

---

# **UnboundID® Directory Proxy Server Administration Guide**

Version 3.1

---



Version 3.1  
April 13, 2011

# Copyright

Copyright © 2011 UnboundID Corporation  
All rights reserved

This document constitutes an unpublished, copyrighted work and contains valuable trade secrets and other confidential information belonging to UnboundID Corporation. None of the foregoing material may be copied, duplicated or disclosed to third parties without the express written permission of UnboundID Corporation.

This distribution may include materials developed by third parties. Third-party URLs are also referenced in this document. UnboundID is not responsible for the availability of third-party web sites mentioned in this document. UnboundID does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. UnboundID will not be responsible or liable for any actual or alleged damage or loss caused or alleged to be caused by or in connection with use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

"UnboundID" is a registered trademark of UnboundID Corporation. UNIX® is a registered trademark in the United States and other countries, licensed exclusively through The Open Group. All other registered and unregistered trademarks in this document are the sole property of their respective owners.

The contents of this publication are presented for information purposes only and is provided "as is". While every effort has been made to ensure the accuracy of the contents, the contents are not to be construed as warranties or guarantees, expressed or implied, regarding the products or services described herein or their use or applicability. We reserve the right to modify or improve the design or specifications of such products at any time without notice.

---

# Contents

<b>Chapter 1 Introduction .....</b>	<b>7</b>
Overview of UnboundID Directory Proxy Server Features .....	7
Overview of Directory Proxy Server Components and Terminology .....	9
About Locations .....	9
About LDAP External Servers .....	9
About LDAP Health Checks .....	10
About Load-Balancing Algorithms .....	11
About Proxy Transformations .....	12
About Request Processors .....	13
About Server Affinity Providers .....	13
About Subtree Views .....	14
About the Connection Pools .....	14
About Client Connection Policies .....	15
About Entry Balancing .....	15
UnboundID Directory Proxy Server Component Architecture .....	15
Architecture of a Simple Directory Proxy Server Deployment .....	16
Architecture of an Entry-Balancing Proxy Server Deployment .....	16
UnboundID Directory Proxy Server Configuration Overview .....	18
<b>Chapter 2 Installing the Directory Proxy Server .....</b>	<b>21</b>
Before You Begin .....	21
Defining a Naming Strategy .....	21
Software Requirements: Java .....	22
Preparing the Operating System .....	25
Getting the Installation Packages .....	27
Installing the Directory Proxy Server .....	28
About the setup Tool .....	28
Installing the First Directory Proxy Server in Interactive Mode .....	28
Installing the First Directory Proxy Server in Non-Interactive Mode .....	31
Installing Subsequent Directory Proxy Servers in Interactive Mode .....	33
Installing Subsequent Directory Proxy Servers in Non-Interactive Mode .....	36
Running the Directory Proxy Server .....	36
Stopping the Directory Proxy Server .....	37
Scheduling a Directory Proxy Server Shutdown .....	37
Restarting the Directory Proxy Server .....	37
Uninstalling the Directory Proxy Server .....	38
Uninstalling the Directory Proxy Server in Interactive Command-Line Mode .....	38
Uninstalling the Directory Proxy Server in Non-Interactive Mode .....	39
Updating the Directory Proxy Server .....	40
Reverting an Update .....	41
Installing the Directory Proxy Management Console .....	41
Installing the Console .....	42
Working With the Directory Proxy Management Console .....	44
Logging into the Directory Proxy Management Console .....	44

<b>Chapter 3 Configuring the Directory Proxy Server .....</b>	<b>47</b>
Overview of the Configuration and Management Tools .....	47
Using the create-initial-proxy-config Tool .....	48
Using dsconfig to Configure the Directory Proxy Server .....	54
Using dsconfig in Interactive Command-Line Mode .....	54
Working with dsconfig Object Complexity Menus .....	54
Using dsconfig in Non-Interactive Command-Line Mode .....	56
Getting the Equivalent dsconfig Non-Interactive Mode Command .....	57
Using dsconfig in Batch Mode .....	57
Configuring Global Administrator Accounts .....	59
Configuring Administrative Server Groups .....	60
Configuring a Server Group .....	60
Configuring Root User Accounts .....	63
Default Root Privileges .....	64
Viewing the Default Root User Privileges Using dsconfig .....	65
Modifying a Root User Password .....	65
Creating a Root User .....	66
Modifying Individual Root User Privileges .....	66
Configuring Locations .....	67
Configuring Server Health Checks .....	71
About the Default Health Checks .....	72
About Creating a Custom Health Check .....	72
Configuring LDAP External Servers .....	76
Configuring Load Balancing .....	80
Configuring Proxy Transformations .....	82
Configuring Request Processors .....	84
Passing LDAP Controls With the Proxying Request Processor .....	86
Configuring Server Affinity .....	88
Configuring Subtree Views .....	90
Configuring Client Connection Policies .....	92
Understanding the Client Connection Policy .....	93
Setting Resource Limits .....	94
Defining the Operation Rate .....	94
Client Connection Policy Deployment Example .....	95
Configuring Entry Balancing .....	98
Determining How to Balance Your Data .....	99
Configuring an Entry Balancing Placement Algorithm .....	99
<b>Chapter 4 Managing Directory Proxy Server .....</b>	<b>101</b>
Managing Logs .....	101
About the Default Logs .....	101
Types of Log Publishers .....	107
Creating New Log Publishers .....	108
Configuring Log Rotation .....	109
Configuring Log Retention .....	110
Managing the Global Index in Entry Balancing Configurations .....	110
Reloading the Global Index .....	111
Monitoring the Size of the Global Index .....	111
Sizing the Global Index .....	112
Priming the Global Index on Start Up .....	113
Priming or Reloading the Global Index from Sun Directory Servers .....	114
Setting Resource Limits .....	114
Monitoring Directory Proxy Server .....	115
Monitoring Server Status Using the status Tool .....	116

---

About the Monitor Entries .....	117
Using the Monitoring Interfaces .....	118
Monitoring with JMX .....	119
Monitoring Over LDAP .....	121
Monitoring Using SNMP .....	121
Profiling Server Performance Using the Periodic Stats Logger .....	133
Configuring a Custom Periodic Stats Logger .....	135
Working with Account Status Notifications .....	140
Working with the Error Log Account Status Notification Handler .....	141
Working with the SMTP Account Status Notification Handler .....	142
Associating Account Status Notification Handlers with Password Policies .....	143
Working with Administrative Alert Handlers .....	144
Configuring the JMX Connection Handler and Alert Handler .....	144
Configuring the SMTP Alert Handler .....	145
Configuring the SNMP Alert Handler .....	146
Working with Virtual Attributes .....	146
<b>Chapter 5 Managing Password Policies .....</b>	<b>149</b>
Viewing Password Policies .....	149
Managing the Default Password Policy Properties .....	151
Modifying the Default Password Policy .....	152
Creating a New Password Policy .....	152
Modifying a User's Password .....	155
Managing Users Accounts .....	155
Disabling Password Policy Properties .....	157
Managing Password Validators .....	157
Managing Password Validators .....	158
Configuring the Attribute Value Password Validator .....	159
Configuring the Character Set Password Validator .....	160
Configuring the Dictionary Password Validator .....	160
Configuring the Length-Based Password Validator .....	161
Configuring the Regular Expression Password Validator .....	162
Configuring the Repeated Character Password Validator .....	163
Configuring the Similarity-Based Password Validator .....	164
Configuring the Unique Characters Password Validator .....	165
<b>Chapter 6 Managing Access Control .....</b>	<b>167</b>
Working with Access Control in the Proxy Server .....	167
Configuring Access Control with Entry Balancing .....	168
Configuring Access Control .....	170
Key Access Control Features .....	170
Summary of Access Control Keywords .....	175
Working with Privileges .....	177
Available Privileges .....	178
Privileges Automatically Granted to Root Users .....	180
Assigning Privileges to Normal Users and Individual Root Users .....	181
Disabling Privileges .....	181
<b>Chapter 7 Managing Replication in an Entry Balancing Deployment .....</b>	<b>183</b>
Overview of Replicating in an Entry Balanced Environment .....	183
Replication Prerequisites in an Entry-Balanced Deployment .....	184
About the --restricted Argument of the dsreplication Command-Line Tool .....	185
Checking the Status of Replication in an Entry-Balanced Deployment .....	185

---

Example of Configuring Entry-Balancing Replication .....	186
Assumptions .....	186
Configuration Summary .....	187
Detaching a Replication Set from a Topology .....	193
<b>Chapter 8 Deploying the Directory Proxy Server .....</b>	<b>197</b>
Creating a Standard Multi-Location Deployment .....	197
Overview of Deployment Steps .....	198
Installing the First Directory Proxy Server .....	198
Configuring the First Directory Proxy Server Using the create-initial-proxy-config Tool .....	199
Configuring the Additional Proxy Servers .....	206
Testing External Server Communications After Initial Setup .....	208
Testing a Simulated External Server Failure .....	210
Expanding the Deployment .....	210
Overview of Deployment Steps .....	211
Testing a Simulated External Server Failure .....	239
Merging Two Data Sets Using Proxy Transformations .....	240
Overview of the Attribute and DN Mapping .....	240
About Mapping Multiple Source DNs to the Same Target DN .....	241
An Example of a Migrated Sample Customer Entry .....	242
Overview of Deployment Steps .....	243
Installing the Schema File .....	243
Creating and Configuring the Proxy Transformations .....	245
Creating a Request Processor to Manage the Proxy Transformations .....	262
Creating a Subtree View .....	267
Editing the Client Connection Policy .....	269
Testing Proxy Transformations .....	273
Deploying an Entry-Balancing Proxy Configuration .....	274
Overview of Deployment Steps .....	275
Installing Directory Proxy Server .....	275
Configuring the Entry-Balancing Directory Proxy Server .....	276
Configuring the Placement Algorithm Using a Batch File .....	285
<b>Chapter 9 Troubleshooting the Directory Proxy Server .....</b>	<b>287</b>
Debugging the Directory Proxy Server .....	287
Working with the Troubleshooting Tools .....	287
Working with the Collect Support Data Tool .....	288
Troubleshooting Tools Provided by the Directory Proxy Server .....	291
Troubleshooting Resources for Java Applications .....	293
Troubleshooting Resources in the Operating System .....	297
Common Problems and Potential Solutions .....	303
The Server Will Not Run Setup .....	303
The Server Will Not Start .....	304
The Server Has Crashed or Shut Itself Down .....	308
The Server Will Not Accept Client Connections .....	308
The Server is Unresponsive .....	309
The Server Is Slow to Respond to Client Requests .....	310
The Server Returns Error Responses to Client Requests .....	311
The Server Routes Requests in an Unexpected Manner .....	311
Problems with the Directory Proxy Server Management Console .....	312
Global Index Growing Too Large .....	312
Forgotten Proxy User Password .....	312
Providing Information for Support Cases .....	313
<b>Index .....</b>	<b>315</b>

---

# 1

## Introduction

UnboundID® Directory Proxy Server is a fast and scalable LDAPv3 gateway for the UnboundID® Directory Server. The directory proxy server architecture can be configured to control how client requests are routed to backend servers.

This chapter provides an overview of the directory proxy server features and components. It contains the following sections:

- Overview of UnboundID Directory Proxy Server Features
- Overview of Directory Proxy Server Components and Terminology
- UnboundID Directory Proxy Server Component Architecture
- UnboundID Directory Proxy Server Configuration Overview

## Overview of UnboundID Directory Proxy Server Features

The UnboundID Directory Proxy Server is a fast, scalable, and easy-to-use LDAP proxy server that provides high availability and additional security for the UnboundID Directory Server, while remaining largely invisible to client applications. From a client perspective, request processing is the same, whether communicating with the directory server directly or going through the directory proxy server.

The UnboundID Directory Proxy Server provides the following set of features:

- **High availability.** The directory proxy server allows you to transparently fail over between servers if a problem occurs, as well as ensuring that the workload is balanced across the topology. If a client does not support following referrals, the proxy server can follow them on the client's behalf.
- **Data mapping and transformation.** The directory proxy server can do DN mapping and attribute mapping to allow clients to interact with the server using older names for directory content. The directory proxy server allows clients to continue working when they would not be able to work directly with the directory server, either because of changes that have occurred at the directory layer or to inherent design limitations in the clients.

- **Horizontal scalability and performance.** Reads can be horizontally scaled using load balancing. In large data centers, if the data set is too large to be cached or to provide horizontal scalability for writes, the directory proxy server can automatically split the data across multiple systems. This feature allows the proxy to improve scalability and performance of the directory environment.
- **Load balancing and failover.** You can spread the workload across proxy servers in a large data center using the directory proxy server's load-balancing algorithms. Load balancing is also useful when a server becomes degraded or non-responsive, because client process requesting is directed to a different server.
- **Security and access control.** The directory proxy server can add additional firewall capabilities, as well as constraints and filtering to help protect the directory server from attacks. You can use a proxy server in a DMZ as opposed to allowing clients to directly access the directory server in the internal network or providing the data in the DMZ. It can help provide secure access to the data and you can define what actions clients are allowed to do. For example, you can prevent clients from making modifications to data when connected via a VPN no matter what their identity or permissions.
- **Tracking of operations across the environment.** In the past, administrators have commonly complained that when they see a request in the access log, they have no idea where it came from and cannot track it back to a particular client. The directory proxy server contains controls that allow administrators to track requests back to the client that issued them. Whenever the proxy server forwards a request to the directory server, it includes a control in the request so that the directory server's access log has the IP address of the client, address and connection ID of the proxy server. In the response back to the client, it similarly includes information about the directory server that processed the request, such as the connection ID and operation ID. This feature makes it easier for administrators to keep track of what is going on in their environment.
- **Monitoring and management tools.** Because the directory proxy server uses many of the components of the UnboundID Directory Server, it can leverage them to provide protocol support, logging, management tools for configuration and monitoring, schema, and so on. You can use the `dsconfig` tool and the Web-based administration console to manage the directory proxy server.
- **Multi-platform support.** The UnboundID®Server is a pure Java application and is certified VMWare Ready™. It is intended to run within the Java Virtual Machine on any Java 6 Standard Edition (SE) or Enterprise Edition (EE) certified platform including, but not limited to, the following:
  - ❑ RedHat® Enterprise Linux® 5.5 (natively or running under VMware® ESX 4.0 Update 2)
  - ❑ Oracle® Solaris™ 10 update 9
  - ❑ RedHat®/CentOS 5.5 & 6
  - ❑ Windows Server® 2003 & 2008
  - ❑ Novell® SUSE® 11.3

Any known OS or JDK platform-specific issues will be documented in the release notes distributed with the product. Direct any questions or requests for additional platform certifications to



# Overview of Directory Proxy Server Components and Terminology

The proxy consists of the following components and functionality that provide the proxy capabilities:

- Locations
- LDAP external servers
- LDAP health checks
- Load-balancing algorithms
- Data transformations
- Request processors
- Server affinity providers
- Subtree views
- Connection pools
- Client connection policies
- Entry balancing

This section describes each component in more detail.

## About Locations

Locations define a group of servers with similar response time characteristics. Each location consists of a name and an ordered list of preferred failover locations. The directory proxy server and each of the backend LDAP external servers can be assigned locations. These locations can be taken into account when deciding how to route requests, so that the server prefers to forward requests to directory servers in the same data center over those in remote locations. As a rule of thumb, if you have multiple data centers then you should have a separate location for each one. In most environments, all directory proxy server instances should have the same configuration except for the attribute that specifies the location of the directory proxy server itself.

For example, a deployment consists of three data centers, one in New York, another in Chicago, and another in Los Angeles. In the New York data center, applications which reside in this data center prefer communicating with directories in this data center. If none of the servers are available, it prefers to failover to the data center in Chicago rather than the data center in Los Angeles. So the New York location contains an ordered list in which the Chicago location is preferred over the Los Angeles data center for failover.

For information about configuring locations, see “Using dsconfig to Configure the Directory Proxy Server” on page 54.

## About LDAP External Servers

You can configure information about the directory server instances accessed by the UnboundID Directory Proxy Server. This configuration information includes the following:

- Server connection information, such as IP address, port, and security layer
- Location
- Authentication information
- Methods for authenticating and authorizing clients.
- Server-specific health checks
- Types of operations allowed. For example, some LDAP external servers may allow only reads and others allow reads and writes, so the proxy server can recognize this and accommodate it.

The UnboundID Directory Proxy Server allows you to configure different types of LDAP external servers. The default configuration for each type is tuned to be the best possible configuration for each.

## About LDAP Health Checks

The LDAP health check component provides information about the availability of LDAP external servers. The health check result includes a server state, which can be one of the following:

- Available: completely accessible for use
- Degraded: the server may be used if necessary, but has a condition which may make it less desirable than other servers (for example, it is slow to respond or has fallen behind in replication)
- Unavailable: completely unsuitable for use (for example, the server is offline or is missing critical data)

Health check results also include a numeric score, which has a value between 1 and 10, that can help rank servers with the same state. For example, if two servers are available and one has a score of 8 and the other a score of 7, the proxy can be configured to prefer the server with the higher score.

The directory proxy server periodically invokes health checks to monitor each LDAP external server, and may also initiate health checks in response to failed operations. It checks the health of the LDAP external servers at intervals configured in the LDAP server's **health-check-frequency** property. However, the directory proxy server has safeguards in place to ensure that only one health check is in progress at any time against a backend server to avoid affecting its ability to process other requests.

