm

See UNIX crypt algorithm.

## D.4 D

### D.4.1 database

A database is a repository that is used for storing information. In the directory server, databases are used as the mechanism for storing data in a back end. The primary database used by the directory server is the Berkeley DB Java Edition, although it is possible to create other back ends with different backing stores.

### D.4.2 database cache

The database cache is a portion of memory that is reserved for holding content from the underlying database. Whenever an attempt is made to retrieve information from the database, the database will first check this cache before going to disk. The database cache can help significantly improve performance by avoiding costly disk I/O.

The database cache may be used either instead of or in addition to the server's entry cache. The database cache frequently creates a more compact representation of the data (which means that more data can be held in the cache in systems with limited memory), but the entry cache generally holds data in a format that can be more efficiently used by the server.

## D.4.3 debug log

The debug log provides a mechanism for obtaining information that may be used for debugging problems that might occur in the server. Debug information is generally data that is useful only in the event of a problem, and is frequently too voluminous to maintain under normal operations. The debug log may be used to report information like:

- Detailed information about exceptions thrown within the server
- Information about data read from or written to network clients
- Information about information read from or written to the database
- Information about decisions made in areas like access control or password policy processing

## D.4.4 delete operation

The LDAP delete operation can be used to remove an entry from the server (or when used in conjunction with the subtree delete control, a subtree). The delete request protocol op is defined as follows:

```
DelRequest ::= [APPLICATION 10] LDAPDN
```

The request includes only the DN of the entry to delete.

The response to an LDAP delete operation is an LDAP result element as defined below:

```
DelResponse ::= [APPLICATION 11] LDAPResult
```

## D.4.5 deprecated password storage scheme

A deprecated password storage scheme is a password storage scheme that is available for use in the server, but is intended primarily for transitional use. If a user has a password encoded with a deprecated storage scheme, then the user will be allowed to authenticate but the password will be re-encoded using the set of default storage schemes defined in the password policy.

This mechanism is primarily intended for cases in which data has been migrated into the directory server from another server uses a password storage scheme that you do not want to continue using (for example, because it is weaker than the default schemes). As users authenticate to the server, their passwords will be transitioned from the deprecated schemes to the default schemes.

## D.4.6 dereference policy

The dereference policy is an element of a search operation that specifies how the server should handle alias entries that may be encountered during search processing. Allowed alias dereference policy values include:

**neverDerefAliases**
The server should not attempt to dereference any aliases that it encounters during search processing.

**derefInSearching**
The server should dereference any entries within the scope of the search operation to determine whether they match the search criteria. The entry specified as the search base DN will not be dereferenced.

**derefFindingBaseObj**
The server should dereference the entry referenced as the search base DN if it is an alias, but any other alias entries within the scope of the search operation will not be dereferenced.

**derefAlways**
The server will dereference any alias entries within the scope of the search operation and will also dereference the base entry if it is an alias.

## D.4.7 DIGEST-MD5 SASL mechanism

The DIGEST-MD5 Simple Authentication and Security Layer mechanism provides a way for clients to authentication to the Directory Server with a username and password in a manner that does not expose the clear-text password, so it is significantly safer than simple authentication or the PLAIN SASL mechanism when the connection between the client and the server is not secure.

The DIGEST-MD5 SASL mechanism is described in RFC 2831 (`http://www.ietf.org/rfc/rfc2831.txt`), but a revised specification is contained in draft-ietf-sasl-rfc2831bis. The process is as follows:

1. The client sends an message to the server with a bind request protocol op type using an authentication type of `SASL` with a mechanism name of `DIGEST-MD5` and no credentials.
2. The server sends a bind response message back to the client with a result code of 14 (SASL bind in progress) and a server SASL credentials element including, among other things, some randomly-generated data (the nonce).
3. The client takes the nonce provided by the server, and some randomly generated data of its own (the cnonce), an authentication ID, an optional authorization ID, the user's clear-text password, and some other information and uses that to create an MD5 digest. The client then sends a second bind request message including that digest and some other clear-text information back to the server.
4. The server uses the authentication ID to identify the user, and then retrieves the clear-text password for that user (if the clear-text password cannot be obtained, then authentication will fail) and uses it to determine whether the provided digest is valid. The server will then send an appropriate response to the client (usually with a result of either `success` or `invalid credentials`) indicating whether the authentication was successful.
5. If the client requested a quality of protection (QoP) value indicating that the connection should be protected with integrity and/or confidentiality, then the server will initiate the necessary negotiation with the client. Note that at the present time, the directory server does not support the use of the DIGEST-MD5 mechanism with the use of integrity or confidentiality protection.

The DIGEST-MD5 SASL mechanism is very similar to CRAM-MD5 SASL mechanism, but it is somewhat strong because CRAM-MD5 includes only random data from the server whereas DIGEST-MD5 includes random data from both the client and the server. DIGEST-MD5 also provides a provision for ensuring connection integrity and/or confidentiality, which CRAM-MD5 does not offer.

## D.4.8 directory information tree

The directory information tree, or DIT, refers to the hierarchical structure of the data in a Directory Server. The DIT contains one or more naming context, which are the base entries for the server, and every other entry is descended from one of those naming

context entries. That is, a naming context entry is special in that it does not have a parent entry.

Consider a scenario, where the entry `dc=example,dc=com` is the naming context, and it has two immediate children, with DNs of `ou=People,dc=example,dc=com` and `ou=Groups,dc=example,dc=com`, respectively, and each of those entries has its own subordinate entries. There is no predefined limit to the maximum depth of a directory tree, and any entry can potentially have one or more subordinate entries. An entry that does not contain any subordinates is said to be a leaf entry, and any entry that has at least one subordinate entry is called a non-leaf entry.

## D.4.9 directory manager

The term `directory manager` is a common name used to refer to a root DN user in the Directory Server. It is so named because the default root user typically uses a bind distinguished name of `cn=Directory Manager`. Unlike many other types of directory servers, the directory server allows multiple root DNs to be defined, although the default root DN is still `cn=Directory Manager`.

## D.4.10 directory server

A directory server is a type of network daemon that stores data in a manner accessible to external clients. Directory servers typically use Lightweight Directory Access Protocol or Directory Services Markup Language for communicating with clients, although some servers use other protocols like DAP or NDS.

Directory servers store data in a hierarchical form (called the directory information tree) and provide the ability for clients to interact with that information, including:

- search operation, which make it possible to find all entry matching a given set of criteria
- add operation, which make it possible to add new entries to the server
- delete operation, which make it possible to remove entries from the server
- modify operation, which make it possible to update existing information in the server
- modify DN operation, which make it possible to rename entries in the server
- bind operation, which make it possible to authenticate users to the server
- compare operation, which make it possible to determine whether entries have a particular attribute-value pair

The directory server uses LDAPv3 for communicating with network clients, and provides a DSML gateway that can be used to handle DSML requests.

## D.4.11 directory server agent

A directory server agent (DSA) is a single instance of a directory server.

## D.4.12 Directory Services Markup Language

The Directory Services Markup Language (DSML) is a protocol that may be used to communicate with directory server. DSML is an alternative to Lightweight Directory Access Protocol, and uses an XML-based representation of requests and responses instead of the Basic Encoding Rules encoding that LDAP uses.

In general, DSML is seen as a relatively weak alternative to LDAP because it provides very little benefit and incurs a significant cost because the XML representation is much more verbose and expensive to process when compared with the BER encoding that LDAP uses. In most cases, it is recommended that LDAP be used instead of DSML to interact with the server.

## D.4.13 distinguished name

A distinguished name (often referred to as a `DN`) is a string that uniquely identifies an entry in the Directory Server. It consists of zero or more distinguished name (RDN) components that identify the location of the entry in the directory information tree. An entry's distinguished name can be thought of as a kind of an analog to an absolute path in a filesystem in that it specifies both the name and hierarchical location.

The RDN components for a distinguished name are separated by commas and are ordered from right to left. The rightmost components of a DN are closest to the server's naming context, and the leftmost components are closest to the leaf entry. That is, if you think of a directory hierarchy as a kind of pyramid with the naming context at the top and the branches descending downward, then the order of RDN components in a DN are listed from bottom to top.

Even though a DN consists of a series of RDN components, when one refers to an entry's RDN, then it is a reference to the leftmost RDN component. The attributes contained in an entry's RDN must also be contained in that entry.

In a DIT, the top entry is the naming context and its DN is `dc=example,dc=com`. To conserve space, only the RDNs of the subordinate entries are displayed, but the full DNs can be obtained by appending the RDN components from bottom to top. For example, the DN of the leftmost entry on the bottom row would be `uid=ann,ou=People,dc=example,dc=com`.

See RFC 4514 (`http://www.ietf.org/rfc/rfc4514.txt`) for more information about LDAP distinguished names and the way in which they should be represented as strings.

## D.4.14 distribution

Distribution is a proxy deployment type in which data is split into *partitions*. The split of data is determined by a distribution algorithm.

## D.4.15 DIT

See directory information tree.

## D.4.16 DIT content rule

A DIT content rule is a schema element that specifies which auxiliary object class are allowed to be used with an entry, as well as which attribute type are required, allowed, and prohibited for use with an entry, based on its structural object class.

The components of a DIT content rule definition include:

- The numeric object identifier of the structural object class with which the DIT content rule is associated.
- An optional set of names for the DIT content rule.
- An optional set of auxiliary object class names or OIDs for the auxiliary classes that are allowed to be used with entries containing the associated structural class.

- An optional set of attribute type names or OIDs for attribute types that are required to be present in entries with the associated structural class. These attributes will be required even if they are not allowed by any of the object classes in the entry.
- An optional set of attribute type names or OIDs for attribute types that may optionally be present in entries with the associated structural class. These attributes will be allowed even if they are not allowed by any of the object classes in the entry.
- An optional set of attribute type names or OIDs for attribute types that are prohibited to be present in entries with the associated structural class. These attributes will be prohibited even if they are allowed by any of the object classes in the entry.

The set of DIT content rules defined in the server may be determined by retrieving the `dITContentRules` attribute of the subschema subentry. For more information about DIT content rules, see Understanding DIT Content Rules.

## D.4.17 DIT structure rule

A DIT structure rule is a schema element that may be used to define the hierarchical relationships between entries. In particular, it defines the kinds of parent entries (based on their structural object class) that an entry with a given structural class is allowed to have.

The components of a DIT structure rule definition include:

- An integer rule ID value that is used to uniquely identify the rule.
- An optional set of names for the DIT structure rule.
- The name or object identifier of the name form with which the DIT structure rule is associated. The name form in turn links the DIT structure rule to a structural object class.
- An optional set of superior rule IDs. If a set of superior rules is defined, then they are used to define the structural classes below which the structural class associated with the rule's name form is allowed to exist.

The set of DIT structure rules defined in the server may be determined by retrieving the `dITStructureRules` attribute of the subschema subentry. For more information about DIT structure rules, see the Understanding DIT Structure Rules.

## D.4.18 DN

See distinguished name.

## D.4.19 DSA

See directory server agent.

## D.4.20 DSA-specific entry

A DSA-Specific Entry (DSE) is a special type of entry that provides information about a directory server agent, which is a synonym for directory server.

Lightweight Directory Access Protocol defines a special entry called the root DSE that provides information about the information contained in the server and the types of operations that it supports.

## D.4.21 DSE

See DSA-specific entry.

## D.4.22 DSML

See Directory Services Markup Language.

## D.4.23 DSML gateway

A DSML gateway (or DSML-to-LDAP gateway) is a special type of network daemon that is used to translate between Directory Services Markup Language and Lightweight Directory Access Protocol. In general, a DSML gateway accepts DSML requests from clients, converts them to LDAP requests that it forwards to a directory server for processing. It then translates the LDAP response from the directory server back to DSML to return to the client.

The directory server supports DSML through a DSML gateway, which is implemented as a Web application that can run in an application server.

## D.4.24 duration

Certain configuration properties take a duration as their allowed value.

A duration includes an integer, and a unit, specified in weeks (w), days (d), hours (h), minutes (m), seconds (s), or miliseconds (ms), or some combination with multiple specifiers. For example, you can specify one week as `1w`, `7d`, `168h`, `10080m`, or `604800s`. Or you can specify ten and a half days as `1w3d12h0m0s`.

Not all properties that require a duration support all duration specifiers (`w`, `d`, `h`, `m`, `s`, and `ms`).

A duration property can also include the following:

**base unit**
Specifies the minimum granularity that can be used to specify duration property values. For example, if the base unit is in seconds, values represented in milliseconds are not permitted.

**maximum unit (optional)**
Specifies the largest duration unit that can be used to specify duration property values. Values presented in units greater than this unit are not permitted.

**lower limit**
Specifies the smallest duration permitted by the property.

**upper limit (optional)**
Specifies the largest duration permitted by the property.

**unlimited duration**
Certain properties allow you to specify an unlimited duration. This is represented using the decoded value, -1, or the encoded string value `unlimited`.

## D.4.25 dynamic group

A dynamic group is a type of group in the directory server that defines its membership using a set of search criteria in the form of an LDAP URL, as opposed to a static group in which the distinguished name of the members are explicitly specified.

Dynamic groups provide an efficient way to manage groups with very large numbers of members. They are much more scalable than static groups, and their membership is automatically updated as entry change so that the match or no longer match the group criteria.

# D.5 E

## D.5.1 entry

An entry is the structure that holds information in a directory server. It consists of the following components:

- A distinguished name that uniquely identifies the entry among all other entries in the server.
- A collection of object class values that are used to govern the contents of the entry.
- A collection of attribute that contain the actual data for the entry.

An entry must always have exactly one structural object class that defines what type of entry it is. It may have zero or more auxiliary object class that may be used identify other characteristics for the entry. Together, the structural and auxiliary classes define a set of required attributes, which must be present in the entry, and optional attributes, which may be included in the entry but are not required.

## D.5.2 entry cache

The entry cache is a mechanism that uses system memory for holding entries in a manner that may be quickly accessed so that it is not necessary to decode them from the database whenever they are needed. Entry caching mechanisms are particularly effective when used with applications that access the same entry multiple times in a sequence of operations. For example, an application which first search operation to find a user entry and then bind operation as that user to verify a password, which is a very common usage pattern.

The entry cache may be used either instead of or in addition to the server's database cache. The database cache generally uses a more compact representation of the data, but the entry cache generally holds data in a format that can be more efficiently used by the server.

Unlike the database cache which is maintained by the underlying database, the entry cache is managed by the directory server itself. There are a number of different entry cache implementations that may be used.

## D.5.3 entry change notification control

The entry change notification control is a control that is included in search result entries returned to clients in response to a search operation that uses the persistent search control. This control contains additional information about the change made to the entry, including the type of change made, the change number (which corresponds to an item in the server's change log, if the server supports a change log), and, if the entry was renamed, the old DN of the entry. The control is described in draft-ietf-ldapext-psearch-03 (`http://tools.ietf.org/html/draft-ietf-ldapext-psearch-03`) and has an OID of 2.16.840.1.113730.3.4.7.

The control is defined as follows:

```
EntryChangeNotification ::= SEQUENCE {
          changeType ENUMERATED {
               add             (1),
               delete          (2),
               modify          (4),
               modDN           (8)
          },
          previousDN   LDAPDN OPTIONAL,     -- modifyDN ops. only
          changeNumber INTEGER OPTIONAL     -- if supported
}
```

### D.5.4 entryDN

An entryDN is an operational attribute that provides a copy of the entry's current distinguished name. Because a DN is not an attribute of the entry, it cannot be used to perform attribute value assertions. The entryDN provides a mechanism to access an entry's DN and is described in RFC 5020.

### D.5.5 entry ID

An entry ID is an integer value that is used to uniquely identify an entry in the Directory Server back end. Although the entry's distinguished name could be used for this purpose, the numeric entry ID is much more compact and more efficient to decode, so it is more appropriate for widespread use.

The entry ID is used as the key to the actual entry data in the id2entry database, and it is used in ID list to identify entries matching the associated index key.

### D.5.6 entryUUID

An entryUUID is a universally unique identifier that is contained in the `entryUUID` operational attribute and is assigned to each entry in the directory server. It is defined in RFC 4530 (`http://www.ietf.org/rfc/rfc4530.txt`) and it is intended to be a unique identifier that will not change over the life of the entry (as opposed to the distinguished name, which can change as a result of a modify DN operation). Because of the greater stability of the entryUUID, it is used by the replication subsystem to track entries even if the DN does change.

### D.5.7 equality index

An equality index is a type of indexwhich is used to identify efficiently which entries are exactly equal to a given assertion value. An equality index may only be maintained for attributes that have a corresponding equality matching rule. That matching rule will be used to normalized value to use as index keys, and the value for that key will be the ID list containing the entry ID of the entries with values that are equal to that normalized value.

### D.5.8 equality search filter

An equality search filter is a type of LDAP search filter that can be used to identify entries that contain a specific value for a given attribute. The server will use an equality matching rule to make the determination.

The string representation of an LDAP equality filter comprises an opening parenthesis followed by the attribute name, an equal sign, the attribute value, and the closing parenthesis. For example, an equality filter of `(uid=john.doe)` will match any entry in which the `uid` attribute contains a value of `john.doe`.

## D.5.9 error log

The error log provides a mechanism for reporting errors, warnings, and other significant events that happen in the life of the server. Each message written to the error log will include a category (indicating the area of the server in which the message was generated) and severity (indicating the relative importance of the message), along with an integer value that uniquely identifies the associated message string.

## D.5.10 export

See LDIF export.

## D.5.11 extended operation

The LDAP extended operation provides a degree of extensibility to the LDAP protocol by allowing clients to request operations not defined in the core protocol specification. Examples of LDAP extended operations include:

**cancel extended operation**
This operation may be used to cancel a previously-requested operation.

**Password Modify extended operation**
This operation may be used to change a user password.

**StartTLS extended operation**
This operation may be used to initiate a secure communication channel over an existing connection.

**"Who Am I?" extended operation**
This operation may be used to determine the authorization identity associated with the client connection.

The extended request protocol op is defined as follows:

```
ExtendedRequest ::= [APPLICATION 23] SEQUENCE {
     requestName      [0] LDAPOID,
     requestValue     [1] OCTET STRING OPTIONAL }
```

The elements of the extended request include:

- The object identifier that is used to indicate the type of operation to perform.
- An optional value containing additional information to use during the course of processing the request.

The response to an LDAP extended operation is defined as follows:

```
ExtendedResponse ::= [APPLICATION 24] SEQUENCE {
     COMPONENTS OF LDAPResult,
     responseName     [10] LDAPOID OPTIONAL,
     responseValue    [11] OCTET STRING OPTIONAL }
```

The extended response includes these elements:

- The elements of the result object.
- An optional OID used to indicate the type of response.
- An optional encoded value with additional information to include in the response.

### D.5.12 extensible match index

An extensible match index is a type of index that is used to help accelerate search operation using an extensible match search filter. Index keys are values that have been normalized value using a specified matching rule, and the corresponding ID list contains the entry ID for all entries that match the value according to that matching rule.

### D.5.13 extensible match search filter

An extensible match search filter is a type of LDAP search filter that can be used to identify matching entries using a specified matching rule.

An extensible matching filter contains the following components:

- The OID of the matching rule to use for the determination. This is an optional element, and if it isn't provided then the attribute type must be given and its default equality matching rule will be used.
- The name of the attribute type that will be targeted. If this is not provided, then all attributes contained in the entry will be examined.
- A flag that indicates whether the matching should be performed against the attributes of the entry's distinguished name *and* the attributes contained in the entry.
- An assertion value that should be used as the target for the matching rule.

The string representation of an LDAP extensible match filter comprises the following components in order:

- An opening parenthesis
- The name of the attribute type, or an empty string if none was provided
- The string `:dn` if the `dnAttributes` flag is set, or an empty string if not
- If a matching rule ID is available, then a string composed of a colon followed by that OID, or an empty string if there is no matching rule ID
- The string `:=`
- The string representation of the assertion value
- A closing parenthesis

### D.5.14 EXTERNAL SASL mechanism

The EXTERNAL Simple Authentication and Security Layer mechanism provides a way for clients to authentication to the Directory Server using information that is available outside of the communication performed at the LDAP protocol level. The most common use of EXTERNAL authentication (and at present, the only form that the directory server supports) is for the server to identify the client based on a certificate that the client presented during Secure Sockets Layer or StartTLS extended operation negotiation. The Directory Server will use a certificate mapper to map the client's certificate to a user in the directory, and may optionally perform additional validation (for example, ensuring that the presented certificate actually exists in the user's entry).

## D.6 F

### D.6.1 failover algorithm

A load balancing algorithm in which all client requests are sent to a main remote LDAP data source. If the main remote LDAP goes down, the request are forwarded to a secondary remote LDAP server, and so on. This ensures the continuation of the service after failure of one or more remote LDAP servers.

### D.6.2 false filter

See LDAP false filter.

## D.7 G

### D.7.1 generalized time

Generalized time is a form at may be used to represent time stamps, along with time zone information. A generalized time value contains the following components:

- Four digits to signify the year.
- Two digits to signify the month (01 for January, 02 for February,..., 12 for December).
- Two digits to signify the day of the month (01 through 28/29/30/31 depending on the month and whether it's a leap year).
- Two digits to signify the hour of the day (00 for midnight through 23 for 11 pm).
- An optional two digits that specify the minute of the hour (between 00 and 59).
- An optional two digits that specify the second of the minute (between 00 and 59, or 60 for leap seconds). This may only be included if the time stamp value also contains the minute of the hour.
- An optional period followed by one or more digits that specify the fraction of a second. This may only be included if the time stamp value contains minute and second information.
- A time zone indicator. This may be either the capital letter `Z` to indicate that the value is in the UTC time zone, or a plus or minus sign followed by two or four digits that specify the offset from UTC time zone.

An example of a time stamp in a generalized time format is `20070508200557Z`, which specifies a time (in the UTC time zone) of 8:05:57 PM on May 28, 2007. An equivalent value in the United States central daylight savings time (a five hour offset from UTC) would be `20070508150557-0500`.

### D.7.2 get effective rights control

The get effective rights control is a type of control that can be used to determine the rights that a given user has when interacting with a given entry. The control has an object identifier of 1.3.6.1.4.1.42.2.27.9.5.2 and uses the following definition:

```
GetRightsControl ::= SEQUENCE {
     authzId    authzId
     attributes  SEQUENCE OF AttributeType
}

-- Only the "dn:DN form is supported.
```

For an example of using this control in a search request, see Searching Using the Get Effective Rights Control.

### D.7.3 global index

In a proxy deployment, the global index maps the data entries to the *distribution partition* where the data is stored. Global indexes map a specific attribute (such as `telephonenumber`). For example, the global index could map `telephonenumber=5551212` to distribution partition 1, while `telephonenumber=4441212` to partition 2.

### D.7.4 global index catalog

A global index catalog contains one or more *global indexes*. A global index catalog can be used with a distribution deployment, in order to diminish the need for broadcasts, since the values of some attributes are mapped to the partition in which the entry is held.

### D.7.5 greater than or equal to search filter

An greater or equal search filter is a type of LDAP search filter that can be used to identify entries that contain a specific value for a given attribute that is greater than or equal to the provided assertion value. The server will use an ordering matching rule to make the determination.

The string representation of an LDAP greater or equal search filter comprises an opening parenthesis followed by the attribute name, a greater than sign, an equal sign, the assertion value, and the closing parenthesis. For example, a greater or equal filter of `(createTimestamp>=20070101000000Z)` will match any entry that has a `createTimestamp` value that is greater than or equal to `20070101000000Z`.

### D.7.6 group

A group is a special type of entry in the Directory Server that is used to represent a set of users in the server. Groups may be used within the server in a number of different ways, like access control and virtual attribute, and they may also be used by clients for various purposes.

There are several different types of groups defined in the server, including:

- static group provide an explicit list of members
- dynamic group obtain their membership information from a set of search criteria
- virtual static group appear to be static groups but obtain their membership information from another type of group, like a dynamic group

### D.7.7 GSSAPI SASL mechanism

The GSSAPI Simple Authentication and Security Layer mechanism provides a way for clients to authentication to the Directory Server using a Kerberos V5 session. Kerberos is a protocol that is commonly used for single sign-on purposes, and provides the option of using integrity and/or confidentiality to protect the communication between the client and the server (although the directory server does not at present support GSSAPI for protecting network content but only for authenticating clients).

The GSSAPI SASL mechanism is described in RFC 4752 (`http://www.ietf.org/rfc/rfc4752.txt`).

# D.8 I

## D.8.1 ID list

An ID list is used as the value of a Directory Server index. It contains a set of entry ID for all entries that match the associated index key.

In some cases, an ID list can have a special value that indicates that there are more entries matching the index key than allowed by the index entry limit. In that case, the index key will no longer be maintained.

## D.8.2 id2entry database

The id2entry database is a type of database that maps an entry ID to the contents of the corresponding entry. The entry ID is used in ID list within index.

## D.8.3 identity mapper

An identity mapper provides logic that can be used to map an authentication ID or authorization ID value to a corresponding user entry. Identity mappers are used in conjunction with a number of Simple Authentication and Security Layer mechanisms, as well as the proxied authorization control and the Password Modify extended operation.

## D.8.4 idle account lockout

Idle account lockout is a part of the Directory Server password policy that may be used to lock user accounts that remain unused for a significant period of time. It requires that the last login time feature be enabled so that user authentication times will be recorded, and any bind operation by a user that has not authenticated within a specified period of time will be rejected.

If a user's account has been locked due to remaining idle for too long, then it may be unlocked by an administrative password reset.

## D.8.5 in-core restart

An in-core restart is a process by which the server may be restarted without actually existing the JVM used to run the server. It can be used to apply any change that requires a server restart other than one that requires the modification of a JVM argument. An in-core restart may be faster than stopping and re-starting the server process, and it has the added benefit of maintaining the JIT cache that has been accumulated from observing processing performed within the JVM.

## D.8.6 index

An index is a mechanism used by the Directory Server database that can be used to efficiently find entries matching search criteria. An index maps a key to an ID list, which is the set of entry ID for the entries that match that index key.

The directory server uses six primary types of indexes:

- approximate index are used to identify entries containing attribute values approximately equal to a given assertion value.

- equality index are used to identify entries containing an attribute value that exactly matches a given assertion value.
- extensible match index are used to identify entries that match a given extensible match filter. This index is not currently supported.
- ordering index are used to identify entries that have values that are greater than or equal to, or less than or equal to, a given assertion value.
- presence index are used to identify entries that contain at least one value for a given attribute.
- substring index are used to identify entries that contain an attribute value matching a given substring assertion.

### D.8.7 index entry limit

The index entry limit is a configuration limit that can be used to control the maximum number of entries that is allowed to match any given index key (that is, the maximum size of an ID list). This provides a mechanism for limiting the performance impact for maintaining index keys that match a large percentage of the entries in the server. In cases where large ID lists might be required, performing an unindexed search can often be faster than one that is indexed.

The index entry limit in the directory server is analogous to the ALL IDs threshold in Oracle Directory Server Enterprise Edition.

### D.8.8 intermediate response

See LDAP intermediate response.

### D.8.9 Internet Draft

An Internet Draft is a form of specification defined through the IETF (`http://www.ietf.org/`). Internet drafts are short-lived specifications that typically go through multiple revisions, and may change significantly between revisions. Internet Drafts that reach a point of stability may be promoted to request for comments. Other drafts may stagnate and become no longer maintained, although in some cases they may still describe viable functionality that is worth implementing in the server.

## D.9 J

### D.9.1 Java Management Extensions

Java Management Extensions (JMX) is a framework is a Java technology that can be used for accessing monitoring and configuration information.

Oracle Unified Directory uses JMX for publishing information from monitor entry. It also uses the JMX notification mechanism for administrative alerts in the event of significant problems or events in the server.

### D.9.2 JMX

See Java Management Extensions.

# D.10 K

## D.10.1 key manager provider

A key manager provider is a component of the server that can provide access to private key information for server certificate.

The key manager providers available for use in the server include the following:

- A mechanism for accessing key information in a JKS keystore
- A mechanism for accessing key information in a PKCS#12 file
- A mechanism for accessing key information in a PKCS#11 token

# D.11 L

## D.11.1 last login time

The last login time feature of the Directory Server is a mechanism that can be used to write the time that the user last authentication to the server using a bind operation. The last login time may be written to a specified attribute with a user-defined format.

Note that in many servers, it may be desirable to define the last login time format to contain only the date but not the time of day. If this format is used, then the value will be only updated once per day, thereby reducing the potential impact on performance for users that authenticate several times throughout the day.

The last login time may be maintained for informational purposes, but it can also be used to enable the idle account lockout feature.

## D.11.2 lastmod plug-in

The lastmod plug-in is a pre-operation idle account lockout that can be used to add the `creatorsName` and `createTimestamp` attributes to an entry as part of an add operation, or update the `modifiersName` and `modifyTimestamp` attributes in an entry as part of a modify operation or modify DN operation operation.

## D.11.3 LDAP assertion control

The LDAP assertion control is a type of control that may be used to perform an operation only if the target entry matches a given assertion filter. It may be used in conjunction with compare operation, delete operation, modify operation, modify DN operation, and search operation.

The LDAP assertion control is described in RFC 4528 (`http://www.ietf.org/rfc/rfc4528.txt`) and has an OID of `1.3.6.1.1.12`. The value of the control should be encoded as an LDAP LDAP search filter.

For an example of using this control in a search request, see Searching Using the LDAP Assertion Control.

## D.11.4 ldapcompare command

The ldapcompare command can be used to request an LDAP compare operation.

For information about using this command, see ldapcompare.

## D.11.5 LDAP Data Interchange Format

The LDAP Data Interchange Format (LDIF) is a mechanism form representing directory data in text form. The LDIF specification is contained in RFC 2849 (http://www.ietf.org/rfc/rfc2849.txt) and describes a format not only for representing directory data but also a mechanism for making changes to that data.

In general, an LDIF record consists of a series of name-value pairs. The name can be followed by a single colon, zero or more spaces, and associated value, or it can be followed by two colons, zero or more spaces, and the base64 encoding representation of the value. Each name-value pair is given on a separate line, and long lines may be wrapped onto two or more lines using an end-of-line character followed by exactly one space at the beginning of the next line. LDIF records should be separated from each other by at least one blank line. Any line that begins with an octothorpe (#) character will be treated as a comment and ignored.

For an LDIF representation of an entry, the first line should contain the distinguished name of the entry. The remaining lines of the LDIF record will represent the attribute of the entry, with the attribute description used as the name. Multivalued attributes will be represented with a separate line per value.

The following provides an example of a user entry represented in the LDAP Data Interchange Format:

```
dn: uid=john.doe,ou=People,dc=example,dc=com
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
uid: john.doe
givenName: John
sn: Doe
cn: John Doe
mail: john.doe@example.com
userCertificate;binary:: MIIB5TCCAU6gAwIBAgIERloIajANBgkqhkiG9w0BAQUFADA3M
 QswCQYDVQQGEwJVUzEVMBMGA1UEChMMRXhhbXBsZSBDb3JwMREwDwYDVQQDEwhKb2huIERvZT
 AeFw0wNzA1MjcyMjM4MzRaFw0wNzA4MjUyMjM4MzRaMDcxCzAJBgNVBAYTAlVTMRUwEwYDVQQ
 KEwxFeGFtcGxlIENvcnAxETAPBgNVBAMTCEpvaG4gRG9lMIGfMA0GCSqGSIb3DQEBAQUAA4GN
 ADCBiQKBgQCWNZB4qs1UvjYgvGvB9udmiUi4X4DeaSm3o0p8PSwpOFxSqgWdSwKgUugZ1EJVy
 YoakljDFsJ0GVown+dIB24V4ozNs6wa0YotIKTV2AcySQkmzzP3e+OnE9Aa1wlB/PVnh1CFLg
 k1UOoruLE10bac5HA8QiAmfNMorU26AwFTcwIDAQABMA0GCSqGSIb3DQEBBQUAA4GBAGrzMKN
 bBRWn+LIfYTfqKYUc258XVbhFri1OV0oF82vyvciYWZzyxLc52EPDsymLmcDh+CdWxy3bVkjd
 Mg1WEtMGr1GsxOVi/vWe+kT4tPhinnB4Fowf8zgqiUKo9/FJN26y7Fpvy1IODiBInDrKZRvNf
 qemCf7o3+Cp00OmF5ey
userPassword: {SSHA}s4Bd9M0tCpRDr8/U+IXetRcAbd8bJY3AFKsn+A==
```

To represent an LDAP add operation in LDIF, the format is exactly the same as to represent an entry, with the exception that the line immediately after the DN should indicate a `changetype` of `add`, as shown in the following example:

```
dn: uid=john.doe,ou=People,dc=example,dc=com
changetype: add
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
uid: john.doe
givenName: John
```

```
sn: Doe
cn: John Doe
mail: john.doe@example.com
userCertificate;binary:: MIIB5TCCAU6gAwIBAgIERloIajANBgkqhkiG9w0BAQUFADA3M
 QswCQYDVQQGEwJVUzEVMBMGA1UEChMMRXhhbXBsZSBDb3JwMREwDwYDVQQDEwhKb2huIERvZT
 AeFw0wNzA1MjcyMjM4MzRaFw0wNzA4MjUyMjM4MzRaMDcxCzAJBgNVBAYTAlVTMRUwEwYDVQQ
 KEwxFeGFtcGxlIENvcnAxETAPBgNVBAMTCEpvaG4gRG9lMIGfMA0GCSqGSIb3DQEBAQUAA4GN
 ADCBiQKBgQCWNZB4qs1UvjYgvGvB9udmiUi4X4DeaSm3o0p8PSwpOFxSqgWdSwKgUugZ1EJVy
 YoakljDFsJ0GVown+dIB24V4ozNs6wa0YotIKTV2AcySQkmzzP3e+0nE9Aa1wlB/PVnh1CFLg
 k1UOoruLE10bac5HA8QiAmfNMorU26AwFTcwIDAQABMA0GCSqGSIb3DQEBBQUAA4GBAGrzMKN
 bBRWn+LIfYTfqKYUc258XVbhFri1OV0oF82vyvciYWZzyxLc52EPDsymLmcDh+CdWxy3bVkjd
 Mg1WEtMGr1GsxOVi/vWe+kT4tPhinnB4Fowf8zgqiUKo9/FJN26y7Fpvy1IODiBInDrKZRvNf
 qemCf7o3+Cp00OmF5ey
userPassword: password
```

To represent an LDAP delete operation in LDIF, the format is simply a line containing the DN of the entry followed by a line indicating a `changetype` of `delete`, like:

```
dn: uid=john.doe,ou=People,dc=example,dc=com
changetype: delete
```

To represent an LDAP modify operation in LDIF, the format is a little more complex. The first line should contain the DN of the entry, and the second should contain a `changetype` of `modify`. The third line should specify the attribute modification type (add, delete, replace, or increment) followed by the attribute description, and there may be additional lines that specify specific values for that change, with the name portion being the attribute description and the value being the corresponding attribute value. There may be multiple attribute modifications described in a single modify change record, with each of them separated by a line containing only a dash, as shown in the following example:

```
dn: uid=john.doe,ou=People,dc=example,dc=com
changetype: modify
replace: userPassword
userPassword: newpassword
-
replace: description
description: This is the first description value
description: This is the second description value
```

To represent an LDAP modify DN operation in LDIF, the first line should contain the DN of the entry, and the second line should contain a `changetype` of `moddn`. The third line should have a name of `newrdn` with a value equal to the new RDN to assign to the entry, and the fourth should have a name of `deleteoldrdn` followed by a value of either 1 (if the `deleteOldRDN` flag should be `true`) or `0` (if it should be false). There can be an optional fifth line with a name of `newsuperior` and a value of the new superior DN if one is included in the request. For example:

```
dn: uid=john.doe,ou=People,dc=example,dc=com
changetype: moddn
newrdn: uid=johnathan.doe
deleteoldrdn: 1
```

## D.11.6 ldapdelete command

The ldapdelete command can be used to request an LDAP delete operation.