The results of health checks performed by the proxy server are made available to the load-balancing algorithms so that they may be taken into account when determining where to send requests. The proxy server will attempt to use servers with a state of available before trying servers with a state of degraded. It will never attempt to use servers with a state of unavailable. Some load-balancing algorithms may also take the health check score into account, such as the health-weighted load-balancing algorithm, which prefers servers with higher scores over those with lower scores. Other load-balancing algorithms do not use the health check scores, such as the round-robin load-balancing algorithm, which balances the load equally among servers with the same state, regardless of the health check score.

In some cases, an LDAP health check may define different sets of criteria for promoting and demoting the state of a server. So, a degraded server may need to meet more stringent requirements to be reclassified as available than it originally took to be considered degraded. For example, if response time is used in the process of determining the health of a server, then the directory proxy server may have a faster response time threshold for transitioning a server from degraded back to available than the threshold used to consider it degraded in the first place. This threshold difference can help avoid cases in which a server repeatedly transitions between the two states because it is operating near the threshold.

For example, the health check used to measure search response time is configured to mark any server to be marked degraded when the search response time is greater than 1 second. You can then configure that the response time must be less than 500 ms before the server is made available again, so that the proxy server does not flip back and forth between available and degraded.

UnboundID Directory Proxy Server provides the following health checks:

- Measure the response time for searches and examine the entry contents. For example, the health check might retrieve a monitoring entry from a server and base the health check result on whether the entry was returned, how long it took to be returned, and whether the value of the returned entry matches what was expected.
- Monitor the replication backlog. If a server falls too far behind in replication, then the directory proxy server can stop sending requests to it. A server is classified as degraded or unavailable if the threshold is reached for the number of missing changes, the age of the oldest missing change, or both.
- Consume directory server administrative alerts. If the directory server indicates there is a problem, for example it enters lock-down mode, then the directory proxy server will stop sending requests to the server. The directory proxy server detects administrative alerts as soon as they are issued and then checks to see if the alert is associated with a server becoming degraded or unavailable. This health check also confirms if a server is placed in the degraded or unavailable state without an alert being issued.
- Monitor the busyness of the server. If a server becomes too busy, then it may be marked degraded or unavailable so that less heavily-loaded servers may be preferred.

For information about configuring health checks, see “Configuring Server Health Checks” on page 71. To associate a health check with an LDAP external server and set the health check frequency, you must configure the **health-check** and **health-check-frequency** properties of the LDAP external server. See “To Configure an External Server Using dsconfig” on page 77 for information about configuring the properties of the external server.

## About Load-Balancing Algorithms

Load-balancing algorithms are used to determine which server in a set of similar servers should be used to process a client request. The algorithm can take the following criteria into account:

- Consider the location of the server. Servers in the same location as the directory proxy server can be preferred over those in alternate locations.
- Consider the health of the server. Servers that are available are preferred over those that are degraded. In some cases, the health check score may also be used to further differentiate between servers with the same health check state.
- Route requests consistently. Requests from a single client may be consistently routed to the same directory server instance to avoid problems such as propagation delay from replication.
- Retry the operation in an alternate server if the request fails or the operation times out. You can control if the retry is allowed and, if so, how many times to retry and the time out interval.

The UnboundID Directory Proxy Server provides the following load-balancing algorithms:

- Fewest locations. Requests are forwarded to the backend server with the fewest operations currently in progress.
- Single server. Requests are always sent to the same server and will not attempt to fail over to another server if the target server is unavailable.
- Round robin. Requests are sent to a given pool of servers.
- Weighted. Administrators explicitly assign numeric weights to individual servers or sets of servers to control how likely they are to be selected for processing requests relative to other servers.
- Health-based weighting. Uses the health check score to assign weights to each of the servers, so that a server with a higher score gets a higher percentage of the traffic than a server with a lower score. The proportion of traffic received is the difference between their health check scores.
- Failover. Requests are always sent to a given server first. If that server fails, then the request is sent to another specified server, and so on through an ordered failover server list.

## About Proxy Transformations

Proxy transformations are used to rewrite requests and responses as they pass through the directory proxy server. Proxy data transformations are helpful for clients that use an old schema or that contain a hard-coded schema.

Proxy transformations can provide DN and attribute mapping altering both requests to the server as well as responses from the server. For example, a client sends a request to `o=example.com` even though the directory server handling the request uses `dc=example,dc=com`. The directory proxy server can transparently remap the request so that the server can process it, and map it back to the original DN of the client request when the value is returned. Or if a client tries to use the attribute `userID`, the directory proxy server can map it to `uid` before send-

ing the request on to the backend LDAP server. The directory proxy server then remaps the response to `userID` when the value is returned.

The directory proxy server also includes a proxy transformation that can be used to suppress a specified attribute, so that it will never be returned to clients. It can also cause the server to reject requests which target that particular attribute. Another proxy transformation can be used to prevent entries that match a given search filter from being returned to clients.

For information about configuring proxy transformations, see “Configuring Proxy Transformations” on page 82.

## About Request Processors

A request processor encapsulates the logic for handling an operation, ensuring that a given operation is handled appropriately. The request processor can either process the operation directly, forward the request to another server, or hand off the request to another request processor.

UnboundID Directory Proxy Server provides the following types of request processor:

- Proxying request processors, which forward operations received by the directory proxy server to other LDAP external servers.
- Entry-balancing request processors, which split data across multiple servers. They determine which set of servers are used to process a given operation. They then hand off operations to proxying request processors so that requests can be forwarded to one of the servers in the set.
- Failover request processors, which perform ordered failover between other types of request processors, sometimes with different behavior for different types of operations. For example, you could use a failover request processor to achieve round-robin load balancing for read operations but failover load-balancing for writes.

For information about configuring request processors, see “Configuring Request Processors” on page 84.

## About Server Affinity Providers

The server affinity provider can be used to establish an affinity to a particular backend server for certain operations. You can configure one of three types of provider:

- Client connection server affinity, so that requests from the same client connection may consistently be routed to the same backend server.
- Client IP address server affinity, so that all requests coming from the same client system will be consistently routed to the same backend server

- Bind DN server affinity, so that all requests from the same user will be consistently routed to the same backend server.

## About Subtree Views

A subtree view can be used to make a portion of the DIT available to a client by associating a request processor with a base DN. Subtree views allow you to route operations concerning one set of data to a particular set of data sources, and operations concerning another set of data to another set of data sources. Multiple subtree views may be involved in processing a request, such as for searches that have a scope that is larger than the subtree view.

The subtree view includes a single base DN used to identify the portion of the DIT. They may have hierarchical relationships, for example one subtree view could be configured for `dc=example,dc=com` and another for `ou=People,dc=example,dc=com`.

For information about configuring a subtree view, see “Configuring Subtree Views” on page 90.

## About the Connection Pools

Based on the type of backend server that you are using, the UnboundID Directory Proxy Server maintains either one or two connection pools to the backend server. It maintains either one pool for all types of operations or two separate pools for processing bind and non-bind operations from clients. When the directory proxy server establishes connections, the proxy authenticates them using whatever authentication mechanism is defined in the configuration of the external server. These connections will be re-used for all types of operations to be forwarded to the backend server. The bind DN and password are configured in the directory proxy server.

Whenever a client sends a bind request to the directory proxy server, the server looks at the type of bind request that was sent. If it is a SASL bind request, then the authentication is processed by the directory proxy server itself and it will not be forwarded to the backend server. However, the directory proxy server may use information contained in the backend server as needed. If the bind request is a simple bind request and the bind DN is within the scope of data supplied by the backend server, then the directory proxy server will forward the client request to the backend server so that it will use the credentials provided by the client.

Regardless of the authentication method that the client uses, the directory proxy server will remember the identity of the client after the authentication is complete and for any subsequent requests sent by that client, it will use the configured authorization method to identify the client to the backend server. Even though the operation is forwarded over a connection that is authenticated as a user defined in the directory proxy server configuration, the request is processed by the backend server under the authority of the end client.

## About Client Connection Policies

Client connection policies define the general behavior the server exhibits when communicating with a set of clients. Each policy consists of the following:

- A set of connection criteria that define which client is associated with the policy based on information the server has about the client, including client address, protocol used, secure communication mechanism, location of the client's entry in the directory and the contents of the client's entry. These criteria are the same as those used for filtered logging. For example, different client connection policies could be established for different classes of users, such as root and non-root users.
- A set of constraints on the type of operations a client may request. You can specify whether a particular type of operation is allowed for clients. For some operation types, such as extended operations, you can allow only a particular subset of an operation type, such as a particular extended operation.
- A set of subtree views that define information about the parts of the DIT the client may access.

When a client connection is established, only one client connection policy is applied. If the criteria for several policies match the same client connection, the evaluation order index is used as a tiebreaker. If no policy matches, the client connection is terminated. If the client binds, changing its identity, or uses StartTLS to convert from an insecure connection to a secure connection, then the connection may be evaluated again to determine if it matches the same or a different client connection policy. The connection can also be terminated if it no longer matches any policy.

For information about configuring a client connection policy, see “Configuring Client Connection Policies” on page 92.

## About Entry Balancing

Entry balancing allows you to automatically spread entries below a common parent among multiple sets of directory servers for improved scalability and performance. Entry balancing can take advantage of a global index, an in-memory cache used to quickly determine which set or sets of servers should be used to process a request based on the entry DNs and/or the attribute values used in the request.

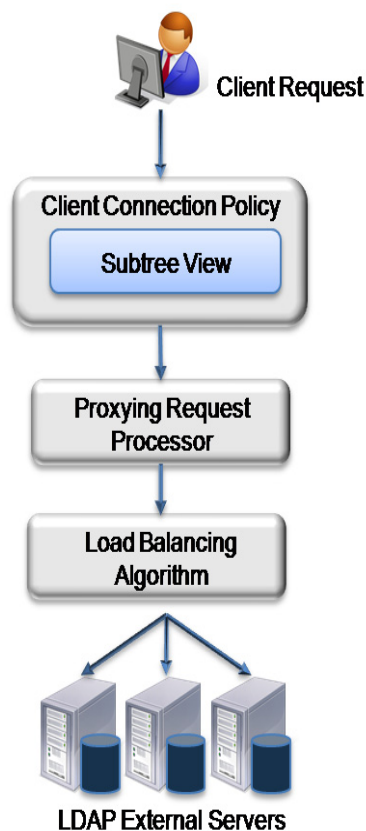
# UnboundID Directory Proxy Server Component Architecture

This section provides an overview of the process flow between the directory proxy server components, for both a simple proxy deployment and an entry balancing deployment.

## Architecture of a Simple Directory Proxy Server Deployment

In a simple directory proxy server deployment, a client request is first processed by a client connection policy as illustrated in Figure 3-1, “Process Flow for Directory Proxy Server”.

FIGURE 3-1. Process Flow for Directory Proxy Server



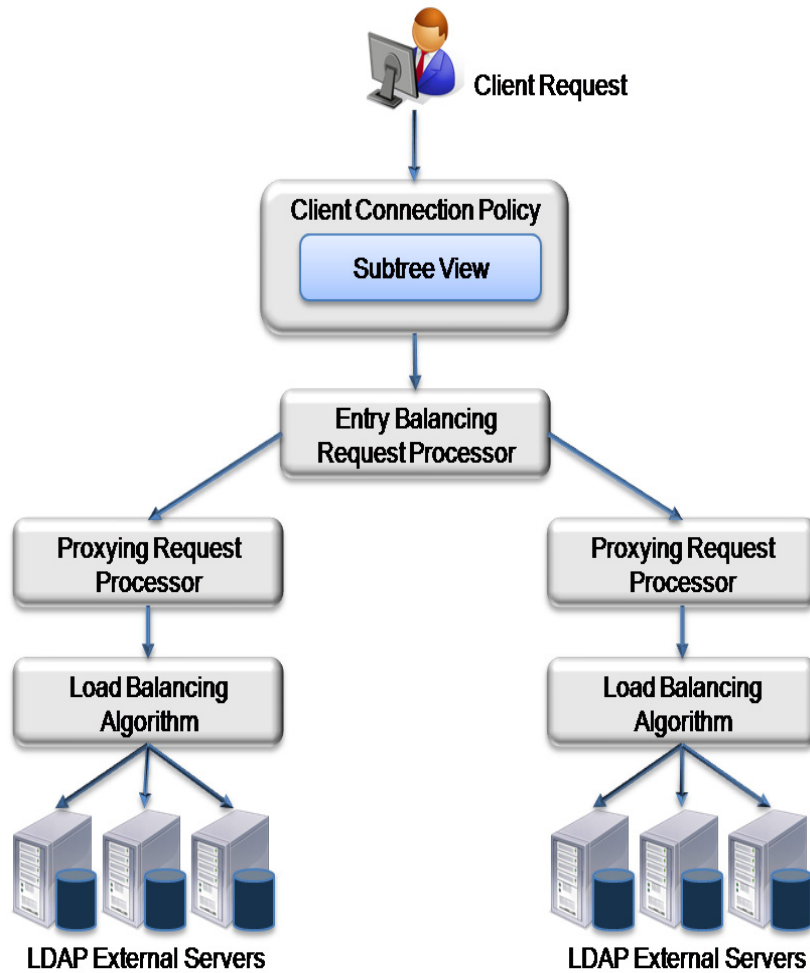
The client connection policy contains a subtree view, which defines the portion of the DIT available to clients. Once the directory proxy server determines that the DIT is available, it passes the request to the request processor, which defines the logic for processing the request. The request processor then passes the request to a load-balancing algorithm, which determines the server in a set of servers responsible for handling the request. Finally, the request is passed to the LDAP external server. The LDAP external server contains properties that define the server’s location in a topology and the health checks used to determine if the server is functioning properly. This information may be used by the load-balancing algorithm in the course of determining how to route requests.

## Architecture of an Entry-Balancing Proxy Server Deployment

Figure 3-2, “Entry-Balancing Directory Proxy Server Process Flow” describes how a client request is treated in an entry-balancing deployment.



FIGURE 3-2. Entry-Balancing Directory Proxy Server Process Flow



Entry balancing is typically used when the data set is too large to fully cache on a single server or when the write performance requirements of an environment are higher than can be achieved with a single replicated set of servers. In such cases, the data may be split across multiple sets of servers, increasing the memory available for caching and the overall write performance in proportion to the number of server sets.

As with a simple proxy deployment, the client request is first processed by the client connection policy, which determines how the directory proxy server communicates with a set of clients. It contains a subtree view that represents the base DN for the entire deployment. The data set splits beneath this base DN.

The request is then passed to the entry-balancing request processor. The entry-balancing request processor contains a global attribute index property, which helps the request processor determine which server set contains the entry and how to properly route the request. It also contains a placement algorithm, which helps it select the server set in which to place new entries created by add requests.

Beneath the entry-balancing request processor are multiple proxying request processors that handle multiple unique sets of data. These request processors pass the request to a load-balancing algorithm, which determines which LDAP external server should handle the request. As with a simple proxy deployment, this LDAP external server contains properties that define the server's location and the health checks used to determine if the server is functioning properly.

## UnboundID Directory Proxy Server Configuration Overview

The configuration of the directory proxy server involves the following steps:

- Configuring the locations for your deployment.

A location is a collection of servers that share access and latency characteristics. For example, your deployment might include two data centers, one in the east and one in the west. These data centers would be configured as two locations in the proxy server. Each location is associated with a name and an ordered list of failover locations, which could be used if none of the servers in the preferred location are available.

- Configuring the proxy server location.

You need to update the configuration to specify the location of the proxy server instance.

- Configuring health checks for the LDAP external servers.

You can configure at what point the proxy server considers an LDAP external server to be available, of degraded availability, or unavailable. Each health check can be configured to be used automatically for all LDAP external servers or for a specified set of servers.

- Configuring the LDAP external servers.

During this step, you define each of the external directory servers, including the server type. You can configure UnboundID Directory Servers, Alcatel-Lucent 8661 Directory Servers, Sun™ Java System Directory Servers, or generic LDAP servers. You also assign the server-specific health checks configured in the previous step.

- Configuring the load-balancing algorithm.

You configure the load-balancing algorithm used by the directory proxy server to determine which server in a set of similar servers should be used to process a client request.

- Configuring the proxying request processor.

In this step, you configure proxying request processors that forward operations received by the directory proxy server to other LDAP external servers.

- Configuring subtree views.

A subtree view defines the portion of the DIT available to a client. Each subtree view can be associated with a load-balancing algorithm to help distribute the work load.

- Configuring the client connection policy.

You configure policies to classify how different client connections are managed by the proxy server. The client connection policy can be used to control the types of operations that a client may perform and the portion of the DIT that the client can access. Restrictions configured in a client connection policy will take precedence over any capabilities granted by access control or privileges.



---

# 2 Installing the Directory Proxy Server

This section describes how to install UnboundID® Directory Proxy Server. It includes pre-installation requirements and considerations.

It includes the following sections:

- Before You Begin
- Getting the Installation Packages
- Installing the Directory Proxy Server
- Running the Directory Proxy Server
- Uninstalling the Directory Proxy Server
- Updating the Directory Proxy Server
- Installing the Directory Proxy Management Console
- Working With the Directory Proxy Management Console

## Before You Begin

The following sections describe requirements and considerations you should make before installing the software and configuring the UnboundID Directory Proxy Server objects.

### Defining a Naming Strategy

The various objects you will be defining in the UnboundID Directory Proxy Server will be specific to a particular location or set of servers. Keep this in mind when you are naming objects, so that they will be easy to identify and group with like objects. For example, all of the servers in the west location could be named using a prefix of “west” so that they are readily identifiable when listed. A health check you want to apply only to these servers could also contain the word “west”, to make it easier to remember to which group of servers it applies.