For information about using this command, see ldapdelete.

## D.11.7 LDAP false filter

An LDAP false filter is a special type of OR search filter that does not contain any embedded filter components. It is called an "LDAP false filter" because it always evaluates to `false` and will never match any entry.

The string representation for an LDAP true filter is `(|)`. LDAP false filters are described in RFC 4526 (`http://www.ietf.org/rfc/rfc4526.txt`).

## D.11.8 LDAP intermediate response

The LDAP intermediate response message is a special type of protocol op that allows the server to send additional messages providing information about the state of an operation before it has completed processing and the final response message is sent. Prior to the introduction of the intermediate response in RFC 3771 (`http://www.ietf.org/rfc/rfc3771.txt`), only search operations were allowed to send multiple responses.

The intermediate response protocol op is defined as follows:

```
IntermediateResponse ::= [APPLICATION 25] SEQUENCE {
          responseName     [0] LDAPOID OPTIONAL,
          responseValue    [1] OCTET STRING OPTIONAL }
```

At present, the directory server does not support any operations that make use of intermediate response messages.

## D.11.9 LDAP message

The LDAP message is the fundamental protocol data unit for LDAP communication. It is the container that is used to hold all request and response elements.

The LDAP message is defined as shown in the following example:

```
LDAPMessage ::= SEQUENCE {
     messageID       MessageID,
     protocolOp      CHOICE {
          bindRequest           BindRequest,
          bindResponse          BindResponse,
          unbindRequest         UnbindRequest,
          searchRequest         SearchRequest,
          searchResEntry        SearchResultEntry,
          searchResDone         SearchResultDone,
          searchResRef          SearchResultReference,
          modifyRequest         ModifyRequest,
          modifyResponse        ModifyResponse,
          addRequest            AddRequest,
          addResponse           AddResponse,
          delRequest            DelRequest,
          delResponse           DelResponse,
          modDNRequest          ModifyDNRequest,
          modDNResponse         ModifyDNResponse,
          compareRequest        CompareRequest,
          compareResponse       CompareResponse,
          abandonRequest        AbandonRequest,
          extendedReq           ExtendedRequest,
          extendedResp          ExtendedResponse,
          ...,
```

```
          intermediateResponse  IntermediateResponse },
     controls       [0] Controls OPTIONAL }
```

The LDAP message includes these elements:

- The message ID, which is the unique identifier that is used to correlate requests and responses. The client includes a message ID in the request, and all response messages for that request will have the same message ID.
- The protocol op, which is the container for the actual request or response.
- An optional set of control that can be used to provide additional information about the way that the request should be processed, or additional information about the response from the server.

## D.11.10 LDAP modify DN operation

The LDAP modify DN operation can be used to change the distinguished name of an entry in the Directory Server. It can alter the relative distinguished name of the entry and/or it can move the entry below a new parent. If the target entry has subordinate entries, then it may be used to move or rename that subtree.

The modify DN request protocol op is defined as follows:

```
ModifyDNRequest ::= [APPLICATION 12] SEQUENCE {
     entry           LDAPDN,
     newrdn          RelativeLDAPDN,
     deleteoldrdn    BOOLEAN,
     newSuperior     [0] LDAPDN OPTIONAL }
```

The modify DN request includes these elements:

- The DN of the entry to rename and/or move.
- The new RDN to use for the entry. If the entry is simply to be moved below a new parent, then it may be the same as the current RDN.
- A flag that indicates whether the current RDN attribute values should be removed from the entry.
- An optional DN specifying the new parent for the entry.

The response to an LDAP modify DN operation is an LDAP result as defined as follows:

```
ModifyDNResponse ::= [APPLICATION 13] LDAPResult}
```

## D.11.11 LDAP modify operation

The LDAP modify operation can be used to alter an existing entry in the Directory Server. The modify request protocol op is defined as follows:

```
ModifyRequest ::= [APPLICATION 6] SEQUENCE {
     object          LDAPDN,
     changes         SEQUENCE OF change SEQUENCE {
          operation       ENUMERATED {
               add     (0),
               delete  (1),
               replace (2),
               ...  },
          modification    PartialAttribute } }
```

The modify request includes these elements:

- The DN of the entry to modify
- One or more modification elements indicating the changes to make in the entry

The response to an LDAP modify operation is an LDAP result defined as shown here:

```
ModifyResponse ::= [APPLICATION 7] LDAPResult
```

## D.11.12 ldapmodify command

The ldapmodify command may be used to request LDAP add operation, delete operation, modify operation, and modify DN operation operations.

For information about using this command, see ldapmodify.

## D.11.13 LDAP no-op control

The LDAP no-op control is a type of control that may be attached to an LDAP add operation, delete operation, modify operation, or modify DN operation to indicate that it should not actually make any change to the content in the server.

The LDAP no-op control is defined in draft-zeilenga-ldap-noop. This is a specification that is still in progress, but the directory server does provide basic support for this control using an object identifier of 1.3.6.1.4.1.4203.1.10.2. The control does not have a value.

The following example shows the use of the no-op control in an `ldapmodify` operation.

```
ldapmodify -h localhost -p 1389 -D "cn=directory manager" -j pwd-file \
-J 1.3.6.1.4.1.4203.1.10.2
dn: uid=aaltay,ou=People,dc=example,dc=com
changetype: modify
replace: telephoneNumber
telephoneNumber: +1 995 589 3333

Processing MODIFY request for uid=aaltay,ou=People,dc=example,dc=com
MODIFY operation failed
Result Code:  16654 (No Operation)
Additional Information:  The modify operation was not actually performed in the
Directory Server back end because the LDAP no-op control was present in the
request
```

## D.11.14 LDAP post-read control

The LDAP post-read control is a type of control that may be attached to an LDAP add operation, modify operation, or modify DN operation operation to request that the server return a copy of the target entry exactly as it was at the end of the processing for that operation. It is one of the LDAP read entry controls defined in RFC 4527 (`http://www.ietf.org/rfc/rfc4527.txt`).

The post-read request control has an OID of 1.3.6.1.1.13.2, and the value should be encoded in the same way as the search attributes in a search operation. The response control has an OID of 1.3.6.1.1.13.2 (the same as the OID for the request control), and the value should be encoded in the same was as a search result entry.

The following example shows the use of the post-read control in an `ldapmodify` request:

```
$ ldapmodify -h localhost -p 1389 -D "cn=directory manager" -j pwd-file \
--postReadAttributes=telephoneNumber
dn: uid=aaltay,ou=People,dc=example,dc=com
changetype: modify
replace: telephoneNumber
telephoneNumber: +1 995 589 3333

Processing MODIFY request for uid=aaltay,ou=People,dc=example,dc=com
MODIFY operation successful for DN uid=aaltay,ou=People,dc=example,dc=com
Target entry after the operation:
dn: uid=aaltay,ou=People,dc=example,dc=com
telephoneNumber: +1 995 589 3333
```

## D.11.15 LDAP pre-read control

The LDAP pre-read control is a type of control that may be attached to an LDAP delete operation, modify operation, or modify DN operation operation to request that the server return a copy of the target entry exactly as it was immediately before the processing for that operation. It is one of the LDAP read entry controls defined in RFC 4527 (http://www.ietf.org/rfc/rfc4527.txt).

The pre-read request control has an OID of 1.3.6.1.1.13.1, and the value should be encoded in the same way as the search attributes in a search operation. The response control has an OID of 1.3.6.1.1.13.1 (the same as the OID for the request control), and the value should be encoded in the same was as a search result entry.

The following example shows the use of the pre-read control in an `ldapmodify` request:

```
$ ldapmodify -h localhost -p 1389 -D "cn=directory manager" -j pwd-file \
--preReadAttributes=telephoneNumber
dn: uid=aaltay,ou=People,dc=example,dc=com
changetype: modify
replace: telephoneNumber
telephoneNumber: +1 995 589 4444

Processing MODIFY request for uid=user.199,ou=People,dc=exampele,dc=com
MODIFY operation successful for DN uid=aaltay,ou=People,dc=example,dc=com
Target entry before the operation:
dn: uid=aaltay.199,ou=People,dc=example,dc=com
telephoneNumber: +1 995 589 3333
```

## D.11.16 LDAP result

The LDAP result element is a generic protocol op that is used for the responses of several types of LDAP operations. The basic definition for the LDAP result is as follows:

```
LDAPResult ::= SEQUENCE {
     resultCode         ENUMERATED {
          success                      (0),
          operationsError              (1),
          protocolError                (2),
          timeLimitExceeded            (3),
          sizeLimitExceeded            (4),
          compareFalse                 (5),
          compareTrue                  (6),
          authMethodNotSupported       (7),
```

```
     strongerAuthRequired         (8),
          -- 9 reserved --
     referral                     (10),
     adminLimitExceeded           (11),
     unavailableCriticalExtension (12),
     confidentialityRequired      (13),
     saslBindInProgress           (14),
     noSuchAttribute              (16),
     undefinedAttributeType       (17),
     inappropriateMatching        (18),
     constraintViolation          (19),
     attributeOrValueExists       (20),
     invalidAttributeSyntax       (21),
          -- 22-31 unused --
     noSuchObject                 (32),
     aliasProblem                 (33),
     invalidDNSyntax              (34),
          -- 35 reserved for undefined isLeaf --
     aliasDereferencingProblem    (36),
          -- 37-47 unused --
     inappropriateAuthentication  (48),
     invalidCredentials           (49),
     insufficientAccessRights     (50),
     busy                         (51),
     unavailable                  (52),
     unwillingToPerform           (53),
     loopDetect                   (54),
          -- 55-63 unused --
     namingViolation              (64),
     objectClassViolation         (65),
     notAllowedOnNonLeaf          (66),
     notAllowedOnRDN              (67),
     entryAlreadyExists           (68),
     objectClassModsProhibited    (69),
          -- 70 reserved for CLDAP --
     affectsMultipleDSAs          (71),
          -- 72-79 unused --
     other                        (80), ...  },
matchedDN          LDAPDN,
diagnosticMessage  LDAPString,
referral           [3] Referral OPTIONAL }
```

The elements of the LDAP result are:

**result code**
An integer value that provides generic information about the result of the operation. The definition above specifies several result codes, but a number of other values are defined in other specifications.

**matched DN**
A DN value that may specify the DN of the closest superior entry found if the request specified an entry that did not exist. It may be an empty DN if the matched DN element is not appropriate for the response.

**Diagnostic Message**
A human-readable message that provides additional information about the result of the processing. It is typically used for error messages, but it may also be present in successful operations. It may be an empty string if there is no message.

**referral**
A set of LDAP URLs to other servers in which the client may attempt the operation. This element may be absent if there are no referrals.

## D.11.17 LDAPS

LDAPS is a term that is used to refer to Lightweight Directory Access Protocol communication over Secure Sockets Layer.

## D.11.18 LDAP search filter

A search filter provides a mechanism for defining the criteria for defining matching entries in an LDAP search operation. There are ten different types of search filters defined in LDAP:

**AND search filter**
Serve as a container for holding zero or more search filter elements. All search filters contained in the AND filter must match the target entry for the AND filter to match.

**OR search filter**
Serve as a container for holding zero or more search filter elements. At least one of the search filters contained in the OR filter must match the target entry for the OR filter to match.

**NOT search filter**
Serves as a container for exactly one search filter element. The embedded filter must not match the target entry for the NOT filter to match.

**equality search filter**
Provides a mechanism for identifying entries that contain a specified value for a given attribute.

**substring search filter**
Provides a mechanism for identifying entries with attribute values matching a specified substring.

**greater than or equal to search filter**
Provides a mechanism for identifying entries with attribute values greater than or equal to a specific value.

**less than or equal to search filter**
Provides a mechanism for identifying entries with attribute values less than or equal to a specific value.

**presence search filter**
Provides a mechanism for identifying entries that contain at least one value for a specified attribute.

**approximate search filter**
Provides a mechanism for identifying entries with attribute values that are approximately equal to a given value.

**extensible match search filter**
Provides a mechanism for using a matching rule to identify matching entries using an extensible mechanism.

See RFC 4515 (`http://www.ietf.org/rfc/rfc4515.txt`) for more information about LDAP search filters and a mechanism for representing them as strings.

## D.11.19 ldapsearch command

The `ldapsearch` command can be used to request an LDAP search operation.

For information about using this command, see ldapsearch.

## D.11.20 LDAP true filter

An LDAP true filter is a special type of AND search filter that does not contain any embedded filter components. It is called an "LDAP true filter" because it always evaluates to `true` and will match any entry.

The string representation for an LDAP true filter is `(&)`. LDAP true filters are described in RFC 4526 (`http://www.ietf.org/rfc/rfc4526.txt`).

## D.11.21 LDAP Subentry

An LDAP subentry is a type of entry that contains the `ldapSubEntry` object class. These entries are meant to hold operational data for the server. They are kind of like operational attribute in that they are not returned to clients unless explicitly requested by including a request control with an OID of 1.3.6.1.4.1.7628.5.101.1 and no value. This behavior is described in draft-ietf-ldup-subentry.

For an example of using this control in a search, see Searching Using the LDAP Subentry Control.

## D.11.22 LDAP URL

An LDAP URL is a type of URL that may be used to reference an entry or set of search criteria. The format of an LDAP URL is described in RFC 4516 (`http://www.ietf.org/rfc/rfc4516.txt`) and may include the following elements:

- The address of the directory server
- The port number of the directory server
- The search base DN
- A set of search attributes
- The search scope for the search
- A LDAP search filter for identifying the entries to match
- A set of extensions that provide information about the way in which the search should be processed

All of these elements are optional. Technically, all that is required of an LDAP URL is the string `ldap://`. However, a more complete URL might be `ldap://directory.example.com:389/dc=example,dc=com?cn,givenName,sn?sub?(uid=john.doe)`.

## D.11.23 LDIF export

An LDIF export operation is a process by which all or part of the content in a Directory Server back end is written to a file using the LDAP Data Interchange Format. An LDIF export can be initiated using the export-ldif command or an LDIF export task.

## D.11.24 LDIF import

An LDIF import operation is a process by which data can be added to a Directory Server back end from a file with information in the LDAP Data Interchange Format. An LDIF import provides a significantly more efficient means of adding a large number of entries to the server than LDAP add operation.

An LDIF import operation can be initiated using the import-ldif command or with the LDIF import task.

## D.11.25 leaf entry

A leaf entry is an entry that does not have any subordinate entries in the server.

## D.11.26 less than or equal to search filter

An less or equal search filter is a type of LDAP search filter that can be used to identify entries that contain a specific value for a given attribute that is less than or equal to the provided assertion value. The server will use an ordering matching rule to make the determination.

The string representation of an LDAP less or equal search filter is composed of an opening parenthesis followed by the attribute name, a less than sign, an equal sign, the assertion value, and the closing parenthesis. For example, a less or equal filter of `(createTimestamp<=20070101000000Z)` will match any entry that has a `createTimestamp` value that is less than or equal to `20070101000000Z`.

## D.11.27 lexico algorithm

A proxy distribution algorithm, in which the data is split into partitions based on alphabetical delimitations. For example, [A-E[ for one partition and [E-H[ for the next partition.

## D.11.28 Lightweight Directory Access Protocol

The Lightweight Directory Access Protocol (LDAP) is a protocol that may be used to communicate with a directory server. It is an open standard that uses the Basic Encoding Rules subset of Abstract Syntax Notation One to encode communication into message.

The core LDAPv3 specification is in RFC 4510 (`http://www.ietf.org/rfc/rfc4510.txt`), with RFC 4511 (`http://www.ietf.org/rfc/rfc4511.txt`) defining the actual encoding for the protocol. A number of other specifications are defined in a number of request for comments and Internet Draft.

LDAP defines a number of different types of operations, including:

**abandon operation**
Provides a way to abort the processing for an operation in progress

**add operation**
Provides a way to add a new entry to the server

**bind operation**
Provides a way to authentication to the server

**compare operation**
Provides a way to determine whether an entry has a specified attribute value assertion

**delete operation**
Provides a way to remove entries from the server

**extended operation**
Provides a way to perform custom processing implemented as an extension to the core LDAP protocol

**modify operation**
Provides a way to alter the contents of an entry in the server

**modify DN operation**
Provides a way to rename an entry in the server

**search operation**
Provides a way to identify all entries that match a given set of criteria

**unbind operation**
Provides a way to indicate that the client wishes to disconnect from the server

### D.11.29 load balancing

Load balancing is a proxy deployment type which provides single access to a set of replicated remote LDAP servers. The choice of the remote LDAP server to which a client requests is sent is determined by a load balancing algorithm.

### D.11.30 lookthrough limit

The lookthrough limit is a configuration option within the Directory Server that can be used to enforce a limit on the number of entries that the server will examine in the course of processing a search operation. This limit applies to all entries that the server examines, regardless of whether it matches the provided search criteria.

The lookthrough limit configuration attribute can be used to limit the impact of unindexed search, or searches with a very large candidate list.

For information about configuring the lookthrough limit, see Setting Resource Limits on a User Account and Setting Root User Resource Limits.

## D.12 M

### D.12.1 MakeLDIF command

The MakeLDIF command provides a mechanism for generating entry in LDAP Data Interchange Format form. The entries will be generated based on a template containing a number of tags that can be used to control the way that the data is generated.

For information about using this command, see make-ldif. Creating MakeLDIF Template Files describes the valid structure and content for MakeLDIF template files.

### D.12.2 manage DSA IT control

The Manage DSA IT control is a type of control that can be used to request that the server treat smart referral as regular entries. It can be attached to a delete operation, modify operation, or modify DN operation operation to request that the server apply the operation to the entry containing the smart referral rather than sending the referral

back to the client. It may also be attached to a search operation to indicate that the server should return the entries containing the smart referrals as search result entry rather than search result reference.

The Manage DSA IT control is defined in RFC 3296 (http://www.ietf.org/rfc/rfc3296.txt). It has an object identifier of 2.16.840.1.113730.3.4.2 with no value.

For an example of using this control in a search request, see Searching Using the Manage DSA IT Control.

## D.12.3 matched DN

A matched DN is an element of an LDAP result object that can provide additional information about the closest matching entry found in the server. It is generally used when a request targets an entry that does not exist, in which case the matched DN should contain the distinguished name of an entry that does exist in the server and is the closest ancestor of the specified entry. For example, if an operation targeted an entry uid=doesnt.exist,ou=People,dc=example,dc=com that did not exist but the entry ou=People,dc=example,dc=com does exist in the server, then that may be returned as the matched DN.

There is no guarantee that a matched DN is returned from an operation targeting an entry that does not exist, in which case the matched DN element of the LDAP result will be an empty string. This may be used, for example, if the request targeted an entry that does not have any hierarchical relationship with any other entry in the server.

## D.12.4 matched values control

The matched values control is a type of control that can be attached to a search operation to indicate that only values matching a specified filter should be included in entries returned to the client. It is described in RFC 3876 (http://www.ietf.org/rfc/rfc3876.txt).

The request control should have an OID of 1.2.826.0.1.3344810.2.3. The value should be encoded as follows:

```
          ValuesReturnFilter ::= SEQUENCE OF SimpleFilterItem

          SimpleFilterItem ::= CHOICE {
               equalityMatch   [3] AttributeValueAssertion,
               substrings      [4] SubstringFilter,
               greaterOrEqual  [5] AttributeValueAssertion,
               lessOrEqual     [6] AttributeValueAssertion,
               present         [7] AttributeDescription,
               approxMatch     [8] AttributeValueAssertion,
               extensibleMatch [9] SimpleMatchingAssertion }

          SimpleMatchingAssertion ::= SEQUENCE {
               matchingRule    [1] MatchingRuleId OPTIONAL,
               type            [2] AttributeDescription OPTIONAL,
--- at least one of the above must be present
               matchValue      [3] AssertionValue}
```

There is no corresponding response control.

For an example of using this control in a search request, see Searching Using the Matched Values Filter Control.

## D.12.5 matching rule

A matching rule is a schema element that defines how the server should interact with values of an attribute. There are three standard types of matching rules:

- Equality matching rules are used to determine whether one attribute value is equal to another. This determination is generally made based on the normalized value, and ignores insignificant differences (for example, differences in capitalization or extra spaces).
- Ordering matching rules are used to determine the relative order between two values in a sorted list. This is used when performing server-side sort control, but it is also used for greater than or equal to search filter and less than or equal to search filter filter components.
- Substring matching rules are used to determine whether a value contains a given substring search filter.

In addition to these standard matching rules, the directory server defines a fourth type, approximate matching rules, which are used to determine whether one value is approximately equal to another. The definition of "approximately equal to" can vary, but one common use is "sounds like".

Common examples of matching rules include:

**booleanMatch**
An equality matching rule that determines whether two Boolean values are equal to each other.

**caseExactMatch**
An equality matching rule that determines whether two string values are equal to each other, without ignoring differences in capitalization.

**caseExactOrderingMatch**
An ordering matching rule that is used to determine the relative order between two string values, without ignoring differences in capitalization.

**caseExactSubstringsMatch**
A substring matching rule that is used to determine whether a string value contains a given substring, without ignoring differences in capitalization.

**caseIgnoreMatch**
An equality matching rule that determines whether two string values are equal to each other, ignoring differences in capitalization.

**caseIgnoreOrderingMatch**
An ordering matching rule that is used to determine the relative order between two string values, ignoring differences in capitalization.

**caseIgnoreSubstringsMatch**
A substring matching rule that is used to determine whether a string value contains a given substring, ignoring differences in capitalization.

**distinguishedNameMatch**
An equality matching rule that determines whether two distinguished name are equal to each other, ignoring extra spaces around commas separating RDN components and equal signs separating RDN names from values. The individual RDN values will be compared based on the matching rules associated with the corresponding RDN attributes.

**generalizedTimeMatch**
An equality matching rule that determines whether two generalized time values are equal to each other.

**generalizedTimeOrderingMatch**
An ordering matching rule that is used to determine the relative order between two generalized time values.

**integerMatch**
An equality matching rule that determines whether two integer values are equal to each other.

**integerOrderingMatch**
An ordering matching rule that is used to determine the relative order between two integer values.

**octetStringMatch**
An equality matching rule that determines whether two values are exactly equal to each other using a byte-for-byte comparison.

In most cases, the directory server will use matching rules in a completely "behind the scenes" manner without the client needing to know about it. Whenever the client references a given attribute type, then the server will automatically know to use the appropriate matching rules for that attribute. However, it is also possible for the client to request that the server use a specific matching rule when performing an operation through the use of an extensible match search filter.

The set of matching rules defined in the server may be determined by retrieving the `matchingRules` attribute of the subschema subentry. For more information about matching rules, see Understanding Matching Rules.

## D.12.6 matching rule use

A matching rule use is a schema element that can be used to determine which attribute type can be used in conjunction with a given matching rule. Note that this only applies when using extensible match search filter.

A matching rule use definition includes an object identifier for the matching rule that it applies to and a list of the names or OIDs of the attribute types that may be used in conjunction with that matching rule. If an attribute is not included in this list, then it cannot be used in conjunction with the associated matching rule. If there is no matching rule use defined for a given matching rule, then it should be assumed that the matching rule can be used with any attribute type.

The set of matching rule uses defined in the server may be determined by retrieving the `matchingRuleUse` attribute of the subschema subentry. For more information about matching rule uses, see Understanding Matching Rule Uses

## D.12.7 MD5

MD5 is a one-way message digest algorithm defined in RFC 1321 (`http://www.ietf.org/rfc/rfc1321.txt`). It can be used to encode a value of an arbitrary length into a 128-bit value that cannot be reversed to determine the original cleartext. It is commonly used as a mechanism for checksumming data, and it is also commonly used for encoding passwords and other sensitive information.

Note that recent advances in cryptography have discovered weaknesses in the MD5 algorithm. These discoveries do not directly impact the security of the way that the

MD5 algorithm is used by the directory server, but nevertheless it may be wise to use a stronger mechanism like the Secure Hash Algorithm.

## D.12.8 message

See LDAP message.

## D.12.9 message ID

The message ID is an integer value that is contained in the message and is used to correlate request and response messages. The client chooses a message ID value to include in the request message, and the server will use the same message ID in all response messages. This makes it possible for the client to have multiple requests in progress on the same connection at any given time. All requests in progress at any given time must have different message IDs. The client will typically keep a sequentially-increasing counter for all request messages so that each request gets a different message ID than the last.

Note that unsolicited notification messages will always have a message ID value of zero. All other LDAP messages should have a message ID value between 1 and 2147483647.

## D.12.10 modification

A modification is an element of an LDAP modify operation that describes a change to a single attribute. A modify request may include one or more modifications to the target entry.

A modification consists of a modification type that describes the type of change (add, delete, replace, or increment), and the attribute including the attribute description and zero or more attribute value.

## D.12.11 modification type

A modification type describes one of the four ways in which an attribute can have its attribute value altered in a modification. The defined modification types are:

**add**
One or more values are to be added to the target attribute. If the attribute does not exist in the target entry, then it will be added with the given values; otherwise the provided values will be appended to the set of values already defined for that attribute. An add modification type must always supply at least one value.

**delete**
One or more values are to be removed from the target attribute, or that attribute is to be removed entirely from the target entry. If one or more specific values are given, then only those values are to be removed from the target attribute (and if they represent the entire set of values for that attribute, then that attribute will be removed from the entry). If no values are given, then the entire attribute (regardless of the number of values it contains) is to be removed from the entry.

**replace**
The set of values for the target attribute should be replaced with the given set of values. A replace can have zero or more values, and the behavior is as follows:

- If the target attribute already exists in the entry with one or more values, and the replace modification does not have any of its own values, then the target attribute will be removed from the entry.

- If the target attribute already exists in the entry with one or more values, and the replace modification has one or more of its own values, then the existing set of values will be replaced with the new set of values.
- If the target attribute does not exist in the entry and the replace modification does not have any of its own values, then no action will be taken.
- If the target attribute does not exist in the entry and the replace modification has one or more of its own values, then the attribute will be created in the entry with the specified set of values.

**increment**
The value of the target attribute should be incremented by the specified amount. The target attribute must exist in the entry with exactly one value, and that value must be an integer. The increment modification must also include exactly one value and that value must be an integer. The existing value is to be incremented by an amount specified by the increment value. If the increment value is negative, then the existing value will be deprecated by an amount equal to the absolute value of the increment value.

### D.12.12 modify DN operation

See LDAP modify DN operation.

### D.12.13 modify operation

See LDAP modify operation.

### D.12.14 monitor entry

A monitor entry is a type of entry in the server that provides information about a server component. It may provide statistical information for performance monitoring, information about the health of the server, or other information that could be of value.

The directory server provides a general-purpose monitor entry with a distinguished name of `cn=monitor`. A number of other monitor entries exist below that point, including:

- Information about each back end configured in the server
- Information about each connection handler configured in the server
- General information about the system on which the server is running
- Information about the state of the server work queue
- Version information for the server
- A stack trace of all threads currently active in the server

## D.13 N

### D.13.1 name form

A name form is a schema element that may be used to control which attribute type may be used in the relative distinguished name for an entry based on its structural object class.

A name form definition include these components:

- An object identifier used to uniquely identify the name form.
- A set of zero or more names that can be used to more easily reference the name form.
- The name or OID of the structural object class with which the name form is associated. Any entry with that structural class will be required to have an RDN which conforms to the requirements of the name form.
- An set of one or more attribute type names or OIDs for attributes that must be present in the RDN of entries with the associated structural class.
- An optional set of one or more attribute type names or OIDs for attributes that may optionally be present in the RDN of entries with the associated structural class.

The set of name forms defined in the server may be determined by retrieving the `nameForms` attribute of the subschema subentry. For more information about name forms, see the Understanding Name Forms.

## D.13.2 naming context

A naming context, also called a suffix, is a top-level entry in the server's directory information tree. It is an entry that does not have a parent.

The set of naming contexts defined in the server is listed in the `namingContexts` attribute of the root DSE. Naming contexts are visible through workflows.

## D.13.3 network group

A network group contains a set of criteria that define categories of client connection. If the client request that is sent to the server meets the policies that are attached to the network group, the network group forwards the request to a *workflow*.

## D.13.4 non-leaf entry

A non-leaf entry is an entry that has at least one subordinate entry in the server.

## D.13.5 normalized value

A normalized value is a value that has been processed in a way that makes it possible to be efficiently compared against other values. The normalization process is performed using matching rule and varies based on the type of matching rule. Some kinds of transformations that may be made include:

- Converting all characters to lowercase (or uppercase) to eliminate insignificant differences in capitalization
- Eliminating unnecessary spaces in the value
- Converting values which may have multiple representations into a common form

## D.13.6 notice of disconnection unsolicited notification

The notice of disconnection is a type of unsolicited notification that can be used to indicate that the server is about to close the connection to the client for some reason (for example, the server is being shut down, or the client has remained idle for too long).

The OID for the extended response containing the notice of disconnection is `1.3.6.1.4.1.1466.20036`. It will not have a response value, but the result code may provide an indication of the reason for the disconnection, and the diagnostic message may provide a human-readable explanation.

### D.13.7 NOT search filter

A NOT search filter is a type of LDAP search filter that is intended to serve as a container that holds exactly one embedded search filter. The NOT filter is essentially an inverse operation, and in order for an entry to match a NOT filter, it must not match the embedded filter.

NOT filters may be represented as a string by enclosing the entire filter in parentheses and placing an exclamation point just after the opening parentheses. For example, a filter of `(!(objectClass=person))` will only match an entry if it does not have an object class value of `person`.

### D.13.8 numeric algorithm

A proxy distribution algorithm in which data is split into partitions based on numerical delimitations. For example, [1-1000[ for one partition, and [1000-2000[ for the next partition.

### D.13.9 nsuniqueid

A unique identifier that is assigned to each entry in the directory server to resolve naming conflicts while migrating legacy applications using Oracle Directory Server Enterprise Edition as an LDAP database to Oracle Unified Directory.

## D.14 O

### D.14.1 object class

An object class is a schema element that correlates an object identifier and a set of names with a set of required and optional attribute type.

The components of an object class definition include:

- An OID used to uniquely identify the object class.
- A set of zero or more names that can be used to more easily reference the object class.
- An optional superior class, which may define additional required and/or optional attribute types.
- An optional object class type value that indicate whether the object class is structural object class, auxiliary object class, or abstract object class.
- An optional set of one or more attribute type names or OIDs for attributes that must be present in entries containing the object class.
- An optional set of one or more attribute type names or OIDs for attributes that may optionally be present in entries containing the object class.

Every entry must have exactly one structural object class, and it may have zero or more auxiliary classes. The complete set of object classes in an entry define the set of attribute types that are required or allowed to be present. The structural class may also

be used to link the entry with a name form, DIT content rule, and/or DIT structure rule.

The set of object classes defined in the server may be determined by retrieving the `objectClasses` attribute of the subschema subentry. For more information about object classes, see the Understanding Object Classes document.

## D.14.2 object class type

An object class type is used to define the category for an object class. There are three object class type values:

**structural object class**
A structural object class is used to define the primary type for an entry. Each entry must have exactly one structural class, and it defines the core type of object that the entry represents.

**auxiliary object class**
An auxiliary object class is used to define a characteristic of an entry. An entry may have zero or more auxiliary classes. The set of auxiliary classes that an entry may have may be controlled by a DIT content rule that is associated with the entry's structural class.

**abstract object class**
An abstract object class is not intended to be used directly in entries but should be subclassed by a structural or auxiliary class.

The inheritance model used for LDAP object classes is very similar to the inheritance model for Java classes. Just like an entry must only exactly one structural object class, a Java class must have exactly one superclass. Similarly, while an entry may have multiple auxiliary classes, a Java class may implement multiple interfaces. Finally, it is not possible to instantiate an abstract Java class, just as it is not possible to create an entry containing only an abstract object class.

## D.14.3 object identifier

An object identifier (OID) is a string that comprises a series of integers separated by periods. It is used as a unique identifier for various types of elements in the Directory Server, including:

- attribute syntax
- matching rule
- attribute type
- object class
- name form
- control
- extended operation
- supported feature

## D.14.4 operation ID

An operation ID is an integer identifier that is assigned to each operation performed on a client connection. It is used primarily for logging purposes, so that it is possible to correlate a response log message with the corresponding request message.

The first operation performed on a client connection is assigned an operation ID of zero, and it is incremented by one for each additional request received on that client connection.