## Software Requirements: Java

Before you install the proxy server, you need to install Java 6. For optimized performance, the UnboundID Directory Server requires Java 6 (recommended JDK1.6.0\_20) for 32-bit and 64-bit architectures, respectively, depending on your system requirements.

**Important.** We have found that an existing Sun JVM bug in JDK1.6u21 and JDK1.6u22 causes promotion failures that lead to extended stop-the-world garbage collection pauses. We recommend that JDK1.6u20 be used with the directory server and that the use of JDK1.6u21 and JDK1.6u22 not be implemented. If your current system uses JDK1.6u18 to JDK1.6u20, then you must also configure an additional Java property for the `start-ds` command that bypasses this bug. See “To Configure the UseSSE42Intrinsics Java Property” on page 25.

### Installing Java 6

UnboundID Directory Proxy Server requires the latest general available release of Java 6 for 32-bit and 64-bit architectures, respectively, depending on your system requirements. Even if your system already has Java installed, you may want to create a separate Java 6 installation for use by the UnboundID Directory Proxy Server to ensure that updates to the system-wide Java installation do not inadvertently impact the directory server. This setup requires that the JDK™, rather than the JRE™, for both the 32-bit and 64-bit versions or both depending on your operating system, be downloaded.

On Solaris™ systems, if you want to use the 64-bit version of Java, then you need to install both the 32-bit and 64-bit versions. The 64-bit version of Java on Solaris is not a full stand-alone installation, but instead relies on a number of files provided by the 32-bit installation. Therefore, the 32-bit version should be installed first, and then the 64-bit version installed in the same location with the necessary additional files.

On other platforms (Linux®, Microsoft® Windows®), the 32-bit and 64-bit versions of Java contain complete installations. If you only want to run the 64-bit version of Java, then it is not necessary to install the 32-bit JDK. If you do want to have both versions installed, then they should be installed in separate directories, because files cannot coexist in the same directory as they do on Solaris systems.

### To Install Java 6 (Self-Extracting Binary for Solaris)

1. Change to the directory in which Java will be installed.
2. Download the 32-bit JDK self-extracting binary.
3. Make sure to set execute permissions on the 32-bit self-extracting binary.

On SPARC processors:

```
chmod +x jdk-6<update>-solaris-sparc.sh
```

On x86/x64/EM64T processors:

```
chmod +x jdk-6<update>-solaris-i586.sh
```

4. Change to the directory where you would like the files to be installed.
5. Run the 32-bit self-extracting binary.

On SPARC processors:

```
./jdk-6<update>-solaris-sparc.sh
```

On x86/x64/EM64T processors:

```
./jdk-6<update>-solaris-i586.sh
```

6. Read the binary code license. If you agree to its terms, accept it. The JDK files are installed in a directory called `jdk1.6.0<update>` in the current directory.
7. If you plan to install the 64-bit JDK, download the 64-bit self-extracting binary. Otherwise, go to step 12.
8. Make sure that execute permissions are set on the 64-bit self-extracting binary.

On SPARC processors:

```
chmod +x jdk-6<update>-solaris-sparcv9.sh
```

On x86/x64/EM64T processors:

```
chmod +x jdk-6<update>-solaris-x64.sh
```

9. Change to the directory where you ran the self-extracting binary for the 32-bit installation. This is the directory that contains the `jdk1.6.0<update>` directory of the 32-bit JDK.
10. Run the 64-bit self-extracting binary.

On SPARC processors:

```
./jdk-6<update>-solaris-sparcv9.sh
```

On x86/x64/EM64T processors:

```
./jdk-6<update>-solaris-x64.sh
```

11. Read the binary code license. If you agree to its terms, accept it.

The supplemental files for 64-bit support are installed in directories named for the machine architecture model, which are added at several locations within the same `jdk1.6.0<update>` directory where the 32-bit JDK was installed. For example, on SPARC processors, the 64-bit Java VM Library (JVM™) file (`libjvm.so`) is stored in the `jdk/lib/sparcv9/server` directory, whereas the version for x64/EM64T is stored in the `jdk/lib/x64/server` directory.

12. Create a symbolic link to the Java installation in the same directory with a generic name of “java”. When the Directory Proxy Server is installed, it will use a Java path referenced by this symlink. In the future, when you install a newer version of Java, you simply update the symbolic link so that the Directory Proxy Server uses the new version.

```
# ln -s /ds/jdk1.6.0_07 /ds/java
```

13. Make sure that execute permissions are set on the 64-bit self-extracting binary.

### To Install Java 6 (Self-Extracting Binary for Linux)

1. Download and check the downloaded file size to ensure that you have downloaded the full, uncorrupted software bundle.
2. Make sure that execute permissions are set on the self-extracting binary.

Run this command for 32-bit:

```
chmod +x jdk-6<version>-linux-i586.bin
```

Run this command for 64-bit:

```
chmod +x jdk-6_<version>-linux-x64.bin
```

3. Change to the directory where you would like the files to be installed. Run the self-extracting binary. On Linux systems, you do not need to pre-install the 32-bit version of the Java 6 JDK if you want to install the 64-bit binary, the 32-bit and 64-bit binaries are complete installations and can be installed as is.

Run this command for 32-bit:

```
./jdk-6<version>-linux-i586.bin
```

Run this command for 64-bit:

```
./jdk-6_<version>-linux-x64.bin
```

4. Read the binary code license. If you agree to its terms, accept it. The JDK files are installed in a directory called `jdk1.6.0_<version>` in the current directory.
5. Create a symbolic link to the Java installation in the same directory with a generic name of “java.” When the Directory Proxy Server is installed, it will use a Java path referenced by this symlink. In the future, when you install a newer version of Java, you simply update the symbolic link so that the Directory Proxy Server uses the new version.

```
# ln -s /ds/jdk1.6.0_07 /ds/java
```

### To Install Java 6 (Windows Installer)

1. Run the Java 6 JDK Windows Installer for 32-bit or 64-bit, respectively. You must have administrative privileges to install the JDK. If you downloaded the file instead of running it directly from the web site, double-click the installer’s icon, and follow the instructions from the installer prompts.

Run this command for 32-bit:

```
jdk-6<version>-windows-i586.bin
```

Run this command for 64-bit:

```
jdk-6_0_<version>-windows-x64.bin
```

2. The installer asks you to reboot your computer.
3. Optionally, update the **PATH** variable.



- a. On Windows XP/Vista, click **Start > Control Panel > System**. On Windows 2000 Server, click **Start > Settings > Control Panel > System**.
- b. Click **Advanced > Environment Variables**.
- c. Add the location of the JDK bin folder to **PATH** in User Variables and System Variables. A typical value for **PATH** is:  
`C:\Program Files\Java\jdk1.6.0_<version>\bin`

## To Configure the UseSSE42Intrinsics Java Property

We have found that a known Oracle (formerly Sun) JVM bug in versions 1.6u21 and 1.6u22 (i.e., use of implementation `String.indexOf()` intrinsic in SSE 4.2) results in garbage collection promotion failures that lead to extended stop-the-world pauses. We highly recommend that users use JDK 1.6u20 with the directory server and not install 1.6u21 and 1.6u22. The use of JDK 1.6u20 also has an additional requirement that the Java property, `--XX:-UseSSE42Intrinsics` be set for the `start-ds` command in the directory server's Java properties file. The Java property avoids the `String.indexOf()` bug.

1. Open the `config/java.properties` file with a text editor.
2. Add the `--XX:-UseSSE42Intrinsics` flag to the `start-ds` command.

```
start-ds.java-args=-server -Xms2g -Xmx2g -XX:+UseConcMarkSweepGC
-XX:+CMSConcurrentMTEnabled -XX:+CMSParallelRemarkEnabled
-XX:+CMSParallelSurvivorRemarkEnabled -XX:ParallelCMSThreads=1
-XX:CMSMaxAbortablePrecleanTime=3600000
-XX:CMSInitiatingOccupancyFraction=80
-XX:+CMSScavengeBeforeRemark -XX:RefDiscoveryPolicy=1
-XX:+UseParNewGC -XX:+UseMembar -XX:+UseBiasedLocking
-XX:+usedCompressedOops -XX:PermSize=64M -XX:+UseNUMA
-XX:+HeapDumpOnOutOfMemoryError -XX:+PrintGCDetails
-XX:-UseSSE42Intrinsics
```

3. Save the file.
4. Run `bin/dsjavaproperties` to apply the change. If this command is not run, you will see a warning to run the command.

```
$ bin/dsjavaproperties
```

## Preparing the Operating System

You should make the following changes to your operating system depending on the production environments on which UnboundID Directory Proxy Server will run.

### Setting File Descriptor Requirements (for Linux Platforms)

UnboundID Directory Proxy Server allows for an unlimited number of connections by default, but is restricted by the file descriptor limit on the operating system. Many Linux distributions

have a default file descriptor limit of 1024 per process, which may be too low for the directory proxy server if it needs to handle a large number of concurrent connections.

To fix this condition, we recommend setting the maximum file descriptor limit per process to 65,535 on Linux systems.

For Linux kernel 2.6.27 or later, a default epoll resource limit on the amount of requested kernel memory is defined. In most cases, this epoll resource limit is too low for most directory applications (for example, 128 bytes on some systems). The epoll resource limit should be set to 65,535. See the instructions below to set this value.

**Note**

On Solaris systems (SPARC, x86, x64), run the directory proxy server as a user, or role, that contains the `sys_resource` privilege. Any process with this privilege can automatically request a higher number of file descriptors.

### To Increase the File Descriptor Limit on Linux Systems

1. Display the current kernel version of your system. Check if the kernel is 2.6.27 or later. If so, the epoll resource limit must be changed.

```
$ ulimit -r
```

2. Display the current hard limit of your system. The hard limit is the maximum server limit that can be set without tuning the kernel parameters in the `proc` filesystem.

```
$ ulimit -aH
```

3. Edit the `/etc/sysctl.conf` file. If there is a line that sets the value of the `fs.file-max` property, make sure its value is at least 65535. If there is no line that sets a value for this property, add the following line to the end of the file:

```
fs.file-max = 65535
```

4. If the Linux kernel is 2.6.27, add the following line in the `/etc/sysctl.conf` to set the epoll resource limit:

```
fs.epoll.max_user_instances = 65535
```

5. Edit the `/etc/security/limits.conf` file. If the file has lines that sets the soft and hard limits for the number of file descriptors, make sure the values are set to 65535. If the lines are not present, add the following lines to the end of the file (before “#End of file”). Also note that you should insert a tab, rather than spaces, between the columns:

```
* soft nofile 65535
* hard nofile 65535
```

6. Reboot your system, and then use the `ulimit` command to verify that the file descriptor limit is set to 65535.

```
$ ulimit -n
```

7. Check that the epoll resource limit was configured correctly.

```
$ cat /proc/sys/fs/epoll/max_user_instances
```

## Setting Filesystem Flushes (Linux)

With the out-of-the-box settings on Linux systems running the `ext3` filesystem, the data is only flushed to disk every five seconds. If the directory proxy server is running on a Linux system using the `ext3` filesystem, consider editing the mount options for that filesystem to include `commit=1` to change the flush frequency from five seconds to one second.

## Editing OS-Level Environment Variables

Having certain environment variables set can impact the directory proxy server in unexpected ways. This is particularly true for environment variables that are used by the underlying operating system to control how it uses non-default libraries.

For this reason, the directory proxy server explicitly overrides the values of key environment variables like `PATH`, `LD_LIBRARY_PATH`, and `LD_PRELOAD` to ensure that something set in the environment used to start the server does not inadvertently impact the behavior of the server.

If there is a legitimate need to edit any of these environment variables, the values of those variables should be set by manually by editing the `set_environment_vars` function of the `lib/_script-util.sh` script.

### Note

On Solaris systems, UnboundID Directory Proxy Server automatically adds, if present, `libumem.so` to the `LD_PRELOAD` (and `LD_PRELOAD_32` and `LD_PRELOAD_64`) environment variables so that the `umem` memory manager will be automatically enabled. The Directory Proxy Server also uses `pricntl`, if present, to enable the fixed priority scheduler.

## Install `systat` and `pstack` (Red Hat)

For Red Hat Linux systems, you should install a couple of packages, `systat` and `pstack`, that are disabled by default but are useful for troubleshooting purposes in the event that a problem occurs. The Directory Proxy Server's troubleshooting tool, `collect-support-data`, uses the `iostat`, `mpstat`, and `pstack` utilities to collect monitoring, statistics and stack trace information on the server's processes.

# Getting the Installation Packages

To begin the installation process, you can obtain the latest zip release bundle from UnboundID and unpack it in a folder of your choice.

By default, the bundle is unpacked in the **UnboundID-Proxy** directory.

```
$ unzip UnboundID-Proxy-3.1.0.0.zip
```

You can now install the directory proxy server.

## Installing the Directory Proxy Server

When you deploy UnboundID Directory Proxy Server in a topology, you generally deploy them in pairs. These pairs are configured identically except for their host name, port name, and possibly their location.

To help administrators easily install identical proxies, the directory proxy server allows you to clone a proxy configuration. First, you install a proxy server using the `setup` tool. Then, you configure it using the `create-initial-proxy-config` tool described in “Using the create-initial-proxy-config Tool” on page 48. Finally, you run the `setup` tool on subsequent servers, indicating that you want to clone the configuration of a peer server.

The following sections describe the `setup` tool in more detail, and tell you how to install first and subsequent directory proxy servers in your directory topology.

### About the setup Tool

One of the strengths of UnboundID Directory Proxy Server is the ease with which you can install a proxy server instance using the `setup` tool. The `setup` tool allows you to quickly install and configure a stand-alone proxy server instance.

To install the proxy server, run the `setup` tool in one of two modes: interactive command-line or non-interactive command-line.

- **Interactive command-line mode.** Interactive command-line mode prompts for information during the installation process.
- **Non-interactive command-line mode.** Non-interactive command-line mode is designed for quick command-line entry or for use in install scripts to automate directory proxy server installations. To run the installation in this mode, `setup` must be run with the `--no-prompt` option as well as all other options required to define the appropriate initial configuration.

All of the steps that are executed during the installation and configuration process should be performed while logged into the system as the user or role under which the server will run.

### Installing the First Directory Proxy Server in Interactive Mode

The `setup` tool provides an interactive text-based interface to install a proxy instance.

1. Go to the installation directory if you are not already there.

```
$ cd /UnboundID-Proxy
```

2. Use the `setup` command with the appropriate `JAVA_HOME` environment variable.

```
$ env JAVA_HOME=/ds/java ./setup
```

**Note**

If your `JAVA_HOME` environment variable is set to an older version of Java, you must explicitly specify the path to the Java 6 JDK installation during setup. You can either set the `JAVA_HOME` environment variable with the Java 6 JDK path or execute the `setup` command in a modified Java environment using the `env` command.

3. Read the UnboundID End-User License Agreement. If you agree to its terms, type **yes** to continue.
4. Type the root user DN, or press Enter or Return to accept the default (cn=Directory Manager), and then type and confirm the root user password.

```
What would you like to use as the initial root user DN for the Proxy
Server [cn=Directory Manager]:
Please provide the password to use for the initial root user:
Please re-enter the password for confirmation:
```

5. Type the port number of your proxy, or press Enter or Return to accept the default port, which is 389, if possible.

```
On which port would you like the Proxy Server to accept connections
from LDAP clients? [389]:
```

6. For SSL and StartTLS, type yes to enable one or both. Otherwise, press Enter or Return to accept the default (no).

If you enable SSL or StartTLS, the setup tool creates a directory proxy server trust store in the `config/truststore` directory. The pin is encoded in the `config/keystore.pin` file.

```
Do you want to enable SSL? (yes / no) [no]:
Do you want to enable StartTLS? (yes / no) [no]:
```

7. If you answered yes for SSL, StartTLS, or both, you will be prompted for the certificate options. If you use the Java or the PKCS#12 key store, you will be asked for the Key Store path, and the key PIN. If you use the PKCS#11 token, you will be asked for only the key PIN.

```
Do you want to enable SSL? (yes / no) [no]: yes
On which port would you like the Proxy Server to accept connections
from LDAPS clients? [1636]:
Do you want to enable StartTLS? (yes / no) [no]: yes
```

**Certificate server options:**

- 1) Generate self-signed certificate (recommended for testing

purposes only)

- 2) Use an existing certificate located on a Java Key Store (JKS)
- 3) Use an existing certificate located on a PKCS#12 key store
- 4) Use an existing certificate on a PKCS#11 token

Enter choice [1]: 2

Java Key Store (JKS) path: /path/to/keystore

Key store PIN:

8. If you want to specify a particular address on which the server listens for client connections, enter yes. Otherwise, accept the default of no.

By default the server listens on all available network interfaces for client connections. Would you like to specify particular addresses on which this server will listen for client connections? (yes / no) (no):

9. If you want to configure your directory proxy server to use entry balancing, where leaf and non-leaf entries below a common parent entry are distributed among multiple directory servers, type yes. Otherwise, accept the default of no.

Note that answering yes allows more memory to be reserved for the directory proxy server. If you do not specify entry balancing during setup, you will not see any more prompts about entry balancing for the remainder of setup and configuration.

Do you anticipate configuring this Proxy Server for entry balancing. Choosing 'yes' will allow you to specify that more memory be allocated to the server and tools? (yes / no) [no]: no

10. This option should only be selected if the directory proxy server is the primary application and no other processes consume a significant amount of memory.

Do you want to tune the JVM of this system such that the memory dedicated to the server is maximized? Choosing 'yes' will allow you to optionally specify the maximum amount of memory to be allocated to the server and tools (yes / no) [no]:

11. Enter the maximum amount of memory you want the directory proxy server to allocate to the server and tools. In this example, the maximum allowed for the server is 1 gigabytes because we are doing a proxy-only configuration without entry balancing.

Enter the maximum amount of memory to be allocated to the server and tools. The format for this value is the same as the -Xmx JVM option which is a number followed by a unit m or g. For example '2g' means 2 gigabytes. The value must be between '64m' and '16g' [16g]:

The command line provides a dynamic value range based on the resources of the system on which the installer is running. In the example above, the range is 64 megabytes to 16 gigabytes.

12. Type yes, or press Enter or Return to accept the default to start the proxy after the configuration has completed. If you plan to configure additional settings or import data, you can type no to keep the server in shutdown mode.

```
Do you want to start the server when the configuration is
completed? (yes /no) [yes]:
```

13. Confirm the configuration, and press Enter or Return to set up the proxy.

```
Setup Summary
=====
LDAP Listener Port: 389
LDAP Secure Access: disabled
Root User DN:      cn=Directory Manager

Start Server when the configuration is completed

What would you like to do?

    1) Set up the server with the parameters above
    2) Provide the setup parameters again
    3) Cancel the setup

Enter choice [1]:

Configuring Directory Proxy Server..... Done.
Starting Directory Proxy Server..... Done.

See /UnboundID-Proxy/logs/setup.log for a detailed log of this
operation.
```

14. Once setup is complete, you are prompted to begin configuration. Select whether you want to create an initial basic configuration using the **create-initial-proxy-config** tool, whether you want to configure by hand using **dsconfig**, or whether to quit and configure your proxy server later. You need to configure your proxy later if you plan to use custom schema in your deployment.

For more information about configuring your directory proxy server using this tool, see “Using the create-initial-proxy-config Tool” on page 48.

```
This server is now ready for configuration What would you like to
do?

    1) Start 'create-initial-proxy-config' to create a basic
       initial configuration (recommended for new users)
    2) Start 'dsconfig' to create a configuration from scratch
    3) Quit

Enter choice [1]:
```

## Installing the First Directory Proxy Server in Non-Interactive Mode

You can run the `setup` command in non-interactive mode to automate the installation process using a script or to run the command directly from the command line. If there is a missing or incorrect argument, the `setup` tool fails and aborts the process.

Use `setup` with the `--no-prompt` options.

```
$ env JAVA_HOME=/ds/java ./setup --no-prompt \  
--rootUserDN "cn=Directory Manager" \  
--rootUserPassword "password" --ldapPort 389 \  
--acceptLicense
```

The **setup** tool automatically chooses the maximum heap size. If you want to set the maximum amount of memory devoted to the server's process heap, include the **--maxHeapSize** option, with the appropriate value. For example, use the **--aggressiveJVMtuning** option to tune the memory used by the directory proxy server as well as the tools:

```
--aggressiveJVMtuning --maxHeapSize 256m
```

The maximum amount of memory allowed for the server is 1 gigabyte.

If you are using entry balancing, tune the amount of memory devoted to the directory proxy server using the **--entryBalancing** option:

```
--entryBalancing --maxHeapSize 1g
```

The amount of memory allowed when using **--entryBalancing** option is calculated and depends on the amount of system memory available. If you are using entry balancing and also want the tools to get more memory, include both the **--entryBalancing** and the **--aggressiveJVMtuning** options.

```
--entryBalancing --aggressiveJVMtuning --maxHeapSize 1g
```

If you have already configured a trust store, you can also use the **setup** tool to enable security. The following example enables security, both SSL and StartTLS. It also specifies a JKS key-store and truststore that define the server certificate and trusted CA. The passwords for the keystore files are defined in the corresponding **.pin** files, where the password is written on the first line of the file. The values in the **.pin** files will be copied to the **server-root/config** directory in the **keystore.pin** and **truststore.pin** files.

Note that the password to the private key within the key store is expected to be the same as the password to the key store. If this is not the case, the private key password can be defined within the Directory Proxy Management Console or **dsconfig** by editing the Trust Manager Provider standard configuration object.

```
$ env JAVA_HOME=/ds/java ./setup --cli \  
--no-prompt --rootUserDN "cn=Directory Manager" \  
--rootUserPassword "password" --ldapPort 389 \  
--enableStartTLS --ldapsPort 636 \  
--useJavaKeystore /path/to/devkeystore.jks \  
--keyStorePasswordFile /path/to/devkeystore.pin \  
--certNickName server-cert \  
--useJavaTrustStore /path/to/devtruststore.jks \  
--trustStorePasswordFile /path/to/devtruststore.pin \  
--acceptLicense
```



## Installing Subsequent Directory Proxy Servers in Interactive Mode

The `setup` tool provides an interactive text-based interface to install a proxy instance that clones a previously installed directory proxy server instance.

1. Go to the installation directory if you are not already there.

```
$ cd /UnboundID-Proxy
```

2. Use the `setup` command with the appropriate `JAVA_HOME` environment variable.

```
$ env JAVA_HOME=/ds/java ./setup
```

<b>Note</b>	If your <code>JAVA_HOME</code> environment variable is set to an older version of Java, you must explicitly specify the path to the Java 6 JDK installation during setup. You can either set the <code>JAVA_HOME</code> environment variable with the Java 6 JDK path or execute the <code>setup</code> command in a modified Java environment using the <code>env</code> command.
-------------	--

3. Read the UnboundID End-User License Agreement. If you agree to its terms, type **yes** to continue.
4. Enter yes in response to add this new server to an existing topology.

```
Would you like to add this server to an existing Directory Proxy
Server topology? (yes / no) [no]: yes
```

5. Enter the host name of the directory proxy server from which configuration settings are copied during setup.

```
Enter the host name of the peer Directory Proxy Server from which
you would like to copy configuration settings. [10.8.1.201]:
```

6. Type the port number of the peer proxy server from which configuration settings are copied during setup. You can press Enter or Return to accept the default port, which is 389, if possible.

```
Enter the port of the per Directory Proxy Server [389]:
```

7. Enter the option corresponding to the type of connection you want to use to connect to the peer Directory Proxy Server.

```
How would you like to connect to the peer Directory Proxy Server?
1) None
2) SSL
3) StartTLS
```

```
Enter choice [1]:
```

8. Type the root user DN of the peer directory proxy server, or press Enter or Return to accept the default (cn=Directory Manager), and then type and confirm the root user password.

Enter the manager account DN for the peer Directory Proxy Server [cn=Directory Manager]:

Enter the password for cn=Directory Manager:

9. Enter the host name of the new, local directory proxy server.

Enter the fully qualified host name or IP address of the local host [10.8.1.201]:

10. Choose the location of your new directory proxy server instance or enter a new one.

Choose the location for this Directory Proxy Server

- 1) east
- 2) Specify a new location

Enter choice [1]:

11. Enter the port number you want your new directory proxy server instance to use to accept client connections.

On which port would you like the Directory Proxy Server to accept connections from LDAP clients? [2389]:

12. For SSL and StartTLS, type yes to enable one or both. Otherwise, press Enter or Return to accept the default (no).

If you enable SSL or StartTLS, the setup tool creates a directory proxy server trust store in the `config/truststore` directory. The pin is encoded in the `config/keystore.pin` file.

Do you want to enable SSL? (yes / no) [no]:

Do you want to enable StartTLS? (yes / no) [no]:

13. If you answered yes for SSL, StartTLS, or both, you will be prompted for the certificate options. If you use the Java or the PKCS#12 key store, you will be asked for the Key Store path, and the key PIN. If you use the PKCS#11 token, you will be asked for only the key PIN.

Do you want to enable SSL? (yes / no) [no]: yes

On which port would you like the Proxy Server to accept connections from LDAPS clients? [1636]:

Do you want to enable StartTLS? (yes / no) [no]: yes

Certificate server options:

- 1) Generate self-signed certificate (recommended for testing purposes only)
- 2) Use an existing certificate located on a Java Key Store (JKS)
- 3) Use an existing certificate located on a PKCS#12 key store
- 4) Use an existing certificate on a PKCS#11 token

Enter choice [1]: 2

Java Key Store (JKS) path: /path/to/keystore

Key store PIN:

14. If you want to specify a particular address on which the new directory proxy server listens for client connections, enter yes. Otherwise, accept the default of no.

By default the server listens on all available network interfaces for client connections. Would you like to specify particular addresses on which this server will listen for client connections? (yes / no) (no):

15. This option should only be selected if the directory proxy server is the primary application and no other processes consume a significant amount of memory.

Do you want to tune the JVM of this system such that the memory dedicated to the server is maximized? Choosing 'yes' will allow you to optionally specify the maximum amount of memory to be allocated to the server and tools (yes / no) [no]:

16. Enter the maximum amount of memory you want the directory proxy server to allocate to the server and tools. In this example, the maximum allowed for the server is 1 gigabytes because we are doing a proxy-only configuration without entry balancing.

Enter the maximum amount of memory to be allocated to the server and tools. The format for this value is the same as the -Xmx JVM option which is a number followed by a unit m or g. For example '2g' means 2 gigabytes. The value must be between '64m' and '16g' [16g]:

The command line provides a dynamic value range based on the resources of the system on which the installer is running. In the example above, the range is 64 megabytes to 16 gigabytes.

17. Type yes, or press Enter or Return to accept the default to start the proxy after the configuration has completed. If you plan to configure additional settings or import data, you can type no to keep the server in shutdown mode.

Do you want to start the server when the configuration is completed? (yes /no) [yes]:

18. Confirm the configuration, and press Enter or Return to set up the proxy.

Setup Summary

=====

LDAP Listener Port: 2389

LDAP Secure Access: disabled

Root User DN: cn=Directory Manager

Start Server when the configuration is completed

- 1) Set up the server with the parameters above
- 2) Provide the setup parameters again
- 3) Cancel the setup

Enter choice [1]:

Applying Peer Configuration..... Done.

Updating Administration data..... Done.

Configuring Directory Proxy Server..... Done.

```
Updating Topology Admin Data..... Done.  
Starting Directory Proxy Server..... Done.
```

To see basic server configuration status and configuration you can launch `/UnboundID-Proxy/status`.

See `/UnboundID-Proxy/logs/setup.log` for a detailed log of this operation.

## Installing Subsequent Directory Proxy Servers in Non-Interactive Mode

You can run the `setup` command in non-interactive mode to automate the installation process of a cloned directory proxy server using a script or to run the command directly from the command line. If there is a missing or incorrect argument, the `setup` tool fails and aborts the process.

Use `setup` with the `--no-prompt` options.

```
$ env JAVA_HOME=/ds/java ./setup --cli --no-prompt \  
--rootUserDN "cn=Directory Manager" \  
--rootUserPassword "password" --ldapPort 1389 \  
--localHostName 10.8.1.221 \  
--peerHostName 10.8.1.201 --peerPort 389 \  
--peerUseNoSecurity
```

## Running the Directory Proxy Server

To start the proxy server, run the `bin/start-proxy` command on UNIX® or Linux systems. Run the `bat/start-proxy` command on Windows systems. The `start-proxy` command starts the proxy server as a background process when no options are specified. To run the proxy server as a foreground process, use the `start-proxy` command with the `--nodetach` option.

### To Start the Directory Proxy Server as a Background Process

Go to the installation directory, and then use `start-proxy`.

```
$ bin/start-proxy
```

### To Start the Proxy Server as a Foreground Process

1. Type `start-proxy` to launch the proxy server as a foreground process.

```
$ bin/start-proxy --nodetach
```

2. You can stop the proxy server by pressing `Ctrl+C` in the terminal window where the server is running or by running the `stop-proxy` utility from another window.

## To Start the Directory Proxy Server at Boot

By default, the directory proxy server does not start automatically when the system is booted. Instead, you must manually start it with the `bin/start-proxy` command. To configure the directory proxy server to start automatically when the system boots, use the `create-rc-script` tool to create a run control (RC) script as follows:

1. Create the startup script.

```
$ bin/create-rc-script --outputFile UnboundID-PS.sh --userName ds
```

2. As root, move the generated `UnboundID-PS.sh` script into the `/etc/init.d` directory, and create symlinks to it from the `/etc/rc3.d` (starting with an "S" to ensure that the server is started) and `/etc/rc0.d` directory (starting with a "K" to ensure that the server is stopped).

```
# mv UnboundID-PS.sh /etc/init.d/
# ln -s /etc/init.d/UnboundID-PS.sh /etc/rc3.d/S50-UnboundID-PS.sh
# ln -s /etc/init.d/UnboundID-PS.sh /etc/rc0.d/K50-UnboundID-PS.sh
```

3. Log out as root, and re-assume the `ds` role if you are on a Solaris system.

## Stopping the Directory Proxy Server

Change to the installation directory and use `stop-proxy`.

```
$ bin/stop-proxy
```

## Scheduling a Directory Proxy Server Shutdown

Use the `stop-proxy` tool with the `--stopTime YYYYMMDDhhmmss` option to schedule a server shutdown. The directory proxy server schedules the shutdown and sends a notification to the `server.out` log. The following example sets up a shutdown task that is scheduled to be processed on October 24, 2008 at 3:45 p.m. local time. The server uses UTC time if the provided timestamp includes a trailing "Z", for example, `20081024154500Z`.

```
$ bin/stop-proxy --port 389 \
  --bindDN "cn=Directory Manager" \
  --bindPassword secret --stopTime 20081024154500
```

## Restarting the Directory Proxy Server

You can restart the directory proxy server using the `stop-proxy` command with the `--restart` or `-R` option. Running this command is equivalent to shutting down the server,

exiting the JVM session, and then starting up again, which requires a re-priming of the JVM cache. To avoid destroying and re-creating the JVM, use an internal restart, which can be issued over LDAP. The internal restart will keep the same Java process and avoid any changes to the JVM options.

Go to the installation directory. Using a loopback interface, run the `stop-proxy` command with the `-R` or `--restart` options.

```
$ bin/stop-proxy --restart \  
  --hostname 127.0.0.1 --port 389 \  
  --bindDN "cn=Directory Manager" --bindPassword secret
```

## Uninstalling the Directory Proxy Server

The proxy server provides an `uninstall` command-line utility for quick and easy removal of the code base. You can uninstall the proxy server using one of the following modes:

- **Interactive command-line mode.** This mode is a text-based interface. The utility prompts you for input if more data is required.
- **Non-interactive command-line mode.** This mode suppresses progress information from being provide in standard output during processing, except for fatal errors. This mode is convenient for scripting and is invoked with the `--no-prompt` option.

### Uninstalling the Directory Proxy Server in Interactive Command-Line Mode

Interactive mode uses a text-based, command-line interface to help you remove your proxy instance. If `uninstall` cannot remove all of the proxy files, the server generates a message with a list of the files and directories that must be manually deleted. The `uninstall` command must be run as either the root user or the same user (or role) that installed the proxy.

1. Go to the installation directory.

```
$ cd /UnboundID-Proxy
```

2. Use the `uninstall` command.

```
$ ./uninstall
```

3. Select the components to be removed. If you want to remove all components, press Enter or Return to accept the default.

```
Do you want to remove all components or select the components to  
remove?
```

```
1) Remove all components  
2) Select the components to be removed  
  
q) quit
```

Enter choice [1]:

4. If your proxy is running, press Enter or Return to shutdown the server before continuing the uninstall process.

The server is currently running and must be stopped before  
uninstallation can continue.

Stop the Server and permanently delete the files? (yes / no) [yes]:

5. Complete the uninstall, and view the logs for any remaining files. Manually remove any remaining files or directories, if required.

## Uninstalling the Directory Proxy Server in Non-Interactive Mode

The `uninstall` utility provides a `--no-prompt` option that you can enter on the command line or use in a script. Another useful argument is the `--forceOnError` option that continues the uninstall process when an error is encountered. If an option is incorrectly entered or if a required option is omitted and the `--forceOnError` option is not used, the command will fail and abort.

### To Uninstall the Directory Proxy Server in Non-Interactive Mode

1. Go to the installation directory.

```
$ cd /UnboundID-Proxy
```

2. Use `uninstall` with the `--remove-all` option to remove all of the proxy's libraries. The `--quiet` option suppresses output information and is optional.

```
$ ./uninstall --remove-all --no-prompt --quiet \  
--forceOnError
```

3. If any files or directories remain, manually remove them.

### To Uninstall Selected Components in Non-Interactive Mode

1. Go to the installation directory.

```
$ cd /UnboundID-Proxy
```

2. Use `uninstall` with the `--backup-files` option to remove the proxy's backup files. Use the `--help` or `-H` option to view the other options available to remove specific components.

```
$ ./uninstall \  
--backup-files \  
--no-prompt \  
--quiet \  
--forceOnError
```

## Updating the Directory Proxy Server

UnboundID issues new software builds periodically and distributes the software package in zip format. Administrators can use the directory proxy server's **update** utility to update the current server code with the latest features and bug fixes. To update the directory proxy server to a newer version, download the build package, and then unzip the new server package on the same host as the server that you wish to update. Before upgrading a server, you should ensure that it is capable of starting without severe or fatal errors.

During an update process, the updater checks a manifest file that contains a MD5 checksum of each file in its original state when installed from zip. Next, it compares the checksum of the new server files to that of the old server. Any files that have different checksums will be updated. For files that predates the manifest file generation, the file is backed up and replaced. The updater also logs all file changes in the history directory to tell what files have been changed.

For schema updates, the **update** tool preserves any custom schema definitions (**99-user.ldif**). For any default schema element changes, if any, the updater will warn the user about this condition and then create a patch schema file and copy it into the server's schema directory. For configuration files, the **update** tool preserves the configuration file, **config.ldif**, unless new configuration options must be added to the directory proxy server.