### D.14.5 operational attribute

A user attribute is an attribute type with an attribute usage of `directoryOperation`, `distributedOperation`, or `dSAOperation`. Operational attributes are used for storing information needed for processing by the server itself or for holding any other data maintained by the server that was not explicitly provided by clients.

Operational attributes are not included in entries returned from search operations unless they are explicitly included in the list of search attributes. An explicit value of + (the plus sign) may also be included to request that all operational attributes be returned.

### D.14.6 ordering index

An ordering index is a type of index that is used to keep track of the relative order of values for an attribute. It is very similar to an equality index except that it uses an ordering matching rule instead of an equality matching rule to normalized value the values. Ordering indexes may not be maintained for attributes that do not have a corresponding ordering matching rule.

### D.14.7 OR search filter

An OR search filter is a type of LDAP search filter that is intended to serve as a container that holds zero or more other search filters. In order for an entry to match an OR filter, it must match at least one of the filters contained in that OR filter.

OR filters may be represented as a string by enclosing the entire filter in parentheses and placing a pipe symbol (`|`) just after the opening parenthesis. For example, a filter of `(|(uid=john.doe)(uid=jane.doe))` represents an OR search filter that embeds the `(uid=john.doe)` and `(uid=jane.doe)` equality filters.

An OR filter that does not contain any embedded filters is called an LDAP false filter. The string representation for an LDAP false filter is `(|)`, and LDAP false filters will never match any target entry.

## D.15 P

### D.15.1 partition

In a proxy distribution deployment, the data is split into smaller chunks of data, each of which is known as a partition. A partition of data is typically stored on a separate remote LDAP server, or on a set of replicated remote LDAP servers to ensure high availability.

### D.15.2 password

A password is a secret value that may be used to provide proof of identity in some authentication mechanisms. In particular, a password is used in simple authentication, as well as the CRAM-MD5 SASL mechanism, DIGEST-MD5 SASL mechanism, and PLAIN SASL mechanism Simple Authentication and Security Layer mechanisms.

The security that a password provides is based entirely on the fact that only the password's owner knows what the password is. If someone else learns a user's password through some means, then that third party can impersonate that user and may be able to perform any operation available to that user.

The Directory Server provides a number of password policy features that can be used to help ensure that passwords are not discovered by third-party individuals (for example, helping to ensure that users aren't allowed to use weak passwords, providing protection against brute-force attacks, requiring authentication attempts and password changes from being performed in a secure manner), but nevertheless passwords are often considered weaker forms of protection than other kinds of identification like certificate.

## D.15.3 password expiration

Password expiration is an element of the Directory Server password policy that can be used to limit the length of time that a user can continue to use the same password. If password expiration is enabled, once a user changes his or her password, they can use it for a length of time specified as the maximum password age. As the password expiration time draws near, the user may receive warning messages in the form of control in the bind response. Once the password has expired, the user will no longer be allowed to authentication.

Once the user's password has expired, it may be necessary for an administrator to password reset before the account may be used. Alternately, if the password policy is configured appropriately, the user may also be able to change their own expired password using the Password Modify extended operation.

## D.15.4 password generator

A password generator is a piece of logic that may be used to generate a password for a user as part of a Password Modify extended operation. It will be used if the password modify request does not include a new password.

## D.15.5 Password Modify extended operation

The Password Modify extended operation is a type of extended operation that may be used to change or password reset user password. It is defined in RFC 3062 (http://www.ietf.org/rfc/rfc3062.txt) and both the request and response operations have an OID of 1.3.6.1.4.1.4203.1.11.1.

The value for the password modify request is:

```
PasswdModifyRequestValue ::= SEQUENCE {
     userIdentity    [0]  OCTET STRING OPTIONAL
     oldPasswd       [1]  OCTET STRING OPTIONAL
     newPasswd       [2]  OCTET STRING OPTIONAL }
```

The value for the password modify response is:

```
PasswdModifyResponseValue ::= SEQUENCE {
     genPasswd       [0]     OCTET STRING OPTIONAL }
```

## D.15.6 password policy

The Directory Server password policy provides a mechanism for controlling how passwords will be stored and maintained in the server, and how users will be allowed to authenticate.

Elements of the password policy include:

- The attribute used to store user passwords. By default, this is the `userPassword` attribute.
- The default set of password storage scheme that will be used to encode passwords stored in the server.
- A set of deprecated password storage scheme that can be used to authenticate users but cause the password to be re-encoded using the default schemes upon a successful bind.
- A flag that indicates whether users will be allowed to change their own passwords.
- A number of settings related to password expiration, including the maximum age for passwords, warnings before expiration, and whether users will be allowed to change their passwords after they expire.
- A number of settings related to account lockout, which can be used to prevent users from authenticating after too many failed attempts.
- Flags that indicate whether users will be required to change their passwords the first time they authenticate and/or whether they will be required to change their passwords after they have been reset by an administrator.
- A set of password validator that can be used to determine whether proposed new password values are acceptable for use.
- A flag that indicates whether users will be required to provide their current passwords to be allowed to change their passwords.
- A flag that indicates whether clients will be allowed to specify new passwords that have already been encoded using one of the password storage schemes defined in the server. Allowing pre-encoded passwords may be necessary for some applications, but may allow the user to bypass certain restrictions, like password validators, that might otherwise be enforced.
- Settings related to maintaining the last login time, including the attribute to use to store its value, the format to use for the time stamp, and whether to lock an account after too much time has elapsed without authenticating.
- Flags that control whether the user will be required to authenticate in a secure manner and/or whether they will be required to change their passwords in a secure manner.

## D.15.7 password policy control

The password policy request control is a type of LDAP control that can be used to request information about the current password policy state for a user entry. It is defined in draft-sisbehera-ldap-password-policy (`https://opends.dev.java.net/public/standards/draft-behera-ldap-password-policy.txt`). Both the request and response controls have an OID of 1.3.6.1.4.1.42.2.27.8.5.1. The request control does not have a value. The response control value is encoded as follows:

```
PasswordPolicyResponseValue ::= SEQUENCE {
     warning [0] CHOICE {
          timeBeforeExpiration [0] INTEGER (0 .. maxInt),
          graceAuthNsRemaining [1] INTEGER (0 .. maxInt) } OPTIONAL,
     error   [1] ENUMERATED {
          passwordExpired             (0),
```

```
accountLocked               (1),
changeAfterReset            (2),
passwordModNotAllowed       (3),
mustSupplyOldPassword       (4),
insufficientPasswordQuality (5),
passwordTooShort            (6),
passwordTooYoung            (7),
passwordInHistory           (8) } OPTIONAL }
```

For an example of using this control in a search request, see Searching Using the Password Policy Control.

## D.15.8 password reset

A password reset is the act of a server administrator changing a user's password. A password reset is a password change that is performed by any user other than the one that owns the account.

## D.15.9 password storage scheme

A password storage scheme provides a mechanism for encoding user passwords for storage in the server. In most cases, the password is encoded in a manner that prevents users from determining what the clear-text password is, while still allowing the server to determine whether the user-supplied password is correct. Password storage schemes currently available for use include:

#### 3DES

The password will be encoded using triple DES. Triple DES is a variation of the Data Encryption Standard (DES) that is three times slower than its predecessor but provides stronger reliability. The algorithm uses three 64-bit keys for a combined key length of 192 bits. The data is encrypted with the first key, decrypted with the second key, and then re-encrypted with the third key. You must ensure that all three keys, the first and the second key, or the second and the third keys are not identical.

#### AES

The Advanced Encryption Standard uses a symmetric block cipher that processes data blocks of 128 bits, using cipher keys with lengths of 128 (AES-128), 192 (AES-192), and 256 (AES-256) bits and is based on the Rijndael algorithm

#### BASE64

The password will be base64 encoding, which provides a very weak form of protection and should only be used for cases in which clients require this storage scheme.

#### BlowFish

The password will be encoded using the BlowFish Algorithm with a 128 bits key length.

#### CLEAR

The password will be stored in clear-text. It will not provide any protection at all, so this should only be used for cases in which clients require this storage scheme.

#### CRYPT

The password will be encoded using the UNIX crypt algorithm. This is a one-way algorithm, but it is considered weak by current standards and should generally only be used for clients which require this storage scheme.

**MD5**
The password will be encoded using an unsalted version of the MD5 message digest algorithm. This is relatively secure, although a salt hash is preferred, and one of the Secure Hash Algorithm variants are considered stronger than MD5.

**RC4**
The password will be encoded using RC4, a stream cipher using a variable key-size stream cipher with byte-oriented operations. The algorithm is based on the use of a random permutation.

**SMD5**
The password will be encoded using a salted version of the MD5 message digest algorithm.

**SHA**
The password will be encoded using an unsalted version of the SHA-1 Secure Hash Algorithm. The salted variant of this algorithm is preferred.

**SSHA**
The password will be encoded using a salted version of the SHA-1 Secure Hash Algorithm. This is the default password storage scheme used by the directory server

**SSHA256**
The password will be encoded using a salted 256-bit version of the SHA-2 Secure Hash Algorithm.

**SSHA384**
The password will be encoded using a salted 384-bit version of the SHA-2 Secure Hash Algorithm.

**SSHA512**
The password will be encoded using a salted 512-bit version of the SHA-2 Secure Hash Algorithm.

Note that the directory server also supports the use of the authentication password syntax.

## D.15.10 password validator

A password validator is a component of the directory server password policy that is used to determine whether a proposed password is acceptable for use. The directory server provides an extensible API for developing custom password validators, but it does come with a number of different types of password validators, including:

- A validator that can be used to reject a password if the value exists in any of the attribute contained in the user's entry.
- A validator that can be used to reject a password if the value does not contain characters from an acceptable range of character sets.
- A validator that can be used to reject a password if it is a word that can be found in a dictionary.
- A validator that can be used to reject a password if it is too long or too short.
- A validator that can be used to reject a password if it contains a string of too many repeated characters.
- A validator that can be used to reject a password if it is too similar to the user's current password.

- A validator that can be used to reject a password if it does not contain enough unique characters.

## D.15.11 persistent search control

The persistent search control is a type of LDAP control that may be used for clients to be notified of changes to entry that match the criteria from the associated LDAP search operation. The persistent search control is described in draft-ietf-ldapext-psearch (`https://opends.dev.java.net/public/standards/draft-ietf-ldapext-psearch.txt`)and has an OID of 2.16.840.1.113730.3.4.3. It is defined as follows:

```
PersistentSearch ::= SEQUENCE {
     changeTypes INTEGER,
     changesOnly BOOLEAN,
     returnECs BOOLEAN
}
```

search result entry returned as part of this search may optionally include the entry change notification control to describe the way in which the entry changed. For an example of using this control in a search, see Searching Using the Persistent Search Control.

## D.15.12 PLAIN SASL mechanism

The PLAIN Simple Authentication and Security Layer mechanism provides a way for clients to authentication to the Directory Server with a username and password. In general, it is very similar to simple authentication, with the exception that the client can identify itself with a username rather than a distinguished name. It also provides the ability for the client to specify an alternate authorization ID.

Like simple authentication, the PLAIN SASL mechanism does not provide any form of protection for the user password, so it may be advisable to only use this authentication method over secure communication channels like those provided by Secure Sockets Layer or StartTLS extended operation.

## D.15.13 plug-in

A plug-in is a piece of code that can be used to interject some custom logic into the way that the Directory Server performs its processing. The directory server supports a number of different types of plug-ins, including:

- Pre-parse plug-ins, which allow the server to alter the contents of a request before the server begins processing on it. Pre-parse plug-ins are available for all types of operations.
- Pre-operation plug-ins, which allow the server to take some action just before the core processing for an operation. Pre-operation plug-ins are available for all types of operations except abandon operation and unbind operation.
- Post-operation plug-ins, which allow the server to take some action just after the core processing for an operation but before the response has been sent to the client (it may be used to alter the response to the client). Post-operation plug-ins are available for all types of operations.
- Post-response plug-ins, which allow the server to take some action after all other processing for an operation has completed. Post-response plug-ins are available for all types of operations except abandon and unbind.

- Search result entry plug-ins, which alter the contents of a search result entry being sent as part of a search operation.
- Search result reference plug-ins, which alter the contents of a search result reference being sent as part of a search operation.
- Intermediate response plug-ins, which alter the contents of an LDAP intermediate response being sent to a client.
- Startup plug-ins, which perform some processing when the server is first starting.
- Shutdown plug-ins, which perform some processing when the server is performing a graceful shutdown.
- Post-connect plug-ins, which perform some processing as part of accepting a new client connection.
- Post-disconnect plug-ins, which perform some processing immediately after a connection is terminated.
- LDIF import plug-ins, which alter the contents of entry being imported from an LDAP Data Interchange Format file.
- LDIF export plug-ins, which alter the contents of entries being exported from a server back end.

### D.15.14 presence index

A presence index is a type of index that is used to keep track of the entries that have at least one value for a specified attribute. There is only a single presence index key per attribute, and its ID list contains the entry ID for all entries that contain the specified attribute.

### D.15.15 presence search filter

A presence search filter is a type of LDAP search filter that can be used to identify entries that have at least one value for a specified attribute. The string representation of an LDAP presence filter comprises an opening parenthesis followed by the attribute name, an equal sign, an asterisk, and the closing parenthesis. For example, an equality filter of `(aci=*)` will match any entry containing at least one value for the `aci` attribute.

### D.15.16 privilege

The directory server provides a privilege subsystem, which can be used to define capabilities that will be granted to users. The privilege subsystem works in conjunction with the access control implementation in the process of determining whether a user will be allowed to perform a certain operation.

Some of the privileges defined in the directory server include:

`bypass-acl`
Allows the user to bypass access control evaluation

`modify-acl`
Allows the user to modify access control rule defined in the server.

`config-read`
Allows the user to have read access to the server configuration

`config-write`
Allows the user to have write access to the server configuration

`server-shutdown`
Allows the user to request that the server shut down

`server-restart`
Allows the user to request that the server perform an in-core restart

`proxied-auth`
Allows the user to request an operation with a different authorization ID

`unindexed-search`
Allows the user to request an unindexed search

`password-reset`
Allows the user to password reset for other users

`update-schema`
Allows the user to update the server schema

See Root Users and the Privilege Subsystem for more information on the privilege subsystem.

## D.15.17 proportional algorithm

A proxy load balancing algorithm in which client requests are distributed to a set of replicated remote LDAP servers. How many requests are sent to each remote LDAP server is determined by the weight set.

## D.15.18 protocol data unit

A protocol data unit (PDU) is a single complete element of network communication. For LDAP, the PDU is the message.

## D.15.19 protocol op

The protocol op is the element in the message that contains the heart of the request or response. That is, it indicates what type of message it is. There are several different kinds of protocol op elements, including:

- The abandon operation
- The add operation
- Thebind operation
- The compare operation
- The delete operation
- The extended operation
- The modify operation
- The modify DN operation
- The search operation
- The unbind operation
- The LDAP intermediate response

### D.15.20 proxied authorization control

The proxied authorization control is a type of control that can be used to request that the associated operation be performed under the authorization of another user.

There are actually two different forms of the proxied authorization control, both of which are request controls that may be attached to an add operation, compare operation, delete operation, modify operation, modify DN operation, or search operation operation.

The proxied authorization v1 control is defined in early versions of draft-weltman-ldapv3-proxy. It has an OID of 2.16.840.1.113730.3.4.12 and the control value should be encoded as:

```
proxyAuthValue::= SEQUENCE {
      proxyDN LDAPDN
}
```

The proxied authorization v2 control is defined in RFC 4370 (`http://www.ietf.org/rfc/rfc4370.txt`). It has an OID of 2.16.840.1.113730.3.4.18 and the value is a string containing the desired authorization ID.

For an example of using this control in a search request, see Searching Using the Proxied Authorization Control.

## D.16 Q

### D.16.1 quality of protection

Quality of protection (QoP) is a property of certain Simple Authentication and Security Layer mechanisms (especially the DIGEST-MD5 SASL mechanism and GSSAPI SASL mechanism mechanisms) that can be used to protect the communication between the client and the server.

There are three different QoP levels:

`auth`
This indicates that the associated SASL mechanism should only be used to authenticate the client connection. It should not provide any other protection for the client-server communication

`auth-int`
This indicates that the associated SASL mechanism should be used for authentication, and then should also provide integrity protection for the communication between the client and server. Integrity protection will not prevent third-party observers from understanding the communication, but it will ensure that a man-in-the-middle is unable to alter that communication in an undetectable manner

`auth-conf`
This indicates that the associated SASL mechanism should be used for authentication, and then should also provide integrity and confidentiality protection for the communication between the client and the server. This will ensure that third-party observers will be unable to understand the communication

At the present time, the directory server supports only the `auth` quality of protection. It does not support either the `auth-int` or `auth-conf` levels.

# D.17 R

## D.17.1 real attributes only control

The real attributes only control is a control that may be used to request that the server only include real attributes in matching entries. That is, virtual attribute are excluded from search result entry.

The real attributes only control has a request object identifier of `2.16.840.1.113730.3.4.17` and no value.