Once the updater finishes its processing, it checks if the newly updated server starts without any fatal errors. If an error occurs during the update process, the **update** tool reverts the server root instance to the server state prior to the update.

### To Update the Directory Proxy Server

Assume that an existing version of the directory proxy server is stored at **/ds/UnboundID-Proxy-old**, which you want to update.

1. Make sure you have complete, readable backup of the existing system before upgrading the directory proxy server build. Also, make sure you have a clear backout plan and schedule.
2. Download the latest version of the UnboundID Directory Proxy Server software and unzip the file. You can use the **unzip -f/--file** option to specify a location on the server. For this example, let's assume the new server is located in the **/ds/UnboundID-Proxy-new** directory.
3. Check the version number of the newly downloaded directory proxy server instance using the **--version** option on any command-line utility. For example, you should see the latest revision number.

```
$ /ds/UnboundID-Proxy/setup --version
UnboundID Directory Proxy Server 3.1.0.0
Build 20100416162859Z
Revision 5905
```



4. Use the **update** tool of the newly unzipped build to update the directory proxy server code. Make sure to specify the directory proxy server instance that you are upgrading with the **--serverRoot** option.

```
$ /ds/UnboundID-Proxy-new/update --serverRoot /ds/UnboundID-Proxy-old
```

Note	The UnboundID Directory Proxy Server provides a web console, the UnboundID Directory Proxy Management Console, to configure and monitor the server. If you update the directory proxy server version, you should also update the Directory Proxy Management Console.
------	--

5. View the log file to see which files were changed. The log file is located in the **<server-root>/history** directory. For example, the file will be labelled with the directory proxy server version number and revision.

```
$ view <server-root>/history/1272307020420-3.1.0.0.6011/update.log
```

## Reverting an Update

Once the directory proxy server has been updated, you can revert to the most recent version (one level back) using the **revert-update** tool. The **revert-update** tool accesses a log of file actions taken by the updater to put the filesystem back to its prior state. If you have run multiple updates, you can run the tool multiple times to revert to each prior update sequentially. You can only revert back one level. For example, if you have run the update twice since first installing the directory proxy server, you can run the update command to revert to its previous state, then run the update command again to return to its original state.

### To Revert to the Most Recent Server Version

Use **revert-update** in the server root directory revert back to the most recent version of the server.

```
$ /ds/UnboundID-Proxy-new/revert-update
```

# Installing the Directory Proxy Management Console

UnboundID Directory Proxy Server provides a graphical web application tool, Directory Proxy Management Console, which is an interface to the proxy server's configuration. The

web console provides functionality equivalent to the `dsconfig` tool in addition to monitoring and server information.

**Note**

Like the `dsconfig` tool, all changes made using the web console are recorded in `logs/config-audit.log`.

## Installing the Console

UnboundID provides a zip file containing the war files necessary to install and configure the Directory Proxy Management Console.

### To Install the Console

1. Download and install Apache Tomcat version 6 or later. For example, download **apache-tomcat-`<version>`.zip** from <http://tomcat.apache.org/>, and then unzip this file in a location of your choice.
2. Download the web console zip file, **UnboundID-DS-web-console-`<version>`.zip** and unzip the file on your server. You should see the following files:

**3RD-PARTY-LICENSE.TXT**

**LICENSE.TXT**

**README**

**proxyconsole.war**

3. If the directory proxy server that you want to manage is listening for LDAP connections on port 389 on the same machine as the console, then copy **proxyconsole.war** to **apache-tomcat-`<version>`/webapps** and go to step 7.
4. Create a **proxyconsole** directory in **apache-tomcat-`<version>`/webapps**. Then, copy the **proxyconsole.war** file to **apache-tomcat-`<version>`/webapps/proxyconsole**.

```
$ mkdir /apache-tomcat-<version>/webapps/proxyconsole
```

```
$ cp proxyconsole.war /apache-tomcat-<version>/webapps/proxyconsole
```

5. Go to the **apache-tomcat-`<version>`/webapps/proxyconsole** directory to extract the contents of the console. The jar command is included with the JDK.

```
$ cd /apache-tomcat-<version>/webapps/proxyconsole
```

```
$ jar xvf proxyconsole.war
```

6. Edit the **proxyconsole/WEB-INF/web.xml** file to point to the correct directory instance. Change the host and port to match your directory. For example, if your system's host name is **server1.example.com** and the LDAP listener port is **1389**, change those lines in the

<servlet> section. The defaults are **localhost** for the host name and **389** for the LDAP port.

```
<context-param>
  <param-name>host</param-name>
  <param-value>localhost</param-value>
</context-param>

<context-param>
  <param-name>port</param-name>
  <param-value>1389</param-value>
</context-param>

<context-param>
  <param-name>com.sun.webui.theme.DEFAULT_THEME</param-name>
  <param-value>suntheme</param-value>
</context-param>
```

**Note**

When managing multiple instances in a replicated environment, the console automatically discovers other instances in the topology by reading the topology information stored under the **cn=admin data** entry.

7. With the out-of-the-box configuration, Tomcat will time out sessions after 30 minutes of inactivity forcing the user to log back in again. This can be changed on an application server wide basis by editing **apache-tomcat-<version>/conf/web.xml**, and updating the value of this configuration parameter:

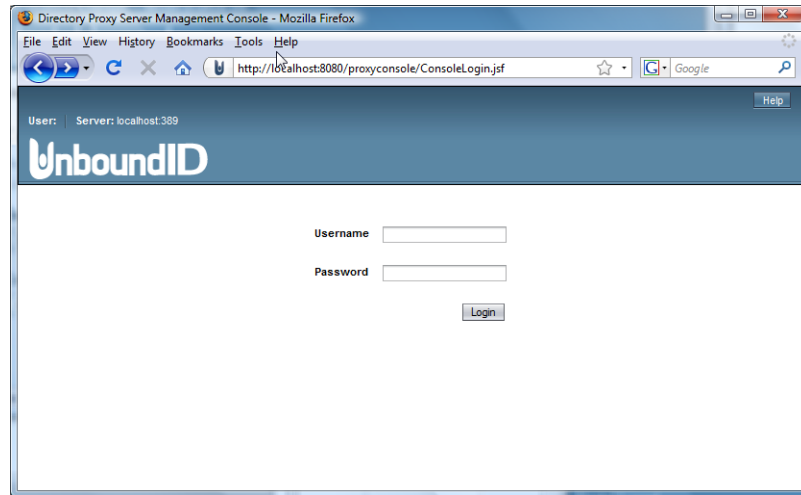
```
<session-config>
  <session-timeout>120</session-timeout>
</session-config>
```

Tomcat will expire sessions after the specified number of minutes. Changing the value to 120, for example, will extend the expiration to two hours. Changes to this setting might not take effect until the Tomcat instance is restarted, so consider changing the value before starting Tomcat for the first time.

8. Start the directory proxy server if it is not already running, and then start the web console using the **apache-tomcat-<version>/bin/startup.sh** script. Use **shutdown.sh** to stop the application server. (On Windows, use **startup.bat** and **shutdown.bat**.) Note that the **JAVA\_HOME** environment variable must be set to specify the location of the Java installation to run the server.

```
$ env JAVA_HOME=/ds/java bin/startup.sh
Using CATALINA_BASE:   /apache-tomcat-6.0.18
Using CATALINA_HOME:   /apache-tomcat-6.0.18
Using CATALINA_TMPDIR: /apache-tomcat-6.0.18/temp
Using JRE_HOME:        /ds/java
```

9. Open a browser to **http://hostname:8080/proxyconsole**. The application server listens to port 8080 for HTTP requests.



## Working With the Directory Proxy Management Console

The Directory Proxy Management Console does not persistently store any credentials for authenticating to the directory proxy server but uses the credentials provided by the user when logging in. When managing multiple server instances, the provided credentials must be valid for each instance. In a replicated environment, admin users created using the `dsframework` command are automatically replicated to each instance.

### Logging into the Directory Proxy Management Console

To log into the console, you can either use a DN (for example, `cn=Directory Manager`) or provide the name of an administrator, which is stored under `cn=admin data`. The `dsframework` command can be used to create a global administrator, for example:

```
$ dsframework create-admin-user \  
--hostname server1.example.com \  
--port 1389 --bindDN "cn=Directory Manager" \  
--bindPassword secret \  
--userID someAdmin --set password:secret
```

### To Log into the Console

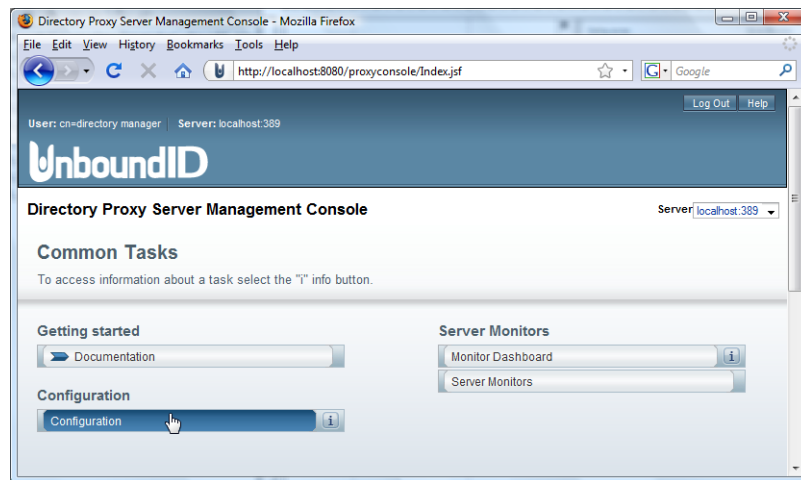
1. Go to the installation directory.  

```
$ cd /ds/UnboundID-Proxy
```

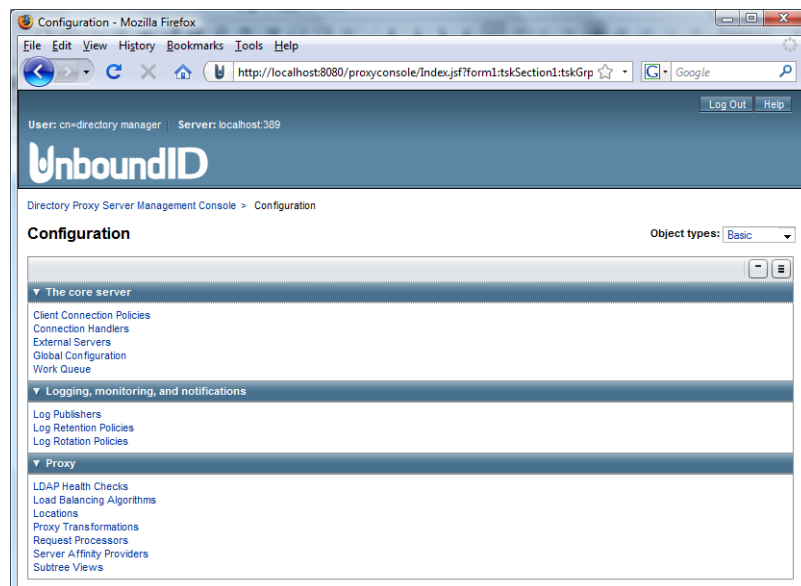
2. Start the directory proxy server.  

```
$ bin/start-proxy
```
3. Start the Apache Tomcat application server.  

```
$ /apache-tomcat-<version>/bin/startup.sh
```
4. Open a browser to `http://hostname:8080/proxyconsole/`.
5. Type the root user DN (or any authorized administrator user name) and password, and then click **Login**.
6. On the Directory Proxy Management Console, click **Configuration**.



7. View the Configuration menu. By default, the console displays the Basic object type properties. You can change the complexity level of the object types using the **Object Types** drop-down list.



### To Update the Console

1. Shut down the console and application server.
2. In the current deployment of the web console, move the `<webapps>/proxyconsole/WEB-INF/web.xml` file to another location.
3. Download and deploy the latest version for the web console. Follow steps 2–5 outlined in the section “To Install the Console” on page 42.
4. Assuming you had not renamed the `.war` file when you originally deployed the web console, replace the newly deployed web console’s `web.xml` to `<webapps>/proxyconsole/WEB-INF/web.xml` file.
5. Start the application server.

### To Uninstall the Console

1. Close the web console, and shut down the application server. (On Windows, use `shutdown.bat`).  
  

```
$ /apache-tomcat-<version>/bin/shutdown.sh
```
2. Remove the `webapps/proxyconsole` directory.  
  

```
$ rm -rf webapps/proxyconsole
```
3. Restart the Tomcat instance if necessary. Alternatively, if no other applications are installed in the Tomcat instance, then the entire Tomcat installation can be removed by deleting the `apache-tomcat` directory tree.

---

# 3

## Configuring the Directory Proxy Server

Once you have initially configured the UnboundID® Directory Proxy Server, you can manage your deployment using the configuration framework and management tools. This chapter briefly describes these tools and provides procedures to help you maintain and update your deployment.

It includes the following sections:

- Overview of the Configuration and Management Tools
- Using the `create-initial-proxy-config` Tool
- Using `dsconfig` to Configure the Directory Proxy Server
- Configuring Global Administrator Accounts
- Configuring Administrative Server Groups
- Configuring Root User Accounts
- Configuring Server Health Checks
- Configuring LDAP External Servers
- Configuring Proxy Transformations
- Configuring Request Processors
- Configuring Server Affinity
- Configuring Subtree Views
- Configuring Client Connection Policies
- Configuring Entry Balancing

### Overview of the Configuration and Management Tools

UnboundID Directory Proxy Server provides the following configuration management tools to simplify the configuration and management of proxy server instances and your deployment.

- The `create-initial-proxy-config` tool. This command-line tool can be used to initially configure the directory proxy server. We strongly recommend that you use the `create-initial-proxy-config` tool for your initial directory proxy server configuration. This tool prompts you for basic information about your topology, including external servers, their locations, and credentials for communicating with them. Once configuration

is complete, the tool writes the configuration in a `dsconfig` batch file and allows you to apply the configuration to the local directory proxy server.

- The `dsconfig` command-line tool. You can run the `dsconfig` command in non-interactive mode to automate the configuration process using a script or to run the command without prompting from the command line. You can also run it using the interactive text-based interface to configure a proxy server instance.
- The UnboundID® Directory Proxy Server Management Console. This graphical web interface can be used to access the Directory Proxy Server configuration information. The web console provides functionality equivalent to the `dsconfig` tool in addition to monitoring and server information.
- The `prepare-external-server` command-line tool. This tool can be used to configure communication between the UnboundID Directory Proxy Server and the UnboundID® Directory Server. This tool can be used in conjunction with `create-initial-proxy-config` or `dsconfig` to simplify configuring your directory proxy server deployment. For more information about using this tool, see “To Configure Server Communications Using the prepare-external-server Tool” on page 79.

## Using the create-initial-proxy-config Tool

The `create-initial-proxy-config` tool helps you to initially configure the local directory proxy server. You are prompted to launch this tool after installing the directory proxy server. The tool assumes the following about your topology:

- All servers are accessible through a single user account. This user account must be a root user that is not generally accessible to clients to avoid inadvertent changes, deletions, or backend server availability issues due to reimporting data.
- All servers support the same type of communication security.
- All external servers are any combination of UnboundID Directory Server, Alcatel-Lucent 8661 Directory Server, Sun™ Directory Server, or Red Hat (including Fedora and 389) instances.

If your topology does have these characteristics, you can use the tool to define a basic configuration that is saved to a `dsconfig` batch file. You can then run the `dsconfig` tool to fine-tune the configuration. You can also use this tool to configure an entry balancing configuration, which allows you to automatically spread entries below a common parent among multiple sets of directory servers for improved scalability and performance.

The `create-initial-proxy-config` tool produces a log file called `create-initial-proxy-config.log` that is stored in the local directory proxy server’s `logs` directory.



You can only run the **create-initial-proxy-config** tool once for the initial configuration of each directory proxy server instance. To tune your configuration, use the **dsconfig** tool.

This section describes how to use this tool to configure a standard directory proxy server deployment as well as an entry balancing configuration.

### To Configure a Standard Directory Proxy Server Deployment Using the create-initial-proxy-config Tool

1. After initial installation, select 1 to start the **create-initial-proxy-config** tool automatically. Otherwise, run it manually at the command line from the installation directory, **UnboundID-Proxy**.

```
$ ./bin/create-initial-proxy-config
```

2. Press return to continue with configuration.

```
Would you like to continue? (yes / no) [yes]:
```

3. Enter the DN for the directory proxy user account, then enter and confirm the password for this account.

```
Enter the DN of the proxy user account [cn=Proxy User,cn=Root DNs,cn=config]:
```

```
Enter the password for 'cn=Proxy User,cn=Root DNs,cn=config':
```

```
Confirm the password for 'cn=Proxy User,cn=Root DNs,cn=config':
```

Note that you should not use **cn=Directory Manager** as the account to use for communication between the directory proxy server and the directory server. For security reasons, the account used to communicate between the directory proxy server and the directory server should not be directly accessible by clients accessing the directory proxy server. For more information about this account, see “Configuring LDAP External Servers” on page 76.

4. Specify whether you will be using secure communication with the directory server instances.

```
1) None
2) SSL
3) StartTLS
```

```
b) back
q) quit
```

```
Enter choice [1]:
```

Note that if you choose to use SSL or StartTLS, the default location of the directory proxy server trust store is **config/truststore**. The pin is encoded in the **config/key-store.pin** file.

5. Specify the base DN of the proxy. You can specify more than one. Press enter when you have finished specifying DNs.

Enter a DN or choose a menu item [dc=example,dc=com] :

6. Because you are configuring a standard deployment of directory proxy server, answer **no** to the question about splitting entries across multiple servers.

7. Define a location for your server. You can specify multiple locations. This example illustrates defining a location named **east**.

Enter a location name or choose a menu item: **east**

8. If you defined more than one location, specify the location that contains the directory proxy server itself.

Choose the location for this Directory Proxy Server

1) **east**

b) **back**

q) **quit**

Enter choice [1]: **1**

9. Define the host, port, and authentication information used by the LDAP external servers. If you have specified more than one location, you will go through this process for each location.

Enter a host:port or choose a menu item [localhost:389]: **ldap-east-01.example.com:389**

10. Select 3 to indicate that you want the tool to create a proxy user account on all of your LDAP external servers.

Would you like to prepare **ldap-east-01.example.com:389** for access by the Proxy Server?

1) **Yes**

2) **No**

3) **Yes, and all subsequent servers**

4) **No, and all subsequent servers**

Enter choice [1]: **3**

Testing connection to **ldap-east-01.example.com:389**..... **Done**

Testing 'cn=Proxy User' access to **ldap-east-01.example.com:389**.....

Failed to bind as 'cn=Proxy User'

11. If the proxy user account did not previously exist on your LDAP external server, create the account by connecting as cn=Directory Manager.

Would you like to create or modify root user 'cn=Proxy User' so that it is available for this Proxy Server? (yes / no) [**yes**]:

Enter the DN of an account on **ldap-east-01.example.com:389** with which to create or manage the 'cn=Proxy User' account [**cn=Directory Manager**]:

```
Enter the password for 'cn=Directory Manager':
```

```
Created 'cn=Proxy User,cn=Root DNs,cn=config'  
Testing 'cn=Proxy User' privileges ..... Done  
Verifying backend 'dc=example,dc=com' ..... Done
```

12. Press enter to finish configuring the location.

```
Enter a host:port or choose a menu item [Press ENTER when finished  
entering servers for 'east']:
```

13. Review the configuration summary. Once you have confirmed that the changes are correct, press enter to write the configuration.

```
b) back  
q) quit  
w) write configuration
```

```
Enter choice [w]:
```

14. Press enter to apply the changes to the directory proxy server. Alternatively, you can quit and instead run the **dsconfig** batch file at a later time.

Once the changes have been applied, you cannot use the **create-initial-proxy-config** tool to configure this directory proxy server again. Instead, use the **dsconfig** tool.

```
Apply these configuration changes to the local Directory Proxy  
Server? (yes /no) [yes]:
```

## To Configure an Entry Balancing Directory Proxy Server Deployment Using the create-initial-proxy-config Tool

1. Follow steps 1-5 of the previous procedure.
2. Because you are configuring an entry balancing deployment of directory proxy server, answer **yes** to the question about splitting entries across multiple servers.
3. Enter the number of backend server sets across which the data will be split. In this example, we enter 2 to specify that the data will be balanced across two sets of servers.

```
Across how many sets of servers is the data balanced?
```

```
c) cancel creating dc=example,dc=com  
q) quit
```

```
Enter a number greater than one or choose a menu item: 2
```

4. Specify the base DN below which the data is balanced. In this example, data below the **ou=people,dc=example,dc=com** entry are balanced across the two server sets.

```
Enter the entry balancing base DN or choose a menu item  
[dc=example,dc=com]: ou=people,dc=example,dc=com
```

5. Define index attributes for frequently queried attributes that can be used by the global index. In this example, we add the `telephoneNumber` attribute to the global index. We also indicate that we want the attribute to be loaded into memory before the directory proxy server starts accepting connections.

```
Would you like to add attributes to the global index? (yes / no)
[no]:
```

```
...
```

```
Enter an attribute name or choose a menu item [Press ENTER when
finished entering index attributes]: telephoneNumber
```

```
Should the index for attribute 'telephoneNumber' be primed such
that it is loaded into memory before the Directory Proxy Server
begins accepting connections?
```

- 1) Yes
- 2) No
- 3) Yes, and all subsequent attributes
- 4) No, and all subsequent attributes

```
Enter choice [1]: 1
```

6. Indicate whether you want to prime the RDN index at startup. Though index priming increases startup time and the load on the backend servers, it leads to better initial proxy performance.

```
Would you like to enable RDN index priming for
'dc=example,dc=com'?
(yes / no) [yes]:
```

7. Enter a base DN of a new subtree view, if required.

```
Enter a DN or choose a menu item [Press ENTER when finished
entering base DN]:
```

8. Define a location of your server. You can specify multiple locations. This example illustrates defining a location named `east`.

```
Enter a location name or choose a menu item: east
```

9. Specify the location that contains the directory proxy server itself.

```
Choose the location for this Directory Proxy Server
```

```
1) east
```

```
b) back
```

```
q) quit
```

```
Enter choice [1]: 1
```

10. Define the host, port, and authentication information used by the LDAP external servers.

```
Enter a host:port or choose a menu item [localhost:389]: \
ldap-east-01.example.com:389
```

11. Select the set of sets of data for which the server will handle requests.

Assign server localhost:389 to handle requests for one or more of the defined sets of data:

- 1) dc=example,dc=com; Server Set 1
- 2) ou=people,dc=example,dc=com; Server Set 2

Enter one or more choices separated by commas:

12. Select 3 to indicate that you want the tool to create a proxy user account on all of your LDAP external servers.

Would you like to prepare ldap-east-01.example.com:389 for access by the Proxy Server?

- 1) Yes
- 2) No
- 3) Yes, and all subsequent servers
- 4) No, and all subsequent servers

Enter choice [1]: 3

Testing connection to ldap-east-01.example.com:389..... Done  
Testing 'cn=Proxy User' access to ldap-east-01.example.com:389.....  
Failed to bind as 'cn=Proxy User'

13. Specify the backend set to which this particular LDAP external server belongs. In this example, the server is part of server set 1.

Assign server ldap-dataset-1-01.example.com:389 to handle requests for one or more of the defined sets of data

- 1) dc=example,dc=com; Server Set 1
- 2) dc=example,dc=com; Server Set 2

Enter one or more choices separated by commas: 1

14. Define servers for your other server sets. Enter return when you have finished.

15. Press enter to complete configuring the location.

16. Review the configuration summary. Once you have confirmed that the changes are correct, press enter to write the configuration.

- b) back
- q) quit
- w) write configuration

Enter choice [w]:

17. Press enter to apply the changes to the directory proxy server. Alternatively, you can quit and instead run the **dsconfig** batch file at a later time.

Once the changes have been applied, you cannot use the `create-initial-proxy-config` tool to configure this directory proxy server again. Instead, use the `dsconfig` tool.

Apply these configuration changes to the local Directory Proxy Server? (yes /no) [yes]:

## Using dsconfig to Configure the Directory Proxy Server

The `dsconfig` tool is the text-based management script used to configure the underlying directory configuration. The tool has three operational modes: interactive mode, non-interactive mode, and batch mode. The tool also offers an enhanced user interface that displays the most common configuration options using object complexity levels. By selecting a complexity level, you can see additional menu configuration properties that are normally hidden for the basic user. The tool also has toggling mechanisms to hide or show advanced properties for specific configuration objects.

### Using dsconfig in Interactive Command-Line Mode

Running `dsconfig` in interactive command-line mode provides a user-friendly, menu-driven interface for accessing and configuring the UnboundID Directory Proxy Server. To start `dsconfig` in interactive command-line mode, simply invoke the `dsconfig` script without any arguments. You will be prompted for information about how to connect and authenticate to the Directory Server, and then a menu will be displayed of the available operation types.

In some cases, a default value will be provided in square brackets. For example, `[389]` indicates that the default value for that field is port 389. You can press Enter or Return to accept the default. To skip the connection and authentication prompts, provide this information using the command-line options of `dsconfig`.

To configure the directory proxy server using `dsconfig` in interactive command-line mode, go to the installation directory and launch the `dsconfig` tool as follows:

```
$ bin/dsconfig
```

### Working with dsconfig Object Complexity Menus

Because some configuration objects are more likely to be modified than others, the UnboundID Directory Proxy Server provides four different complexity menus that hide or expose configuration objects to the user. The following complexity menus are available:

- **Basic.** Only includes the components that are expected to be configured most frequently.
- **Standard.** Includes all components in the Basic menu plus other components that might occasionally need to be altered in many environments.

- **Advanced.** Includes all components in the Basic and Standard menus plus other components that might require configuration under special circumstances or that might be potentially harmful if configured incorrectly.
- **Expert.** Includes all components in the Basic, Standard, and Advanced menus plus other components that should almost never require configuration or that could seriously impact the functionality of the server if configured incorrectly.

The purpose of complexity types is to present only those properties that an administrator will likely use. The Object Complexity type is a convenience feature designed to make the menu more readable.

## To Change the dsconfig Object Complexity Level

1. Use the dsconfig tool to configure the LDAP external server locations.

```
$ bin/dsconfig
```

2. Type the hostname or IP address for your directory proxy server, or press enter to accept the default, localhost.

```
Directory Proxy Server hostname or IP address [localhost]:
```

3. Type the number corresponding how you want to connect to the proxy server, or press Enter to accept the default, LDAP.

```
How do you want to connect?
```

- 1) LDAP
- 2) LDAP with SSL
- 3) LDAP with StartTLS

4. Type the port number for your directory proxy server, or press Enter to accept the default, 389.

```
Directory Proxy Server port number [389]:
```

5. Type the administrator's bind DN or press Enter to accept the default (cn=Directory Manager), and then type the password.

```
Administrator user bind DN [cn=Directory Manager]:  
Password for user 'cn=Directory Manager':
```

6. In the Directory Proxy Server configuration console menu, type o (letter “o”) to change the complexity level. By default, basic objects are displayed.

```
>>>> >>>> UnboundID Directory Proxy Server configuration con-  
sole main menu
```

```
What do you want to configure?
```

- |                             |                          |
|-----------------------------|--------------------------|
| 1) Client Connection Policy | 9) Log Publisher         |
| 2) Connection Handler       | 10) Log Retention Policy |
| 3) External Server          | 11) Log Rotation Policy  |

```
4) Global Attribute Index      12) Placement Algorithm
5) Global Configuration        13) Proxy Transformation
6) LDAP Health Check          14) Request Processor
7) Load-Balancing Algorithm    15) Subtree View
8) Location                    16) Work Queue

o) 'Basic' objects are shown - change this
q) quit

Enter choice: o
```

7. Type a number corresponding to a complexity level of your choice.

```
>>>> Currently 'Basic' objects are shown.  Change this globally as follows

1) Basic - show only the most commonly accessed objects
2) Standard - show objects intended for most users
3) Advanced - show all objects except a few intended for experts-only
4) Expert - show all objects

?) help

Enter choice [Basic]: 2
```

## Using dsconfig in Non-Interactive Command-Line Mode

The `dsconfig` non-interactive command-line mode provides a simple way to make arbitrary changes to the directory proxy server by invoking it on the command line. If you want to use administrative scripts to automate the process of making configuration changes, then running `dsconfig` command in non-interactive mode is more convenient. Note, however, that if you plan to make changes to multiple configuration objects at the same time, then the batch mode might be more appropriate.

You can use the `dsconfig` tool to update a single configuration object using command-line arguments to provide all of the necessary information. The general format for the non-interactive command line is:

```
$ bin/dsconfig --no-prompt {subcommand} {globalArgs} {subcommandArgs}
```

The `--no-prompt` argument indicates that you want to use non-interactive mode. The `{subcommand}` is used to indicate which general action to perform. The `{globalArgs}` argument provides a set of arguments that specify how to connect and authenticate to the directory proxy server, and the `{subcommandArgs}` argument contains a set of arguments specific to the particular subcommand that you wish to invoke.

### Note

Global arguments can appear anywhere on the command line (including before the subcommand, and after or intermingled with subcommand-specific arguments). The subcommand-specific arguments can appear anywhere after the subcommand.



For example, use the `dsconfig` command in non-interactive mode to create a new location called East.

```
$ bin/dsconfig --no-prompt \  
  create-location \  
  --bindDN "cn=DirectoryManager" \  
  --bindPassword password \  
  --location-name East\  

```

## Getting the Equivalent dsconfig Non-Interactive Mode Command

While the `dsconfig` non-interactive command-line mode is convenient for scripting and automating processes, obtaining the correct arguments and properties for each configuration change can be quite time consuming.

To facilitate easy and quick configuration, you can use an option to display the equivalent non-interactive command using `dsconfig` interactive mode. The command displays the equivalent `dsconfig` command to recreate the configuration in a scripted configuration or to more quickly enter any pending changes on the command line for another server instance.

### Note

There are two other ways to get the equivalent `dsconfig` command. One way is by looking at the `logs/config-audit.log`. It might be more convenient to set the directory proxy server up the way you want and then get the `dsconfig` arguments from the log. Another way is by configuring an option using the Directory Proxy Server Management Console. The console shows the equivalent `dsconfig` command prior to applying the change.

## To Get the Equivalent dsconfig Non-Interactive Mode Command from Interactive Mode

1. Using `dsconfig` in interactive mode, make changes to a configuration but do not apply the changes (by typing **£**). The previous example makes changes to the **db-cache-percent**.
2. Type **d** to view the equivalent non-interactive command.
3. View the equivalent command (highlighted in bold), and then press **Return** to continue.

## Using dsconfig in Batch Mode

The UnboundID Directory Proxy Server provides a `dsconfig` batch mechanism that reads multiple `dsconfig` invocations from a file and executes them sequentially. The batch file provides advantages over standard scripting by minimizing LDAP connections and JVM invocations that normally occur with each `dsconfig` call. If a `dsconfig` command has a missing or incorrect argument, the command will fail and abort the batch process. Any command that was successfully executed prior to the abort will be applied to the directory proxy server. The `--no-prompt` option is required with `dsconfig` in batch mode.

You can view the `logs/config-audit.log` file to review the configuration changes made to the directory proxy server and use them in the batch file. The batch file can have blank lines for spacing and lines starting with a pound sign (#) for comments.

## To Configure the Directory Proxy Server in dsconfig Batch Mode

1. Go to the installation directory.
2. Create a text file that lists each `dsconfig` command with the complete set of properties that you want to apply to the directory proxy server. The items in this file should be in the same format as those accepted by the `dsconfig` command.

```
# This dsconfig operation creates the exAccountNumber global
# attribute index.

dsconfig create-global-attribute-index
--processor-name dc_example_dc_com-eb-req-processor
--index-name exAccountNumber --set prime-index:true
--set guaranteed-unique:true

# Here we create the entry-count placement algorithm with the
# default behavior of adding entries to the smallest backend
# dataset first.

dsconfig create-placement-algorithm --processor-name
dc_example_dc_com-eb-req-processor
--algorithm-name example_com_entry_count
--type entry-counter --set enabled:true
--set "poll-interval:1 m"

# Note that once the entry-count placement algorithm is created
# and enabled, we can delete the round-robin algorithm.
# Since an entry-balancing proxy must always have a placement
# algorithm, we add a second algorithm and then delete the
# original round-robin algorithm created during the setup
# procedure.

dsconfig delete-placement-algorithm
--processor-name dc_example_dc_com-eb-req-processor
--algorithm-name round-robin
```

3. Use `dsconfig` with the `--batch-file` option to execute the batch file as follows:

```
root@proxy-01:bin/dsconfig --no-prompt
--bindDN "cn=directory manager" --bindPassword password
--port 389 --batch-file ../dsconfig.post-setup
Batch file '../dsconfig.post-setup' contains 3 commands

Executing: create-global-attribute-index --no-prompt
--bindDN "cn=directory manager" --bindPassword *****
--port 1389
--processor-name dc_example_dc_com-eb-req-processor
--index-name exAccountNumber --set prime-index:true
--set guaranteed-unique:true

Executing: create-placement-algorithm --no-prompt
--bindDN "cn=directory manager" --bindPassword *****
```

```
--port 1389
--processor-name dc_example_dc_com-eb-req-processor
--algorithm-name example_com_entry_count --type entry-counter
--set enabled:true --set "poll-interval:1 m"

Executing: delete-placement-algorithm --no-prompt
--bindDN "cn=directory manager" --bindPassword *****
--port 1389
--processor-name dc_example_dc_com-eb-req-processor
--algorithm-name round-robin
```

## Configuring Global Administrator Accounts

A global administrator is responsible for managing administrative server groups. A server group is an administration domain that allow you to synchronize configuration changes to one or all of the servers in the group. For example, you would set up such a group when configuring a replication topology, where configuration changes to one server can be applied to all of the servers at a time.

You can create new global administrators and remove existing global administrators using the `dsframework` tool. The global administrator entries are located in the `cn=Administrators,cn=admin data`.