In the following search, the `numsubordinates` virtual attribute is requested and returned:

```
$ ldapsearch -D "cn=directory manager" -j pwd-file -b
"ou=people,dc=example,dc=com" \
  -s base "objectclass=*" numsubordinates
version: 1
dn: ou=People,dc=example,dc=com
numSubordinates: 50
```

In the following search, the `numsubordinates` virtual attribute is requested but is not returned because the real attributes only control is used:

```
$ ldapsearch -D "cn=directory manager" -j pwd-file -J "2.16.840.1.113730.3.4.17" \
  -b "ou=people,dc=example,dc=com" -s base "objectclass=*" numsubordinates
version: 1
dn: ou=People,dc=example,dc=com
```

## D.17.2 referential integrity

Referential integrity is a mechanism for ensuring that any references to an entry are updated whenever that entry is removed or altered. Historically, referential integrity is primarily used to ensure that attributes with a distinguished name syntax (especially group membership attributes like `member` and `uniqueMember`) are properly maintained in the event of delete operation and modify DN operation operations. For a delete operation, any references to the target entry will be removed. For modify DN operations, any references to the target entry will be renamed accordingly.

The directory server provides a configurable referential integrity plug-in that you can install using the dsconfig command.

## D.17.3 referral

A referral provides a reference to an alternate location in which an operation may be processed. A referral may be included in an LDAP result object with a result code of `10` and an appropriate set of LDAP URL. It may also be returned to clients in a search result reference.

## D.17.4 relative distinguished name

A relative distinguished name, or RDN, is a single component within a distinguished name. It comprises one or more name-value pairs, in which the name and the value are separated by an equal sign (for example, for an RDN of `uid=ann`, the name is `uid` and the value is `ann`), and if there are multiple name-value pairs then they should be

separated by plus signs (for example, for an RDN of `cn=Jon Doe+employeeNumber=12345`, the name-value pairs are `cn=John Doe` and `employeeNumber=12345`). In practice, RDNs containing multiple name-value pairs (called "multivalued RDNs") are rare, but they can be useful at times when either there is no unique attribute in the entry or you want to ensure that the entry's DN contains some useful identifying information.

Even though a DN may be composed of multiple RDN components, the leftmost component is typically referred to as the entry's RDN. For example, in a DN of `uid=john.doe,ou=People,dc=example,dc=com`, the RDN would be `uid=john.doe`. The attribute values specified in an entry's RDN must be contained in that entry, so the entry `uid=john.doe,ou=People,dc=example,dc=com` must have a `uid` value of `john.doe`.

## D.17.5 replica

A replica is a Directory Server instance that participates in replication.

## D.17.6 replication

Replication is a form of data synchronization that is used to ensure that changes in the directory environment are reflected in each instance of the server. That is, whenever a change is made in one Directory Server instance, that same change is also made in every other instance.

## D.17.7 replication repair control

The replication repair control is a control that can be used to resolve replication inconsistencies on a single server in a topology.

The replication repair control has a request object identifier of `1.3.6.1.4.1.26027.1.5.2` and no value.

For an example of using the replication repair control, see Detecting and Resolving Replication Inconsistencies.

## D.17.8 request for comments

A request for comments (RFC) is an IETF (`http://www.ietf.org/`) specification that has been promoted from an Internet Draft and may be considered significantly more stable than drafts.

## D.17.9 restore

A restore operation provides a mechanism for replacing the contents of a Directory Server back end with information taken from a previous backup. It can serve as a disaster recovery mechanism, and in some cases can be used for binary copy initialization of a replica.

## D.17.10 result

See LDAP result.

## D.17.11 result code

A result code is an integer value that provides general information about the result of the operation. Defined result codes include:

| Value | Name | Description |
|---|---|---|
| 0 | Success | This is used to indicate that the associated operation completed successfully. |
| 1 | Operations Error | This is used to indicate that the associated request was out of sequence with another operation in progress (for example, a non-bind request in the middle of a multi-stage SASL bind). |
| 2 | Protocol Error | This is used to indicate that the client sent data to the server that did not comprise a valid LDAP request. |
| 3 | Time Limit Exceeded | This is used to indicate that processing on the associated request was terminated because it took too long to complete. For a search operation, it is possible that some of the matching entries had been returned when the time limit was reached. |
| 4 | Size Limit Exceeded | This is used to indicate that there were more entries matching the criteria contained in a search operation than were allowed to be returned by the size limit configuration. |
| 5 | Compare False | This is used to indicate that a compare operation completed successfully, but the provided attribute value assertion did not match the target entry. |
| 6 | Compare True | This is used to indicate that a compare operation completed successfully, and the provided attribute value assertion matched the target entry. |
| 7 | Auth Method Not Supported | This is used to indicate that the Directory Server does not support the requested authentication method. |
| 8 | Strong Auth Required | This is used to indicate that the Directory Server requires that the client use a strong authentication mechanism. |
| 10 | Referral | This is used to indicate that the requested operation could not be processed in the target server but may be attempted in elsewhere. |
| 11 | Admin Limit Exceeded | This is used to indicate that processing on the requested operation could not be completed because an administrative limit was reached. For a search operation, it is possible that some of the matching entries had been returned when the administrative limit was reached. |
| 12 | Unavailable Critical Extension | This is used to indicate that the request included a critical control that could not be processed by the server. |
| 13 | Confidentiality Required | This is used to indicate that the requested operation requires a secure communication channel between the client and the server. |
| 14 | SASL Bind In Progress | This is used to indicate that a SASL bind operation requires multiple stages and the response containing this result code is one of the intermediate stages. |
| 16 | No Such Attribute | This is used to indicate that the associated request targeted an attribute or attribute value that does not exist in the specified entry. |
| 17 | Undefined Attribute Type | This is used to indicate that the associated request included an attribute type that is not defined in the server schema. |

| Value | Name | Description |
|---|---|---|
| 18 | Inappropriate Matching | This is used to indicate that the associated search request included a filter with a component targeting an attribute type for which no appropriate matching rule is defined. |
| 19 | Constraint Violation | This is used to indicate that the requested operation could not be completed because it would have violated some constraint defined in the server (for example, it would have duplicated a value for a unique attribute). |
| 20 | Attribute or Value Exists | This is used to indicate that an operation attempted to create an attribute value in an entry that already existed in the entry, or that it attempted to create an additional value for a single-valued attribute. |
| 21 | Invalid Attribute Syntax | This is used to indicate that requested operation attempted to specify a value that violated the syntax for the associated attribute type. |
| 32 | No Such Object | This is used to indicate that the requested operation targeted an entry that does not exist in the server. |
| 33 | Alias Problem | This is used to indicate that an operation targeted an alias entry and that operation is not allowed on alias entries. |
| 34 | Invalid DN Syntax | This is used to indicate that the requested operation included an entry DN that was malformed. |
| 35 | Is Leaf | This is used to indicate that the requested operation targeted a leaf entry but the operation requires a non-leaf entry. |
| 36 | Alias Dereferencing Problem | This is used to indicate that the associated search operation encountered an alias that could not be properly dereferenced. |
| 48 | Inappropriate Authentication | This is used to indicate that the client attempted to bind in a manner that is inappropriate for the target user (for example, the user attempted simple authentication but does not have a password). |
| 49 | Invalid Credentials | This is used to indicate that the client attempted to authenticate with invalid credentials (for example, the target DN or password was incorrect). |
| 50 | Insufficient Access Rights | This is used to indicate that the client was not allowed to perform the requested operation. |
| 51 | Busy | This is used to indicate that the server is too busy to process the requested operation. |
| 52 | Unavailable | This is used to indicate that the server is unavailable for processing operations. |
| 53 | Unwilling to Perform | This is used to indicate that the server is unwilling to perform the requested operation for some reason. |
| 54 | Loop Detect | This is used to indicate that the server encountered a loop of some type (for example, a chaining loop or an alias loop). |
| 60 | Sort Control Missing | This is used to indicate that the client requested a search operation containing the virtual list view control that did not also include the server-side sort control. |

| Value | Name | Description |
|---|---|---|
| 61 | Offset Range Error | This is used to indicate that the request included a virtual list view control that specified an invalid offset (for example, one that was beyond the end of the result set). |
| 64 | Naming Violation | This is used to indicate that the operation attempted to create an entry with a DN that violated a naming constraint (for example, using an RDN attribute that is not allowed by the associated name form). |
| 65 | Object Class Violation | This is used to indicate that the operation attempted to create or modify an entry so that the set of attributes it contained were in violation of the associated object class definitions (for example, it included an attribute that was not allowed or was missing a required attribute). |
| 66 | Not Allowed On Nonleaf | This is used to indicate that the associated operation was not allowed on non-leaf entries (for example, an attempt to delete an entry that has one or more subordinate entries). |
| 67 | Not Allowed On RDN | This is used to indicate that the associated operation is not allowed on the RDN attribute for an entry. |
| 68 | Entry Already Exists | This is used to indicate that the add or modify DN operation would have resulted in an entry with a DN that already exists in the server. |
| 69 | Object Class Mods Prohibited | This is used to indicate that the requested operation attempted to alter the structural object class for the entry in a manner that was not allowed. |
| 71 | Affects Multiple DSAs | This is used to indicate that the requested operation would have impacted multiple servers (for example, a modify DN operation would have moved an entry from one server to another through a chained back end). |
| 76 | Virtual List View Error | This is used to indicate that the associated search operation could not be completed successfully because a problem occurred while processing the virtual list view request. |
| 80 | Other | This indicates that the operation failed for some reason that is not more appropriately classified by any other defined result code. |
| 81 | Server Down | This is a client-side result code that is used to indicate that the client detected that an established connection was no longer available. |
| 82 | Local Error | This is a client-side result code that is used to indicate that some client-side problem occurred that prevented it from completing the associated processing successfully. |
| 83 | Encoding Error | This is a client-side result code that is used to indicate that an error occurred while attempting to encode the request to send to the server. |
| 84 | Decoding Error | This is a client-side result code that is used to indicate that an error occurred while attempting to decode the response received from the server. |
| 85 | Timeout | This is a client-side result code that is used to indicate that the client did not receive a response in an acceptable length of time. |

| Value | Name | Description |
|---|---|---|
| 86 | Authentication Type Unknown | This is a client-side result code that is used to indicate that the client does not support the requested authentication method. |
| 87 | Filter Error | This is a client-side result code that is used to indicate that a provided filter string could not be parsed as a valid filter. |
| 88 | User Canceled | This is a client-side result code that is used to indicate that the client canceled the request. |
| 89 | Parameter Error | This is a client-side result code that is used to indicate that there was a problem with a parameter provided for a request element. |
| 90 | No Memory | This is a client-side result code that is used to indicate that the client ran out of memory while attempting to process the requested operation (for example, while queueing the search result entries). |
| 91 | Connect Error | This is a client-side result code that is used to indicate that the client could not establish a connection to the target server. |
| 92 | Not Supported | This is a client-side result code that is used to indicate that the requested operation is not supported by the client. |
| 93 | Control Not Found | This is a client-side result code that is used to indicate that a response did not include an expected control. |
| 94 | No Results Returned | This is a client-side result code that is used to indicate that the server did not return any results for a search request when at least one was expected. |
| 95 | More Results to Return | This is a client-side result code that is used to indicate that there are more results to return than those that have already been retrieved. |
| 96 | Client Loop | This is a client-side result code that is used to indicate that the client detected a referral loop. |
| 97 | Referral Limit Exceeded | This is a client-side result code that is used to indicate that the client received too many referrals in the course of processing a request. |
| 100 | Invalid Response | This is a client-side result code that is used to indicate that the result received for the associated operation is invalid. |
| 101 | Ambiguous Response | This is a client-side result code that is used to indicate that the result received from the server was ambiguous (for example, there was more than one response received fro the associated operation). |
| 112 | TLS Not Supported | This is used to indicate that the server does not support the StartTLS extended operation. |
| 113 | Intermediate Response | This result code is used for intermediate response messages sent by the server in the course of processing the request. |
| 114 | Unknown Type | This is used to indicate that the server received a request with an invalid or unknown protocol op type. |
| 118 | Canceled | This is used to indicate that the server canceled processing on the request at the request of the client. |

| Value | Name | Description |
|---|---|---|
| 119 | No Such Operation | This is used to indicate that the client attempted to cancel a request that was unknown to the server (for example, because it had already completed processing). |
| 120 | Too Late | This is used to indicate that the client attempted to cancel a request that had already been processed beyond a point at which it could no longer be canceled. |
| 121 | Cannot Cancel | This is used to indicate that the client attempted to cancel an operation that could not be canceled (for example, a bind, unbind, abandon, cancel, or StartTLS request). |
| 122 | Assertion Failed | This is used to indicate that the associated operation was not processed because the request included an LDAP assertion control with an assertion filter that did not match the target entry. |
| 123 | Authorization Denied | This is used to indicate that the associated operation was not processed because the request included a proxied authorization control but the client was not allowed to use that control. |

## D.17.12 root DN

A root DN (or root user) is a type of account that exists in the Directory Server which is generally given full access to all data in the server, much like the root user in UNIX systems. Root users by default will be allowed to bypass access control evaluation, will have full access to the server configuration, and perform most other types of operations.

The directory server is different from most other servers with regard to root users in two key ways:

- The directory server can be configured with multiple root users. This is a good thing because it allows each root user to have a different set of credentials so that each administrator can have a separate root account that is independent from the others rather than a single account that is shared by all administrators.
- All of the rights given to root users are assigned through privilege. Using the privilege subsystem, it is possible to create non-root users with some or all of the capabilities normally available only to root users. It is also possible to take away privileges from root users if so desired.

For more information on root users and the privilege subsystem, see the Root Users and the Privilege Subsystem document.

## D.17.13 root DSE

The root DSE is a special entry that provides information about the contents and capabilities of the server. The distinguished name is a zero-length string with no relative distinguished name components, also called the null DN.

The attribute contained in the root DSE include:

`namingContexts`
Lists the naming context for the server

`supportedAuthPasswordSchemes`
Lists the object identifier of the supported password storage scheme using the authentication password syntax

`supportedControl`
Lists the OIDs of the supported control in the server

`supportedExtension`
Lists the OIDs of the supported extension in the server

`supportedFeatures`
Lists the OIDs of the supported feature in the server

`supportedSASLMechanisms`
Lists the OIDs of the supported Simple Authentication and Security Layer mechanisms in the server

`vendorName`
Provides the name of the vendor for the server

`vendorVersion`
Provides a product version string

The following example demonstrates how to use the `ldapsearch` command to read the root DSE. In this example the file `/tmp/pwd.txt` contains the Directory Manager password. The server is listening for LDAP requests on port 1389.

```
$ ldapsearch -D "cn=Directory Manager" -j /tmp/pwd.txt -p 1389 -b "" \
  -s base "(objectclass=*)" +
dn:
supportedLDAPVersion: 2
supportedLDAPVersion: 3
supportedExtension: 1.3.6.1.4.1.4203.1.11.3
supportedExtension: 1.3.6.1.4.1.4203.1.11.1
supportedExtension: 1.3.6.1.4.1.26027.1.6.1
supportedExtension: 1.3.6.1.4.1.26027.1.6.3
supportedExtension: 1.3.6.1.4.1.26027.1.6.2
supportedExtension: 1.3.6.1.1.8
supportedExtension: 1.3.6.1.4.1.1466.20037
vendorName: Oracle Corporation
entryDN:
ds-private-naming-contexts: cn=admin data
ds-private-naming-contexts: cn=ads-truststore
ds-private-naming-contexts: cn=backups
ds-private-naming-contexts: cn=config
ds-private-naming-contexts: cn=monitor
ds-private-naming-contexts: cn=schema
ds-private-naming-contexts: cn=tasks
supportedControl: 1.2.826.0.1.3344810.2.3
supportedControl: 1.2.840.113556.1.4.319
supportedControl: 1.2.840.113556.1.4.473
supportedControl: 1.2.840.113556.1.4.805
supportedControl: 1.3.6.1.1.12
supportedControl: 1.3.6.1.1.13.1
supportedControl: 1.3.6.1.1.13.2
supportedControl: 1.3.6.1.4.1.26027.1.5.2
supportedControl: 1.3.6.1.4.1.42.2.27.8.5.1
supportedControl: 1.3.6.1.4.1.42.2.27.9.5.2
supportedControl: 1.3.6.1.4.1.42.2.27.9.5.8
supportedControl: 1.3.6.1.4.1.4203.1.10.2
supportedControl: 1.3.6.1.4.1.7628.5.101.1
supportedControl: 2.16.840.1.113730.3.4.12
supportedControl: 2.16.840.1.113730.3.4.16
supportedControl: 2.16.840.1.113730.3.4.17
```

```
supportedControl: 2.16.840.1.113730.3.4.18
supportedControl: 2.16.840.1.113730.3.4.19
supportedControl: 2.16.840.1.113730.3.4.2
supportedControl: 2.16.840.1.113730.3.4.3
supportedControl: 2.16.840.1.113730.3.4.9
supportedSASLMechanisms: PLAIN
supportedSASLMechanisms: EXTERNAL
supportedSASLMechanisms: CRAM-MD5
supportedSASLMechanisms: DIGEST-MD5
supportedFeatures: 1.3.6.1.1.14
supportedFeatures: 1.3.6.1.4.1.4203.1.5.1
supportedFeatures: 1.3.6.1.4.1.4203.1.5.2
supportedFeatures: 1.3.6.1.4.1.4203.1.5.3
subschemaSubentry: cn=schema
hasSubordinates: true
entryUUID: d41d8cd9-8f00-3204-a980-0998ecf8427e
numSubordinates: 1
namingContexts: dc=example,dc=com
vendorVersion: Oracle Unified Directory 11.1.1.5.0
supportedAuthPasswordSchemes: MD5
supportedAuthPasswordSchemes: SHA1
supportedAuthPasswordSchemes: SHA256
supportedAuthPasswordSchemes: SHA384
supportedAuthPasswordSchemes: SHA512
```

For more information on how to search the root DSE entry, see Using Advanced Search Features.