### To Create a New Global Administrator Using `dsframework`

1. Use `dsframework` to create a new global administrator.

```
$ bin/dsframework create-admin-user \
--userID admin2 --set password:secret \
--hostname server1.example.com --port 1389 \
--bindDN "uid=admin,dc=example,dc=com" --bindPassword secret
```

2. To verify the creation of the new administrator, use the `list-admin-user` subcommand with `dsframework`.

```
$ bin/dsframework list-admin-user \
--port 1389 \
--bindDN "uid=admin,dc=example,dc=com" \
--bindPassword secret
```

```
id: admin2
id: admin
```

### To Remove a Global Administrator

1. Use `dsframework` to delete an existing global administrator.

```
$ bin/dsframework delete-admin-user \
--userID admin2 \
--hostname server1.example.com \
--port 1389 \
```

```
--bindDN "uid=admin,dc=example,dc=com" \  
--bindPassword secret
```

2. To verify the deletion of the global administrator, use the **list-admin-user** subcommand with **dsframework**.

```
$ bin/dsframework list-admin-user \  
--port 1389 \  
--bindDN "uid=admin,dc=example,dc=com" \  
--bindPassword secret
```

```
id: admin
```

## Configuring Administrative Server Groups

The UnboundID Directory Proxy Server provides a mechanism for setting up administrative domains that synchronize configuration changes among servers in a server group. After you have set up a server group, you can make an update on one server using **dsconfig**, then you can apply the change to the other servers in the group using the **--applyChangeTo server-group** option of the **dsconfig** non-interactive command. If you want to apply the change to one server in the group, use the **--applyChangeTo single-server** option. When using **dsconfig** in interactive command-line mode, you will be asked if you want to apply the change to a single server or to all servers in the server group.

### Configuring a Server Group

You can create an administrative server group using the **dsframework** tool. The general process is to create a group, register each server, add each server to the group, and then set a global configuration property to use the server group.

The following example procedure adds three directory proxy server instances into the server group labelled "group-one". The commands are run on **server1.example.com**.

Server	Host Name	LDAP Port
instance 1	server1.example.com	1389
instance 2	server2.example.com	2389
instance 3	server3.example.com	3389

### To Create a Server Group

1. Create a group called "group-one" using **dsframework**.

```
$ bin/dsframework create-group --groupName group-one \  
--description "Server Group One" \  
--hostname server1.example.com --port 1389 \  
--bindDN "uid=admin,dc=example,dc=com" --bindPassword secret
```

2. Register each directory proxy server that you want to add to the server group.

```
$ bin/dsframework register-server \  
--serverID server1.example.com:1389 \  
--set hostname:server1.example.com \  
--set ldapport:1389 \  
--set ldapEnabled:true \  
--hostname server1.example.com --port 1389 \  
--bindDN "uid=admin,dc=example,dc=com" --bindPassword secret  
  
$ bin/dsframework register-server \  
--serverID server2.example.com:2389 \  
--set hostname:server2.example.com \  
--set ldapport:2389 \  
--set ldapEnabled:true \  
--hostname server1.example.com --port 1389 \  
--bindDN "uid=admin,dc=example,dc=com" --bindPassword secret  
  
$ bin/dsframework register-server \  
--serverID server3.example.com:3389 \  
--set hostname:server3.example.com \  
--set ldapport:3389 \  
--set ldapEnabled:true \  
--hostname server1.example.com --port 1389 \  
--bindDN "uid=admin,dc=example,dc=com" --bindPassword secret
```

3. Add each directory proxy server to the group.

```
$ bin/dsframework add-to-group \  
--groupName group-one \  
--memberName server1.example.com:1389 \  
--hostname server1.example.com --port 1389 \  
--bindDN "uid=admin,dc=example,dc=com" --bindPassword secret  
  
$ bin/dsframework add-to-group \  
--groupName group-one \  
--memberName server2.example.com:2389 \  
--hostname server1.example.com --port 1389 \  
--bindDN "uid=admin,dc=example,dc=com" --bindPassword secret  
  
$ bin/dsframework add-to-group \  
--groupName group-one \  
--memberName server3.example.com:3389 \  
--hostname server1.example.com --port 1389 \  
--bindDN "uid=admin,dc=example,dc=com" --bindPassword secret
```

4. Set a global configuration property using the **dsconfig** tool.

```
$ bin/dsconfig set-global-configuration-prop \  
--set configuration-server-group:group-one \  
--hostname server1.example.com --port 1389 \  
--bindDN "uid=admin,dc=example,dc=com" --bindPassword secret \  
--no-prompt
```

5. Test the server group. In this example, enable the log publisher for each directory proxy server in the group, **server-group**, by using the **--applyChangeTo** option.

```
$ bin/dsconfig set-log-publisher-prop \  
--publisher-name "File-Based Audit Logger" \  
--set enabled:true \  
--applyChangeTo server-group \  
--hostname server1.example.com --port 1389 --bindDN "uid=admin,dc=example,dc=com" \  
--bindPassword secret --no-prompt
```

View the property on the first directory proxy server instance.

```
$ bin/dsconfig get-log-publisher-prop \  
  --publisher-name "File-Based Audit Logger" \  
  --property enabled \  
  --hostname server1.example.com --port 1389 \  
  --bindDN "uid=admin,dc=example,dc=com" \  
  --bindPassword secret --no-prompt
```

```
Property : Value(s)  
-----:-----  
enabled  : true
```

View the property on the second directory proxy server instance.

```
$ bin/dsconfig get-log-publisher-prop \  
  --publisher-name "File-Based Audit Logger" \  
  --property enabled \  
  --hostname server2.example.com --port 2389 \  
  --bindDN "uid=admin,dc=example,dc=com" --bindPassword secret \  
  --no-prompt
```

```
Property : Value(s)  
-----:-----  
enabled  : true
```

View the property on the third directory proxy server instance.

```
$ bin/dsconfig get-log-publisher-prop \  
  --publisher-name "File-Based Audit Logger" \  
  --property enabled \  
  --hostname server3.example.com --port 3389 \  
  --bindDN "uid=admin,dc=example,dc=com" --bindPassword secret \  
  --no-prompt
```

```
Property : Value(s)  
-----:-----  
enabled  : true
```

6. Test the server group by disabling the log publisher on the first directory proxy server instance by using the **--applyChangeTo single-server** option.

```
$ bin/dsconfig set-log-publisher-prop \  
  --publisher-name "File-Based Audit Logger" \  
  --set enabled:disabled \  
  --applyChangeTo single-server \  
  --hostname server1.example.com --port 1389 \  
  --bindDN "uid=admin,dc=example,dc=com" --bindPassword secret \  
  --no-prompt
```

View the property on the first directory proxy server instance. The first directory proxy server instance should be disabled.

```
$ bin/dsconfig get-log-publisher-prop \  
  --publisher-name "File-Based Audit Logger" \  
  --property enabled \  
  --hostname server1.example.com --port 1389 \  
  --bindDN "uid=admin,dc=example,dc=com" --bindPassword secret \  
  --no-prompt
```

```
Property : Value(s)
-----:-----
enabled  : false
```

View the property on the second directory proxy server instance.

```
$ bin/dsconfig get-log-publisher-prop \
  --publisher-name "File-Based Audit Logger" \
  --property enabled \
  --hostname server2.example.com --port 2389 \
  --bindDN "uid=admin,dc=example,dc=com" --bindPassword secret \
  --no-prompt
```

```
Property : Value(s)
-----:-----
enabled  : true
```

View the property on the third directory proxy server instance.

```
$ bin/dsconfig get-log-publisher-prop \
  --publisher-name "File-Based Audit Logger" \
  --property enabled \
  --hostname server3.example.com --port 3389 \
  --bindDN "uid=admin,dc=example,dc=com" --bindPassword secret \
  --no-prompt
```

```
Property : Value(s)
-----:-----
enabled  : true
```

## Configuring Root User Accounts

The UnboundID Directory Proxy Server provides a default root user, `cn=Directory Manager`, that is stored in the server's configuration file. The root user is the LDAP-equivalent of a UNIX® superuser account and inherits its read-write privileges from the default root privilege set.

To limit full access to all of the directory proxy server, we recommend that you create separate administrator accounts with limited privileges so that you can identify the administrator responsible for a particular change. Having separate user accounts for each administrator also makes it possible to enable password policy functionality (such as password expiration, password history, and requiring secure authentication) for each administrator.

## Default Root Privileges

The UnboundID Directory Proxy Server contains a privilege subsystem that allows for a more fine-grained control of privilege assignments. The default set of root privileges are listed in Table 3.

**TABLE 1.** Default Root Privileges

Privilege	Description
backend-backup	Allows the user to perform backend backup operations.
backend-restore	Allows the user to perform backend restore operations.
bypass-acl	Allows the user to bypass access control evaluation.
config-read	Allows the user to read the server configuration.
config-write	Allows the user to update the server configuration.
disconnect-client	Allows the user to terminate arbitrary client connections.
ldif-export	Allows the user to perform LDIF export operations.
ldif-import	Allows the user to perform LDIF import operations.
lockdown-mode	Allows the user to request a server lockdown.
modify-acl	Allows the user to modify access control rules.
password-reset	Allows the user to reset user passwords.
privilege-change	Allows the user to change the set of privileges for a specific user, or to change the set of privileges automatically assigned to a root user.
server-restart	Allows the user to request a server restart.
server-shutdown	Allows the user to request a server shutdown.
stream-values	Allows the user to perform a stream values extended operation that obtains all entry DNs and/or all values for one or more attributes for a specified portion of the DIT.
unindexed-search	Allows the user to perform an unindexed search in the Oracle Berkeley DB Java Edition backend.
update-schema	Allows the user to update the server schema.



## Viewing the Default Root User Privileges Using dsconfig

The root DN accounts are the only user accounts that are stored within the directory proxy server's configuration under `cn=Root DNS`, `cn=config`. You can view the default privileges automatically granted to root users using the `dsconfig` tool.

### To View the Default Privileges Assigned to Root Users

Use `dsconfig` to view the root DN.

```
$ bin/dsconfig get-root-dn-prop \  
  --port 1389 --bindDN "cn=Directory Manager" \  
  --bindPassword secret --no-prompt  
Property                               : Value(s)  
-----:-----  
default-root-privilege-name : backend-backup  
default-root-privilege-name : backend-restore  
default-root-privilege-name : bypass-acl  
default-root-privilege-name : config-read  
default-root-privilege-name : config-write  
default-root-privilege-name : disconnect-client  
default-root-privilege-name : ldif-export  
default-root-privilege-name : ldif-import  
default-root-privilege-name : lockdown-mode  
default-root-privilege-name : modify-acl  
default-root-privilege-name : password-reset  
default-root-privilege-name : privilege-change  
default-root-privilege-name : server-restart  
default-root-privilege-name : server-shutdown  
default-root-privilege-name : stream-values  
default-root-privilege-name : unindexed-search  
default-root-privilege-name : update-schema
```

## Modifying a Root User Password

Root users are governed by the Root Password Policy and by default, their passwords never expire. However, in the event that you want to change a root user's password, you can use the `ldappasswordmodify` tool.

### To Modify a Root User Password

1. Open a text editor and create a text file containing the new password. In this example, name the file `rootuser.txt`.

```
$ echo password > rootuser.txt
```

2. Use `ldappasswordmodify` to change the root user's password.

```
$ bin/ldappasswordmodify \  
  --port 1389 --bindDN "cn=Directory Manager" \  
  --bindPassword secret --newPasswordFile rootuser.txt
```

```
The LDAP password modify operation was successful
```

3. Remove the text file.

```
$ rm rootuser.txt
```

## Creating a Root User

You can create another root user by adding an entry under the **cn=Root DNs,cn=config** entry. By default, the new root user will receive the default set of root privileges. If you want the new root user to have a limited set of privileges, you can remove the privileges using the **dsconfig** tool.

### To Create a Root User

1. Open a text editor, and create a file containing the root user entry.

```
dn: cn=Directory Manager2,cn=Root DNs,cn=config
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: ds-cfg-root-dn-user
userPassword: password
cn: Directory Manager2
sn: Manager2 ds-cfg-alternate-bind-dn:
cn=Directory Manager2
givenName: Directory
```

2. Use **ldapmodify** to add the entry.

```
$ bin/ldapmodify --port 1389 --bindDN "cn=Directory Manager" \
--bindPassword secret --defaultAdd --filename "rootuser.ldif"
Processing ADD request for cn=Directory Manager2,cn=Root DNs,cn=config
ADD operation successful for DN cn=Directory Manager2,cn=Root DNs,cn=config
```

## Modifying Individual Root User Privileges

All root users automatically inherit the default root privileges defined in the **default-root-privilege-name** configuration property. However, you can give individual root users additional privileges that are not included in the set of default root privileges. You can also remove default root privileges from individual root users. Modifying the privileges of the root user can be accomplished by adding the **ds-privilege-name** operational attribute to the entry for the root user. Any values containing a privilege name will grant that privilege to the user in addition to the set of default root privileges. Any values containing a minus sign followed by a privilege name will remove that privilege from that root user, even if it is included in the set of default root privileges.

### To Modify the Privileges for an Individual Root User

1. Open a text editor, and create a file containing the changes to the root user entry. The following example grants the **proxied-auth** privilege and removes the **server-shutdown** and **server-restart** privileges.

```
dn: cn=Directory Manager2,cn=Root DNs,cn=config
changetype: modify
add: ds-privilege-name
ds-privilege-name: proxied-auth
```

```
ds-privilege-name: -server-shutdown
ds-privilege-name: -server-restart
```

2. Use `ldapmodify` to apply the change.

```
$ bin/ldapmodify --port 1389 --bindDN "cn=Directory Manager" --
bindPassword secret \ --filename "modifyRootUserPrivi-
leges.ldif"
Processing MODIFY request for cn=Directory Manager2,cn=Root
DNs,cn=config
MODIFY operation successful for DN cn=Directory
Manager2,cn=Root DNs,cn=config
```

## Configuring Locations

UnboundID Directory Proxy Server defines locations, both for the LDAP external servers and the proxy server instances themselves. A location defines a collection of servers that share access and latency characteristics. For example, your deployment might include two data centers, one in the east and one in the west. These data centers would be configured as two locations in the proxy server. Each location is associated with a name and an ordered list of failover locations, which could be used if none of the servers in the preferred location are available. You can define these locations using the Directory Proxy Server Management Console or the command line.

The proxy server itself is also associated with a location. This location is specified in the global configuration properties of the proxy server itself. When load-balancing algorithms are configured later, this location tell the proxy server instance that it should prefer LDAP external servers in its location, as there will be less latency.

### To Configure Locations Using `dsconfig`

1. Use the `dsconfig` tool to configure the LDAP external server locations.

```
$ bin/dsconfig
```

2. Type the hostname or IP address for your directory proxy server, or press enter to accept the default, `localhost`.

```
Directory Proxy Server hostname or IP address [localhost]:
```

3. Type the number corresponding how you want to connect to the proxy server, or press Enter to accept the default, LDAP.

```
How do you want to connect?
1) LDAP
2) LDAP with SSL
3) LDAP with StartTLS
```

4. Type the port number for your directory proxy server, or press Enter to accept the default, 389.

```
Directory Proxy Server port number [389]:
```

5. Type the administrator's bind DN or press Enter to accept the default (cn=Directory Manager), and then type the password.

```
Administrator user bind DN [cn=Directory Manager]:
Password for user 'cn=Directory Manager':
```

6. In the Directory Proxy Server configuration console menu, enter the number corresponding to location configuration, in this example 8. Then, enter 2 to create a new location.

```
Enter choice: 8
```

```
>>>> Location management menu
What would you like to do?
```

- 1) List existing Locations
- 2) Create a new Location
- 3) View and edit an existing Location
- 4) Delete an existing Location
  
- b) back
- q) quit

```
Enter choice [b]: 2
```

7. Enter the name of the new location. This example demonstrates configuring a location called East. Enter f to finish configuring the location. Repeat this procedure to create a location called West.

```
>>>> Enter a name for the location that you want to create: east
```

```
>>>> Configure the properties of the location
```

Property	Value(s)
-----	
1) description	-
2) preferred-failover-location	-
?) help	
f) finish - create the new location	
d) display the equivalent dsconfig arguments to create this object	
b) back	
q) quit	

```
Enter choice [b]: f
```

8. Next, edit the configuration of an existing location, in this example a location named East.

```
>>>> Location management menu
What would you like to do?
```

- 1) List existing locations
- 2) Create a new location
- 3) View and edit an existing location
- 4) Delete an existing location

```
b) back
q) quit

Enter choice [b]: 3

>>>> Select the location from the following list:

    1) East
    2) West

    b) back
    q) quit

Enter choice [b]: 1
```

9. Define the preferred failover location property for East. This property provides alternate locations that can be used if servers in this location are not available. If more than one location is provided, the directory proxy server tries the locations in the order listed.

```
>>>> Configure the properties of the Location

    Property                                Value(s)
    -----
    1) description                          -
    2) preferred-failover-location          -

    ?) help
    f) finish - create the new location
    d) display the equivalent dsconfig arguments to create this object
    b) back
    q) quit

Enter choice [b]: 2

...

Do you want to modify the 'preferred-failover-location' property?

    1) Use the value: West
    2) Add one or more values

    ?) help
    q) quit

Enter choice [1]: 2

Select the locations you wish to add:

    1) East
    2) West
    3) Create a new location
    4) Add all locations

...

Enter one or more choices separated by commas[b]: 2
```

10. Verify and apply your change to the property.

```
Do you want to modify the 'preferred-failover-location' property?

    1) Use the value: West
    2) Add one or more values
    3) Remove one or more values
```

```

4) Leave undefined
5) Revert changes

?) help
q) quit

Enter choice [1]:

>>>> Configure the properties of the location

      Property                      Value(s)
      -----
1) description                      -
2) preferred-failover-location      West

?) help
f) finish - apply any changes to the Location
d) display the equivalent dsconfig command lines to either
   re-create this object or only to apply pending changes
b) back
q) quit

Enter choice [b]: f

```

11. Repeat steps 8 and 9 for the West location, assigning it a failover location of East.

## To Configure the Directory Proxy Server's Location Using dsconfig

1. Use the dsconfig tool to configure the proxy server location.

```
$ bin/dsconfig
```

2. Type the hostname or IP address for your directory proxy server, or press Enter to accept the default, localhost.

```
Directory Proxy Server hostname or IP address [localhost]:
```

3. Enter how you want to connect to the proxy server.

```
How do you want to connect?
1) LDAP
2) LDAP with SSL
3) LDAP with StartTLS
```

4. Type the port number for your directory proxy server, or press Enter to accept the default, 389.

```
Directory Proxy Server port number [389]:
```

5. Type the administrator's bind DN or press Enter to accept the default (cn=Directory Manager), and then type the password.

```
Administrator user bind DN [cn=Directory Manager]:
Password for user 'cn=Directory Manager':
```

6. In the Directory Proxy Server configuration console menu, enter the number corresponding to global configuration, in this example 5. Then enter 1 to view and edit the global configuration.

```
Enter choice: 5

>>>> Global configuration management menu
What would you like to do?

    1) View and edit the Global Configuration

    b) back
    q) quit

Enter choice [b]: 1
```

7. Enter the number associated with the location configuration property, in this example 2.

```
Enter choice [b]: 2
```

8. Specify a new location for this proxy server instance, in this example the East location. Operations involving communications with other servers may prefer servers in the same location to ensure low-latency responses.

```
>>>> Configuring the 'location' property
...

Do you want to modify the 'location' property?

    1) Leave undefined
    2) Change it to the location: East
    3) Change it to the location: West
    4) Create a new location

    b) back
    q) quit

Enter choice [b]: 2
```

9. Enter `f` to finish the operation.

```
Enter choice [b]: f
```

## Configuring Server Health Checks

You can use the UnboundID Directory Proxy Server to configure different types of health checks for your deployment. The health checks define server availability as either being available, unavailable, or degraded. The server is given a value from 0 to 10, which is used to determine if the server is available and how that server compares to other servers with the same state. Load-balancing algorithms can be used to check the score and prefer servers with higher scores over those with lower scores.