### D.17.14 route

In proxy mode, the path on which requests are sent to the remote LDAP server when using a load balancing algorithm.

## D.18 S

### D.18.1 salt

A salt is a collection of random data that may be combined with clear-text data (often a password) that can be used to change the way that it is encoded. In particular, the salt is used to introduce randomness into the encoding process to help thwart dictionary attacks. In general, the salt is appended to the clear-text password, which is the encoded using the desired message digest algorithm, and then the clear-text salt is appended to the message digest and the resulting value is base64 encoding. This makes it possible to determine what the salt was so that it can be used to determine whether a user-supplied password is correct.

The UNIX crypt algorithm uses a relatively weak 12-bit salt, which means that there are only 4096 ways of encoding any value. This is a relatively low number, and therefore it is possible to construct dictionaries of every possible encoding for a wide range of values for use in breaking user passwords. Other password storage scheme in the directory server use a 64-bit salt which provide 18446744073709551616 different ways of encoding any one value.

## D.18.2 saturation algorithm

A proxy load balancing algorithm in which client requests are routed to a priority remote LDAP server. When the main remote LDAP server reaches its *saturation threshold*, the requests are routed to a secondary remote LDAP server.

## D.18.3 saturation alert

The limit at which a notification is sent to the administrator to indicate that the remote LDAP server is overloaded. Usually, the saturation alert is set higher than the *saturation threshold*.

## D.18.4 saturation threshold

The saturation threshold is the limit at which the data source is considered overloaded and can no longer handle incoming requests in an optimal way. The saturation threshold is used as part of the proxy saturation algorithm.

## D.18.5 schema

The schema of a Directory Server defines a set of rules that govern the kinds of information that the server can hold. Directory schema includes a number of different elements, including:

**attribute syntax**
Provide information about the kind of information that can be stored in an attribute.

**matching rule**
Provide information about how to make comparisons against attribute values.

**matching rule use**
Indicate which attribute types may be used in conjunction with a particular matching rule.

**attribute type**
Define an object identifier and a set of names that may be used to refer to a given attribute, and associates that attribute with a syntax and set of matching rules.

**object class**
Define named collections of attributes and classify them into sets of required and optional attributes.

**name form**
Define rules for the set of attributes that should be included in the relative distinguished name for an entry.

**DIT content rule**
Define additional constraints about the object classes and attributes that may be used in conjunction with an entry.

**DIT structure rule**
Define rules that govern the kinds of subordinate entries that a given entry may have.

attribute are the elements responsible for storing information in a directory, and the schema defines the rules for which attributes may be used in an entry, the kinds of values that those attributes may have, and how clients may interact with those values.

Clients may learn about the schema elements that the server supports by retrieving an appropriate subschema subentry.

## D.18.6 schema checking

Schema checking is the process of ensuring that an entry conforms to the constraints defined by the server schema. This includes:

- Make sure the entry contains exactly one structural object class.
- If there is a name form for the entry's structural class, ensure that the relative distinguished name attributes conform with that name form.
- If there is a DIT content rule for the entry's structural class, make sure that all of the auxiliary object class are defined.
- Make sure that all of the object class contained in the entry are defined in the schema.
- Make sure that all of the attribute contained in the entry are defined in the schema and allowed by the object classes and/or DIT content rule.
- Make sure that all attributes required by the entry's object classes and/or DIT content rule are present.
- Make sure that all single-valued attributes contained in the entry only have one value.
- Make sure that the entry's position in the directory information tree conforms with DIT structure rule definitions.

## D.18.7 search attributes

The search attributes element of a search operation provides a way of representing the attribute that should be included in search result entry. In general, the set of search attributes is a list of zero or more attribute description for the attributes to return. If values are specified, then all user attribute and no operational attribute will be returned.

In addition to specific attribute descriptions, a number of special values can be provided with various meanings:

- The string `1.1` indicates that no attributes should be included in matching entries.
- The string `*` (the asterisk) indicates that all user attributes should be included in matching entries. This is needed if the server returns all user attributes in addition to one or more operational attributes.
- The string `+` (the plus sign) indicates that all operational attributes should be included in matching entries.
- An object class name can be provided, prefixed with the `@` character. This indicates that all attributes referenced by that object class should be included in matching entries.

## D.18.8 search base DN

The search base DN is an element of the search operation that works in conjunction with the search scope to define the subtree of entries that should be considered when processing the search operation. Only entries at or below the search base DN and within the scope will be considered candidates for matching against the LDAP search filter.

## D.18.9 search filter

See LDAP search filter.

## D.18.10 search operation

The LDAP search operation can be used to identify entries in the Directory Server that match a given set of criteria. It may return zero or more entries, and also zero or more referrals.

The search request protocol op is defined as follows:

```
SearchRequest ::= [APPLICATION 3] SEQUENCE {
      baseObject      LDAPDN,
      scope           ENUMERATED {
          baseObject              (0),
          singleLevel             (1),
          wholeSubtree            (2),
          ...  },
     derefAliases    ENUMERATED {
          neverDerefAliases       (0),
          derefInSearching        (1),
          derefFindingBaseObj     (2),
          derefAlways             (3) },
     sizeLimit       INTEGER (0 ..  maxInt),
     timeLimit       INTEGER (0 ..  maxInt),
     typesOnly       BOOLEAN,
     filter          Filter,
     attributes      AttributeSelection }
```

The elements of the search request include:

- The search base DN, which specifies the location in the directory information tree in which to perform the search.
- The search scope, which specifies the scope of entries at or below the base DN to consider when processing the search.
- The dereference policy to use if any aliases are encountered during processing.
- The size limit, which specifies the maximum number of entries that should be returned from the search (or zero if there should not be any maximum number of entries).
- The time limit, which specifies the maximum length of time in seconds that the server should spend processing the search (or zero if there should not be a maximum number of entries).
- The typesOnly flag, which indicates whether the entries returned should include attribute types only or both types and values.
- The LDAP search filter, which specifies the criteria to use to identify matching entries.
- The search attributes that indicate which attributes should be included in matching entries, or an empty list to indicate that all user attribute should be returned.

There are three types of result elements that can be returned in response to a search request: zero or more search result entry, zero or more search result reference, and exactly one search result done message. The entries and references can be returned in any order (and with search entries and references interspersed), and the search result done message will come last to indicate that there are no more results.

The search result entry protocol op is defined as follows:

```
SearchResultEntry ::= [APPLICATION 4] SEQUENCE {
     objectName      LDAPDN,
     attributes      PartialAttributeList }

PartialAttributeList ::= SEQUENCE OF
                     partialAttribute PartialAttribute
```

Each search result entry includes the DN of the entry and zero or more attributes (potentially including only the attribute type names without the values if the `typesOnly` element of the request is true) as defined in the search attribute list.

The search result reference protocol op is defined as follows:

```
SearchResultReference ::= [APPLICATION 19] SEQUENCE
                          SIZE (1..MAX) OF uri URI
```

Each search result reference includes one or more LDAP URL specifying an alternate location in which the client may search for additional matching entries.

The search result done message is an LDAP result defined as follows:

```
SearchResultDone ::= [APPLICATION 5] LDAPResult
```

## D.18.11 search result done

A search result done message is a message provided as part of a search operation to indicate that the search has completed and that there will be no more search result entry or search result reference messages.

## D.18.12 search result entry

A search result entry is an entry returned as part of a search operation. It will contain at least the distinguished name of the entry, and can contain zero or more attributes. The attributes can contain only attribute type names or both types and values (based on the value of the typesOnly flag from the search request). The attributes returned can be based on the search attributes from the client request, but can be pared down based on the server's access control configuration.

## D.18.13 search result reference

A search result reference provides a mechanism for returning information to clients as part of a search operation that indicates an alternate location in which the client may perform the search to locate additional matching entries. The alternate locations will be specified in the form of LDAP URL.

## D.18.14 search scope

The LDAP search scope indicates the set of entries at or below the search base DN that may be considered potential matches for a search operation.

There are four defined search scope values:

**baseObject**
This specifies that the search operation should only be performed against the entry specified as the search base DN. No entries below it will be considered.

Consider a scenario of DIT, which has a baseObject scope with a search base DN of `dc=example,dc=com`.

**singleLevel**
This specifies that the search operation should only be performed against entries that are immediate subordinates of the entry specified as the search base DN. The base entry itself is not included, nor are any entries below the immediate subordinates of the search base entry.

**wholeSubtree**
This specifies that the search operation should be performed against the entry specified as the search base and all of its subordinates to any depth.

**subordinateSubtree**
This specifies that the search operation should be performed against all subordinate entries below the search base to any depth, but the search base entry itself should not be included.

## D.18.15 Secure Hash Algorithm

The Secure Hash Algorithm (SHA) is a one-way message digest algorithm. There are actually two different forms of the Secure Hash Algorithm:

- SHA-1 is defined in RFC 3174 (`http://www.ietf.org/rfc/rfc3174.txt`) and generates a 160-bit digest.
- SHA-2 is defined in RFC 4634 (`http://www.ietf.org/rfc/rfc4634.txt`) and can be used to generate 256-bit, 384-bit, or 512-bit digests.

All forms of the Secure Hash Algorithm are considered stronger than the MD5 algorithm. There have been recent advancements that may indicate a weakening of the SHA-1 variant, but nevertheless there is no evidence to suggest that the way it is used in the directory server is under any danger, nor is there any concern about any of the SHA-2 encodings.

## D.18.16 Secure Sockets Layer

The Secure Sockets Layer (SSL) is a mechanism for wrapping network communication in a security layer that can be used to encrypt communication between the client and the server. It also provides an integrity mechanism to ensure that the communication is not altered between the client and the server. The encryption is based on cryptography using certificate.

SSL was originally a proprietary protocol developed by Netscape Communications. It has since been standardized, but the name has been changed to Transport Security Layer. Nevertheless, SSL is still a commonly-used term to refer to this capability, and it is the term used throughout the directory server in order to avoid confusion with the StartTLS extended operation.

## D.18.17 server-side sort control

The server-side sort control is a type of control that can be attached to a search operation to request that the results be sorted before they are returned to the client. It is defined in RFC 2891 (`http://www.ietf.org/rfc/rfc2891.txt`).

The request control has an object identifier of 1.2.840.113556.1.4.473 and the value is encoded as follows:

```
SortKeyList ::= SEQUENCE OF SEQUENCE {
```

```
     attributeType   AttributeDescription,
     orderingRule    [0] MatchingRuleId OPTIONAL,
     reverseOrder    [1] BOOLEAN DEFAULT FALSE }
```

For an example of using this control in a search request, see Searching Using the Server-Side Sort Control.

The response control has an OID of 1.2.840.113556.1.4.474 and its value is encoded as follows:

```
SortResult ::= SEQUENCE {
     sortResult  ENUMERATED {
          success                   (0), -- results are sorted
          operationsError           (1), -- server internal failure
          timeLimitExceeded         (3), -- timelimit reached before
                                         -- sorting was completed
          strongAuthRequired        (8), -- refused to return sorted
                                        -- results via insecure
                                         -- protocol
          adminLimitExceeded       (11), -- too many matching entries
                                         -- for the server to sort
          noSuchAttribute          (16), -- unrecognized attribute
                                         -- type in sort key
          inappropriateMatching    (18), -- unrecognized or
                                         -- inappropriate matching
                                         -- rule in sort key
          insufficientAccessRights (50), -- refused to return sorted
                                         -- results to this client
          busy                     (51), -- too busy to process
          unwillingToPerform       (53), -- unable to sort
          other                    (80)
          },
     attributeType [0] AttributeDescription OPTIONAL }
```

## D.18.18 simple authentication

Simple authentication is the process of authentication to the Directory Server using a distinguished name and password. This is done using an bind operation (and when the bind is performed using simple authentication, it is often called a "simple bind"). The client uses the provided DN to identify itself to the server, and the password is used to verify that the client is who it claims to be.

Note that simple authentication does not protect the password in any way, and therefore it is generally recommended that it only be used over a secure communication channel like that provided by Secure Sockets Layer or StartTLS extended operation.

## D.18.19 Simple Authentication and Security Layer

The Simple Authentication and Security Layer (SASL) is an extensible framework that is primarily used for authentication users, but in some cases it may also be used for protecting the underlying communication channel. The core functionality of SASL is described in RFC 4422 (`http://www.ietf.org/rfc/rfc4422.txt`), but a number of SASL mechanisms are described in other specifications.

The SASL mechanisms supported by the directory server include:

#### ANONYMOUS SASL mechanism

This mechanism doesn't actually authenticate users to the server, but can be used to destroy a previous authentication session.

#### CRAM-MD5 SASL mechanism

This mechanism provides a way for users to authenticate to the server using a password in a manner that does not expose the password itself. It is similar to, but weaker than, the DIGEST-MD5 SASL mechanism, and doesn't provide any way for ensuring connection integrity or confidentiality.

#### DIGEST-MD5 SASL mechanism

This mechanism provides a way for users to authenticate to the server using a password in a manner that does not expose the password itself. It is similar to, but stronger than, the CRAM-MD5 SASL mechanism, and also provides a way to ensure connection integrity and/or confidentiality.

#### EXTERNAL SASL mechanism

This mechanism provides a way for users to authenticate to the server using information available outside of the LDAP communication that has been performed (for example, the certificate that a client presented when performing Secure Sockets Layer or StartTLS extended operation negotiation).

#### GSSAPI SASL mechanism

This mechanism provides a way for users to authenticate to the server using a Kerberos V5 session. It also provides a mechanism that can be used to ensure connection integrity and/or confidentiality.

#### PLAIN SASL mechanism

This mechanism provides a way for users to authenticate to the server with a username and password. It is similar to the protection offered by simple authentication, but may be more convenient in that users can identify themselves with a username rather than a distinguished name.

### D.18.20 simple paged results control

The simple paged results control is a type of control that can be attached to a search operation to indicate that only a subset of the results should be returned. It may be used to iterate through the search results a page at a time. It is similar to the virtual list view control with the exception that it doesn't require the results to be sorted and can only be used to iterate sequentially through the search results.

The simple paged results control is defined in RFC 2696 (`http://www.ietf.org/rfc/rfc2696.txt`). The same control is used in both the search request and search result done messages. It has an object identifier of 1.2.840.113556.1.4.319, and the value is encoded as follows:

```
realSearchControlValue ::= SEQUENCE {
     size            INTEGER (0..maxInt),
                             -- requested page size from client
                             -- result set size estimate from server
     cookie          OCTET STRING
}
```

For an example of using this control in a search request, see Searching Using the Simple Paged Results Control.

### D.18.21 size limit

The server size limit is a configuration option that controls the maximum number of entries that may be returned from any single search operation. This is a server-wide setting and may be overridden by a per-user configuration in the `ds-rlim-size-limit` operational attribute in the user's entry.

The server size limit (or per-user value) may also be restricted by the size limit element in the search request message.

### D.18.22 smart referral

A smart referral is a special type of entry that can be placed in the directory information tree that reference content in another server and/or location of the DIT. Smart referral entries contain the `referral` object class with one or more instances of the `ref` attribute containing LDAP URL that should be used in the referral.

### D.18.23 StartTLS extended operation

The StartTLS extended operation is a type of extended operation that can be used to initiate a Transport Security Layer-secured communication channel over an otherwise clear-text connection. It allows clients to use the same network port for both secure and insecure communication.

The StartTLS extended operation is defined in RFC 4511 (`http://www.ietf.org/rfc/rfc4511.txt`) and further described in RFC 4513 (`http://www.ietf.org/rfc/rfc4513.txt`). It uses an OID of 1.3.6.1.4.1.1466.20037 with no value. The response includes an OID of 1.3.6.1.4.1.1466.20037 (the same as the request OID) with no value.

### D.18.24 static group

A static group is a type of group in the directory server that defines its membership by providing an explicit set of distinguished name of the entry that are members of the group.

Static groups are very well supported by external clients, but are not as scalable as dynamic group when handling large numbers of members.

### D.18.25 structural object class

A structural object class is one of the primary object class type. A structural object class is special in that it defines the core type for any entry that contains it. An entry must have exactly one structural class (although that structural class may inherit from other structural or abstract object class classes).

The structural object class for an entry may be used by other schema elements for defining constraints on directory data. It may be used by a name form definition to control the attributes used in the relative distinguished name for the entry, and in turn by a DIT structure rule to control the types of parent entries that it may have. The structural object class may also be used by a DIT content rule to control the set of auxiliary object class and required, allowed, and prohibited attribute type for the entry.

### D.18.26 subentry

See LDAP Subentry.

## D.18.27 subschema subentry

A subschema subentry is a special entry within the Directory Server that provides information about the schema elements defined in the server. Attributes in this entry include:

`ldapSyntaxes`
The set of attribute syntax defined in the server schema.

`matchingRules`
The set of matching rule defined in the server schema.

`matchingRuleUse`
The set of matching rule use defined in the server schema.

`attributeTypes`
The set of attribute type defined in the server schema.

`objectClasses`
The set of object class defined in the server schema.

`nameForms`
The set of name form defined in the server schema.

`dITContentRules`
The set of DIT content rule defined in the server schema.

`dITStructureRules`
The set of DIT structure rule defined in the server schema.

Note that all of these are operational attribute and therefore will not be returned unless explicitly requested.

Also note that it is technically possible for directory servers to have multiple subschema subentries with different sets of schema definitions that govern different portions of the directory information tree. The schema that applies to any given entry may be determined by retrieving the `subschemaSubentry` virtual attribute from that entry. The directory server currently supports only a single schema, and by default publishes that schema at `cn=schema`.

## D.18.28 substring assertion

A substring assertion is the argument provided to a substring matching rule in the process of determining whether an attribute has any attribute value that matches a given substring.

The substring assertion contains at least one component from the following set:

- Zero or one subInitial element, which must appear at the beginning of the target value.
- Zero or more subAny elements, which may appear anywhere in the middle of the value. If there are multiple subAny elements, then a matching attribute value must contain all of the subAny elements in the order they appear in the substring assertion with no overlap (i.e., no character in an attribute value can be part of two different substring assertion components). If subInitial and/or subFinal components are present, then none of the subAny elements may overlap with them either.
- Zero or one subFinal element, which must appear at the end of the target value.

The substring assertion is used when processing a substring search filter.

## D.18.29 substring index

A substring index is a type of index that is used to keep track of which entries contain specific substrings. Index keys for a substring index consist of six-character substrings taken from attribute values and the corresponding values are ID list containing the entry ID of the entries containing those substrings. The attribute's substring matching rule is used to normalized value the values for the index keys, and substring indexes cannot be defined for attributes that do not contain substring matching rules.

## D.18.30 substring search filter

A substring search filter is a type of LDAP search filter that can be used to identify entries that contain a value for a given attribute that matches a specified substring. The server will use a substring matching rule to make the determination.

The substring search filter must contain a substring assertion, which will have at least one component from the following types:

- A subInitial component, whose value should be contained at the start of any matching value. There may be either zero or one subInitial component in a substring filter.
- A set of subAny components, whose values should be contained anywhere in the matching value. There may be zero or more subAny components in a substring filter, and they should be contained in the value in the order they appear in the substring filter, after any subInitial component and before any subFinal component.
- A subFinal component, whose value should be contained at the end of a matching value. There may be either zero or one subFinal component in a substring filter.

The string representation of an LDAP substring filter comprises an opening parenthesis followed by the attribute name, an equal sign, the substring assertion with the individual components separated by asterisks, and the closing parenthesis. For example, a substring filter of `(cn=ab*def*mno*stu*yz)` contains a subInitial component of `ab`, subAny components of `def`, `mno`, and `stu`, and a subFinal component of `yz`.

## D.18.31 subtree

There are two definitions for the term "subtree".

The general definition for the term is simply a portion of the directory information tree, including an entry and all of its subordinates.

The term subtree is also described in RFC 3672 (`http://www.ietf.org/rfc/rfc3672.txt`) in the form of a subtree specification. A subtree specification provides a mechanism for grouping entries based on a given set of criteria.

## D.18.32 subtree delete control

The subtree delete control is a type of control that can be attached to a delete operation that will allow the entry and all of its subordinate entries to be deleted. Normal delete operations may target only leaf entry, but the subtree delete control may be used to target non-leaf entry.

The subtree delete request control has an OID of `1.2.840.113556.1.4.805` with no value. There is no corresponding response control.

The following example shows the use of this control to delete the `ou=People,dc=example,dc=com` subtree.

```
$ ldapdelete -p 1389 -h localhost -D cn=directory manager -j pwd-file \
  -J 1.2.840.113556.1.4.805
ou=People,dc=example,dc=com
Processing DELETE request for ou=People,dc=example,dc=com
```

## D.18.33 supported control

A supported control is a mechanism for identifying the request control supported by the Directory Server. The object identifier of these controls are listed in the `supportedControl` attribute of the server's root DSE.

For a list of all controls currently supported in Oracle Unified Directory, see Supported LDAP Controls

## D.18.34 supported extension

A supported extension is a mechanism for identifying the extended operation supported by the Directory Server. The object identifier of these extended operations are listed in the `supportedExtension` attribute of the server's root DSE.

For a list of all supported extensions for the directory server, see Supported Extended Operations.

## D.18.35 supported feature

A supported feature is a mechanism for identifying optional capabilities that the Directory Server supports. A number of the features supported by the server are listed in the `supportedFeatures` attribute of the server's root DSE, which lists the object identifier of the supported features.

Some of the supported features for the directory server include:

**1.3.6.1.4.1.4203.1.5.1**
Indicates that the server supports the use of the + indicator when requesting all operational attribute as specified in RFC 3673 (`http://www.ietf.org/rfc/rfc3673.txt`).

**1.3.6.1.4.1.4203.1.5.2**
Indicates that the server supports the ability to include one or more object class names in the set of search attributes as specified in RFC 4529 (`http://www.ietf.org/rfc/rfc4529.txt`).

**1.3.6.1.1.14**
Indicates that the server supports the increment modification type, which is part of the increment modify extension as described in RFC 4525 (`http://www.ietf.org/rfc/rfc4525.txt`).

**1.3.6.1.4.1.4203.1.5.3**
Indicates that the server supports LDAP true filter and LDAP false filter as described in RFC 4526 (`http://www.ietf.org/rfc/rfc4526.txt`).

### D.18.36 synchronization

Data synchronization is a mechanism for keeping track of changes in the directory environment and allowing them to be reflected elsewhere.

The primary type of data synchronization provided by the directory server is replication.

## D.19 T

### D.19.1 task

A task provides a set of logic for performing some type of processing in the server. Tasks are generally used to perform administrative functions within the server. Examples of tasks available for use include:

- Adding a new file to the server schema
- backup up the contents of a server back end
- restore a previous backup
- Performing an LDIF import operation
- Performing an LDIF export operation
- Initializing a replica in the server replication environment
- Performing an in-core restart
- Performing a server shutdown

Tasks can be recurring, that is scheduled to execute at regular intervals according to a specific schedule. For example, backup tasks can be made recurring in order to back up the server data on a regular basis. For information about scheduling tasks, see Scheduling and Configuring Tasks

### D.19.2 time limit

The server time limit is a configuration option that controls the maximum length of time in seconds that the server may spend processing a search operation. This is a server-wide setting and may be overridden by a per-user configuration in the `ds-rlim-time-limit` operational attribute in the user's entry.

The server time limit (or per-user value) may also be restricted by the time limit element in the search request message.

### D.19.3 transaction

A transaction is a collection of one or more read and/or write operations that occur within a database. Transactions may be described by the acronym ACID, which stands for atomicity, consistency, isolation, and durability. The directory server uses transactions in the Berkeley DB Java Edition to ensure that multiple changes made as part of a single LDAP operation (for example, updates to both the id2entry database and to index).

Even though the Directory Server uses transactions internally for its operations in the database, it does not currently expose a transactional mechanism that allows clients to perform several operations as a single atomic unit. There is an Internet Draft that

describes a potential mechanism for exposing transactions (draft-zeilenga-ldap-txn), but the directory server does not currently support this capability.

### D.19.4 Transport Security Layer

The Transport Security Layer (TLS) is a mechanism for securing network communication between clients and servers. It is the name given to the standardized form of the Secure Sockets Layer.

In most cases, the term "SSL" is preferred over "TLS" because it is the more popular term, and also to avoid confusion with the StartTLS extended operation.

### D.19.5 true filter

See LDAP true filter

### D.19.6 trust manager provider

A trust manager provider is a component of the server that can provide information that can be used to determine whether to trust certificates presented to the server.

See the Trust Manager Provider Configuration (`http://www.opends.org/promoted-builds/latest/OpenDS/build/docgen/configuration_guide/trust-manager-provider.html`) for information about the trust manager providers available for use in the directory server.

### D.19.7 typesOnly flag

The TypesOnly flag is an element of an search operation that indicates whether attributes returned as part of search result entry should include only the attribute description or both the attribute description and the attribute value.

## D.20 U

### D.20.1 unbind operation

The LDAP unbind operation is used to indicate that the client wants to disconnect from the server.

Note that the unbind operation cannot be used to destroy an authentication session while leaving the underlying connection established. If the client does not close the connection after sending an unbind request, then the server will. If there is a need to revert a connection to an unauthenticated state, then an anonymous bind operation should be performed.

The LDAP unbind request protocol op is defined as follows:

```
UnbindRequest ::= [APPLICATION 2] NULL
```

An unbind request does not contain any elements, and the server will not send a response to an unbind request.

### D.20.2 unindexed search

An unindexed search is one that cannot be processed using the set of index defined in the server. It will necessitate iterating through most or all of the entries in the database.

Unindexed searches can be expensive for the server to process, users will only be allowed to perform unindexed searches if they have the `unindexed-search` privilege.

For more information, see Indexing Directory Data.

### D.20.3 UNIX crypt algorithm

The UNIX crypt algorithm is a standard mechanism for encoding user passwords using a DES-based encryption scheme that ultimately results in a one-way message digest. It is called the "UNIX crypt" algorithm because it has historically been used as the default mechanism for encoding passwords in UNIX-based systems.

Note that the UNIX crypt algorithm is considered weak because it is based on a 56-bit encryption algorithm and uses only a 12-bit salt. Therefore, it should only be used in cases where clients expect to be able to retrieve the password from the server and compare its value against what the user supplied instead of attempting to verify it using an bind operation.

### D.20.4 unsolicited notification

An unsolicited notification is a type of extended operation message that is special in that the server generates this kind of message without any corresponding request from the client. It may be used to notify the client of some important information.

The directory server currently supports a single unsolicited notification: the notice of disconnection unsolicited notification, which can be used to inform the client that the server is closing the connection.

### D.20.5 URL

See URL.

### D.20.6 user attribute

A user attribute is an attribute type with an attribute usage of `userApplications`. User attributes are used for actually storing information in the directory, as opposed to operational attribute which are used for storing state information used for internal server processing.

Whenever a search operation does not request any specific attributes to be returned, then all user attributes in matching entries will be returned. An explicit value of `*` (the asterisk) may also be included to explicitly include all user attributes.

## D.21 V

### D.21.1 virtual attribute

A virtual attribute is a type of attribute in which the attribute value are not actually stored in the back end but are instead dynamically generated in some manner. The

values can be obtained in various manners, depending on the type of virtual attribute. Some virtual attributes use a hard-coded value, while others compute their values at runtime based on some kind of logic.

See the Virtual Attribute Configuration (http://www.opends.org/promoted-builds/latest/OpenDS/build/docgen/configuration_guide/virtual-attribute.html) for information about the types of virtual attributes available for use in the directory server.

## D.21.2 virtual attributes only control

The virtual attributes only control requests that the server include only virtual attribute in matching entries. That is, real attributes are excluded from search result entry.

The virtual attributes only control has a request object identifier of `2.16.840.1.113730.3.4.19` and no value.

The following example shows a search on the base DN without the virtual attributes only control:

```
$ ldapsearch -p 1389 -D "cn=directory manager" -j pwd-file -b "dc=example,dc=com"
\
  -s base "objectclass=*"
version: 1
dn: dc=example,dc=com
objectClass: domain
objectClass: top
dc: example
```

The following example shows the same search with the virtual attributes only control:

```
$ ldapsearch -p 1389 -D "cn=directory manager" -j pwd-file \
  -J "2.16.840.1.113730.3.4.19"   -b "dc=example,dc=com" -s base "objectclass=*"
version: 1
dn: dc=example,dc=com
```

## D.21.3 virtual directory

A virtual directory is a type of network daemon that communicates with clients using Lightweight Directory Access Protocol but obtains the underlying data from a combination of different sources. Virtual directories may have a number of different capabilities, including:

- Providing an LDAP front end to a different repository, like a relational database or a flat file
- Providing a mechanism to merge data from multiple repositories

## D.21.4 virtual list view control

The virtual list view (VLV) control can be attached to a search operation to indicate that only a subset of the results are to be returned. It can be used to iterate through the search results a page at a time. It is similar to the simple paged results control with the exception that it can be used to retrieve an arbitrary subset of the results from the server, and it requires that the search request also include the server-side sort control to ensure that the results are consistently sorted across requests.

The VLV control is defined in draft-ietf-ldapext-ldapv3-vlv-09 (http://tools.ietf.org/html/draft-ietf-ldapext-ldapv3-vlv-09). The request control has an object identifier of 2.16.840.1.113730.3.4.9 and the value is encoded as follows:

```
VirtualListViewRequest ::= SEQUENCE {
     beforeCount    INTEGER (0..maxInt),
     afterCount     INTEGER (0..maxInt),
     target       CHOICE {
          byOffset        [0] SEQUENCE {
               offset          INTEGER (1 .. maxInt),
               contentCount    INTEGER (0 .. maxInt) },
          greaterThanOrEqual [1] AssertionValue },
     contextID     OCTET STRING OPTIONAL }
```

The response control has an OID of 2.16.840.1.113730.3.4.10 and the value is encoded as shown below:

```
VirtualListViewResponse ::= SEQUENCE {
     targetPosition    INTEGER (0 .. maxInt),
     contentCount     INTEGER (0 .. maxInt),
     virtualListViewResult ENUMERATED {
          success (0),
          operationsError (1),
          protocolError (3),
          unwillingToPerform (53),
          insufficientAccessRights (50),
          timeLimitExceeded (3),
          adminLimitExceeded (11),
          innapropriateMatching (18),
          sortControlMissing (60),
          offsetRangeError (61),
         other(80),
          ... },
     contextID     OCTET STRING OPTIONAL }
```

For an example of using this control in a search request, see Searching Using the Virtual List View Control

## D.21.5 virtual static group

A virtual static group is a special type of group that appears to be static group to external clients but obtains its membership information from another group (like a dynamic group) in the server.

Virtual static groups are primarily used in cases where a client application only supports static groups but have a very large number of members that are better suited for maintaining in a dynamic group.

## D.21.6 VLV index

A virtual list view (VLV) index is a mechanism used by the Directory Server database that can be used to efficiently process searches with virtual list view control. A VLV index effectively notifies the server that a virtual list view, with specific query and sort parameters, will be performed. This index also allows the server to collect and maintain the information required to make using the virtual list view faster. A VLV index stores sorted blocks of ID list, which are a set of entry ID and the attribute values of the entry to sort on.

# D.22 W

## D.22.1 "Who Am I?" extended operation

The "Who Am I?" extended operation provides an extended operation for determining the authorization identity of a client connection. It is defined in RFC 4532 (http://www.ietf.org/rfc/rfc4532.txt).

The request object identifier for the "Who Am I?" extended operation is 1.3.6.1.4.1.4203.1.11.3, and there should not be a request value. The response should not include a response OID, and the value should be a string containing the client's authorization identity (or it may be an empty string if the authorization identity is that of the anonymous user).

The information provided by the "Who Am I?" extended operation is similar to that provided by the authorization identity control except that it can be used at any time after the client has authenticated, whereas the authorization identity control can only be included with a bind request.

## D.22.2 work queue

The Directory Server work queue is the mechanism that it uses to keep track of outstanding requests and ensuring that they are processed in an appropriate manner. The work queue functionality is provided by an extensible API, but the default implementation is relatively simple: a queue is serviced by a number of worker thread. As long as there are free worker threads, then the queue will generally remain empty. If all worker threads are busy, then subsequent requests will be placed in the work queue so that they are processed in a FIFO manner.

## D.22.3 worker thread

A worker thread is a thread used to process requests in the Directory Server. Worker threads are associated with the work queue, and they will operate in a loop that includes picking up a request from the queue (waiting for a request to arrive if necessary), processing that request appropriate, and then returning to the queue for the next request.

## D.22.4 workflow

A workflow defines the processing for a given naming context. The overall processing is split into a set of ordered and synchronized tasks, defined by *workflow elements*.

## D.22.5 workflow element

A workflow element is the key building block of a workflow processing. It defines how the client request sent to the server will be treated. The workflow elements implement the main tasks in the proxy server, including for example, load balancing and distribution.

## D.22.6 writability mode

The writability mode of the Directory Server is used to control whether write operations are allowed. The writability mode configuration can be restricted to a single back end or it can apply to the entire server.

The following writability modes are available:

**enabled**
The server attempts to process all write operations

**disabled**
The server rejects all write operations

**internal-only**
The server attempts to process write operations initiated as internal operations or through synchronization but rejects any request coming from an external client

An entryDN is an operational attribute that provides a copy of the entry's current DN. Because a DN is not an attribute of the entry, it cannot be used to perform attribute value assertions. The entryDN provides a mechanism to access an entry's DN and is described in RFC 5020 (http://www.ietf.org/rfc/rfc5020.txt).