Health checks can be global or server specific. For example, the server health check may be used to perform a search in the LDAP external server. If the search fails, the server may be classified as unavailable. Otherwise, the time it takes to retrieve the entry determines the availability or degradation of the server.

For more information about health checks and the type of health checks supported by UnboundID Directory Proxy Server, see “About LDAP Health Checks” on page 10.

## About the Default Health Checks

By default, the directory proxy server has two health check instances enabled for use on all servers:

- **Consume Admin Alerts.** This health check detects administrative alerts as soon as they are issued and then checks to see if the alert is associated with a server becoming degraded or unavailable.
- **Get Root DSE.** This health check detects if the root DSE entry exists on the LDAP external server. As this entry always exists on an UnboundID Directory Server, the absence of the entry suggests that the LDAP external server may be degraded or unavailable.

## About Creating a Custom Health Check

You can create a new health check from scratch or use an existing health check as a template for the configuration of a new health check. If you choose to create a custom health check, you can create one of the following types:

- **Admin Alert Health Check.** This health check watches for administrative alerts generated by the LDAP external server to determine whether the server has entered a degraded or unavailable state.
- **Replication Backlog Health Check.** This health check monitors the replication backlog of the LDAP external server instance. If a server falls too far behind in replication, then the directory proxy server can stop sending requests to it. A server is classified as degraded or unavailable if the threshold is reached for the number of missing changes, the age of the oldest missing change, or both.
- **Search LDAP Health Check.** This health check performs searches on an LDAP external server and gauges the health of the server depending upon the entry returned. For example, if an error occurs while attempting to communicate with the server, then the server is considered unavailable. You can also run a filter on the entry returned, for example running a search on a monitor attribute.
- **Work Queue Busyness Health Check.** This health check may be used to monitor the percentage of time that worker threads in backend servers spend processing requests.

## To Configure a Health Check Using dsconfig

1. Use the `dsconfig` tool to configure a health check.

```
$ bin/dsconfig
```



2. Type the hostname or IP address for your directory proxy server, or press enter to accept the default, localhost.

Directory Proxy Server hostname or IP address [localhost]:

3. Enter how you want to connect to the proxy server, or press Enter to accept the default, LDAP.

How do you want to connect?

- 1) LDAP
- 2) LDAP with SSL
- 3) LDAP with StartTLS

Enter choice [1]:

4. Type the port number for your directory proxy server, or press Enter to accept the default, 389.

Directory Proxy Server port number [389]:

5. Type the administrator's bind DN or press Enter to accept the default (cn=Directory Manager), and then type the password.

Administrator user bind DN [cn=Directory Manager]:

Password for user 'cn=Directory Manager':

6. In the Directory Proxy Server configuration console menu, enter the number corresponding to LDAP health checks, in this example 6. Enter 2, then press n to create a new health check from scratch.

Enter choice: 6

>>>> LDAP Health Check management menu

What would you like to do?

- 1) List existing LDAP Health Checks
  - 2) Create a new LDAP Health Check
  - 3) View and edit an existing LDAP Health Check
  - 4) Delete an existing LDAP Health Check
- 
- b) back
  - q) quit

Enter choice [b]: 2

>>>> Select an existing LDAP Health Check to use as a template for the new LDAP Health Check configuration or 'n' to create one from scratch:

- 1) Consume Admin Alters
  - 2) Get Root DSE
- 
- n) new LDAP Health Check created from scratch
  - c) cancel
  - q) quit

Enter choice [n]: n

7. Select the type of health check you want to create. This example demonstrates the creation of a new search LDAP health check.

>>>> Select the type of LDAP external server health check that you want to create:

- 1) Admin Alert LDAP Health Check
- 2) Replication Backlog LDAP Health Check
- 3) Search LDAP Health Check

? ) help  
c ) cancel  
q ) quit

Enter choice [c]: 3

8. Specify a name for the new health check. In this example, the health check is named Get example.com.

>>>> Enter a name for the search LDAP Health Check that you want to create:  
Get example.com

9. Enable the new health check.

>>>> Configuring the 'enabled' property

Indicates whether this LDAP health check is enabled for use in the server.

Select a value for the 'enabled' property:

- 1) true
  - 2) false
- ? ) help  
c ) cancel  
q ) quit

Enter choice [c]: 1

10. Next, configure the properties of the health check. This example changes the base DN and the response time threshold. The response time threshold can be different for local and non-local servers, for example, to accommodate responses coming over a WAN.

>>>> Configure the properties of the Search LDAP Health Check

	Property	Value(s)
	-----	-----
1)	description	-
2)	enabled	true
3)	use-for-all-servers	false
4)	base-dn	" "
5)	scope	base-object
6)	filter	(objectClass=*)
7)	maximum-local-available-response-time	1 s
8)	maximum-nonlocal-available-response-time	1 s

```

9)  minimum-local-degraded-response-time      500 ms
10) minimum-nonlocal-degraded-response-time    500 ms
11) maximum-local-degraded-response-time       10 s
12) maximum-nonlocal-degraded-response-time    10 s
13) minimum-local-unavailable-response-time     5 s
14) minimum-nonlocal-unavailable-response-time  5 s
15) allow-no-entries-returned                  true
16) allow-multiple-entries-returned            true
17) available-filter                           -
18) degraded-filter                           -
19) unavailable-filter                         -

?)  help
f)  finish - create the new Search LDAP Health Check
d)  display the equivalent dsconfig arguments to create this object
b)  back
q)  quit

```

Enter choice [b]: 4

>>>>Configuring the 'base-dn' property

The base DN to use for the search. If no value is provided, then the search will be based at the server's root DSE.

Syntax: DN

Do you want to modify the 'base-dn' property?

- 1) Keep the default value: ""
- 2) Change the value

```

?)  help
q)  quit

```

Enter choice [1]: 2

11. Next, configure the response time thresholds. The maximum available response time defines the time after which the server is no longer considered available. The minimum degraded response time is shorter than the maximum response time to avoid constant state fluctuations. The server has to be significantly improved before it is upgraded out of the degraded state.

This example configures the maximum allowed response time threshold for local servers to be 500 ms.

Enter choice [b]: 7

...

Do you want to modify the 'maximum-local-available-response-time' property?

- 1) Keep the default value: 1 s
- 2) Change the value

```

?)  help
q)  quit

```

Enter choice [1]: 2

Enter a value for the 'maximum-local-available-response-time' property [continue]: 500 ms

12. Configure the new health check to be automatically used for all LDAP external servers by setting the use-for-all-servers property as follows.

Enter choice [b]: 3

```
...
Do you want to modify the 'use-for-all-servers' property?

1) Keep the default value: false
2) Change it to the value: true

?) help
q) quit

Enter choice [1]: 2

13. Enter f to finish the operation.

Enter choice [b]: f
```

## Configuring LDAP External Servers

During this step, you define each of the LDAP external servers that the directory proxy server communicates with, including the server type. You also assign the server-specific health checks configured in the previous step.

UnboundID Directory Proxy Server includes a tool, `prepare-external-server`, for configuring communication between the directory proxy server and the LDAP backend server. After you add a new LDAP server, we strongly recommend that you run this script to automatically create the user account necessary for communications. When you run this tool, you must supply the user account and password that you specified for the directory proxy server during configuration, **cn=Proxy User** by default.

Note that you should not use **cn=Directory Manager** as the account to use for communication between the directory proxy server and the directory server. For security reasons, the account used to communicate between the directory proxy server and the directory server should not be directly accessible by clients accessing the directory proxy server. The account that you choose should meet the following criteria:

- For all server types, it should not exist in the directory proxy server but only in the backend directory server instances.
- For UnboundID Directory Server and Alcatel-Lucent 8661 Directory Server, this user should be a root user.
- For UnboundID Directory Server and Alcatel-Lucent 8661 Directory Server, this user should not automatically inherit the default set of root privileges, but instead should have exactly the following set of privileges: `bypass-read-acl`, `config-read`, `lockdown-mode`, `proxied-auth`, and `stream-values`.
- For Sun Directory Servers, the account should be created below the **cn=Root DNs**, **cn=config** entry and the **nsSizeLimit**, **nsTimeLimit**, **nsLookThroughLimit**, and **nsIdleTimeout** values for the account should be set to `-1`. You also need to create

access control rules to grant the user account appropriate permissions within the server. The `prepare-external-server` tool handles all of this work automatically.

## To Configure an External Server Using `dsconfig`

1. Use the `dsconfig` tool to create and configure external servers.

```
$ bin/dsconfig
```

Specify the hostname, connection method, port number, and bind DN as described in previous procedures.

2. In the Directory Proxy Server configuration console menu, enter the number corresponding to external servers, in this example 3. Then enter 2 to create a new external server.

```
Enter choice: 3
```

```
>>>> External server management menu
What would you like to do?
```

- 1) List existing external servers
- 2) Create a new external server
- 3) View and edit an existing external server
- 4) Delete an existing external server
  
- b) back
- q) quit

```
Enter choice [b]: 2
```

3. Select the type of server you want to create. This example creates a new UnboundID Directory Server.

```
>>>> Select the type of external server that you want to create:
```

- 1) **UnboundID** DS external server
- 2) JDBC external server
- 2) LDAP external server
- 3) Sun DS external server
  
- ? ) help
- c) cancel
- q) quit

```
Enter choice [c]: 1
```

4. Specify a name for the new external server. In this example, the external server is named `east1`.

```
>>>> Enter a name for the UnboundID DS external server that you want to
create: east1
```

5. Configure the host name or IP address of the target LDAP external server.

```
Enter a value for the 'server-host-name' property: east1.example.com
```

6. Next, configure the location property of the new external server.

```
Enter choice [b]: 4
```

```
...
```

```
Do you want to modify the 'location' property?
```

- 1) Leave undefined
- 2) Change it to the location: East
- 3) change it to the location: West
- 4) Create a new location

```
? ) help
```

```
q ) quit
```

```
Enter choice [1]: 2
```

7. Next, define the bind DN and bind password.

```
Enter choice [b]: 5
```

```
...
```

```
Do you want to modify the 'bind-dn' property?
```

- 1) Leave undefined
- 2) Change the value

```
? ) help
```

```
q ) quit
```

```
Enter choice [1]: 2
```

```
Enter a value for the 'bind-dn' property [continue]: uid=proxy,dc=example,dc=com
```

```
Enter choice [b]: 6
```

```
...
```

```
Do you want to modify the 'password' property?
```

- 1) Leave undefined
- 2) Change the value

```
? ) help
```

```
q ) quit
```

```
Enter choice [1]: 2
```

```
Enter a value for the 'password' property [continue]:  
Confirm the value for the 'password' property:
```

8. Enter `f` to finish the operation.

```
Enter choice [b]: f
```

```
The Unbound DS external server was created successfully.
```

For our example, repeat steps 2-8 to create one server, east1:389, which is in the east location, and two servers, west1:389, and west2:389, which are in the west location.

Once you have completed adding the servers, run the **prepare-external-server** tool to configure communications between the directory proxy server and the UnboundID Directory Servers.

## To Configure Server Communications Using the **prepare-external-server** Tool

Use the **prepare-external-server** tool if you have added LDAP external servers using **dsconfig**. The **create-initial-proxy-config** tool automatically runs the **prepare-external-server** tool to configure server communications so that you do not need to invoke it separately. The **create-initial-proxy-config** tool verifies that the proxy user account exists and has the correct password and required privileges. If it detects any problems, it prompts for manager credentials to rectify them.

If you want the **prepare-external-server** tool to add the LDAP external server's certificates to the directory proxy server's trust store, you must include the **--proxyTrustStorePath** option, the **--bindDN**, and either the **--proxyTrustStorePassword** or the **--proxyTrustStorePasswordFile** option. The default location of the directory proxy server trust store is **config/truststore**. The pin is encoded in the **config/keystore.pin** file.

For example, run the tool as follows to prepare an UnboundID Directory Server on the local host and listening on port 1389 for access by the directory proxy server using the default user account **cn=Proxy User**:

```
prepare-external-server --port 1389 --baseDN dc=example,dc=com
--proxyBindPassword secret
```

The following example illustrates how to run the tool to prepare an UnboundID Directory Server on the local host and listening on port 1636. The directory server is being accessed by a directory proxy server that uses the default user account **cn=Proxy User**. In anticipation of the directory proxy server using SSL-based communication with this LDAP external server, the server's trusted certificate will be placed in the trust store:

```
$:~$ ./UnboundID-Proxy/bin/prepare-external-server --baseDN dc=example,dc=com
--proxyBindPassword password --hostname server1 --useSSL --port 1636
--proxyTrustStorePath /full/path/to/trust/store --proxyTrustStorePassword secret
```

```
Testing connection to server1:1636 .....
```

```
Do you wish to trust the following certificate?
```

```
Certificate Subject: CN=server1, O=Example Self-Signed Certificate
Issuer Subject:      CN=server1, O=Example Self-Signed Certificate
Validity:            Thu May 21 08:02:30 CDT 2009 to Wed May 16 08:02:30 CDT 2029
```

```
Enter 'y' to trust the certificate or 'n' to reject it.
```

```
y
```

```
The certificate was added to the local trust store
```

```
Done
Testing 'cn=Proxy User' access to server1:5636 ..... Failed to bind as 'cn=Proxy User'

Would you like to create or modify root user 'cn=Proxy User' so that it is available
for this Proxy Server? (yes / no) [yes]:

Enter the DN of an account on server1:5636 with which to create or manage the 'cn=Proxy
User' account [cn=Directory Manager]:

Enter the password for 'cn=Directory Manager':

Created 'cn=Proxy User,cn=Root DNs,cn=config'

Testing 'cn=Proxy User' privileges ..... Done
```

## Configuring Load Balancing

You can distribute the load on your proxy server using one of the load-balancing algorithms provided with UnboundID Directory Proxy Server. In general, the directory proxy server will prefer local servers over non-local servers, unless you configure the **use-location** property of the load-balancing algorithm to false. Within a given location, the directory proxy server prefers available servers over degraded servers. This means that if at all possible, the directory proxy server will send requests to servers that are local and available before considering selecting any server that is non-local or degraded.

The directory proxy server provides the following load-balancing algorithms:

- **Fewest operations load balancing**  
This algorithm forwards requests to the backend server with the fewest operations currently in progress. This algorithm can also take the location and health of the servers into account when making the determination, so that servers in remote locations and/or with poor availability have a lower priority.
- **Round robin load balancing**  
This algorithm spreads the work across a set of servers. You can specify that the algorithm spread work only to servers in the same location as the proxy server. If these servers are not available, then it only user servers in the primary failover location, and so on.
- **Single server load balancing**  
This algorithm forwards all operations to a single UnboundID Directory Server that you specify.
- **Failover load balancing**  
This algorithm forwards requests to servers in a given order, optionally taking the location into account. If the preferred server is not available, then it will fail over to the alternate server in a predefined order.
- **Weighted load balancing.**  
This algorithm uses statically defined weights for sets of servers.
- **Health weighted load balancing**  
This algorithm assigns weights to servers based on their health scores and, optionally, their



locations. For example, servers with a higher health check score will receive a higher proportion of the requests than servers with lower health check scores.

- Location-based load balancing  
This algorithm forwards requests to servers in a given location.

## To Configure Load Balancing Using `dsconfig`

1. Use the `dsconfig` tool to create and configure a subtree view.

```
$ bin/dsconfig
```

Specify the hostname, connection method, port number, and bind DN as described in previous procedures.

2. In the Directory Proxy Server configuration console menu, enter the number associated with load-balancing algorithms, in this example 7.

```
Enter choice: 7
```

```
>>>> Load-Balancing Algorithm management menu
```

```
What would you like to do?
```

- 1) List existing Load-Balancing Algorithms
- 2) Create a new Load-Balancing Algorithm
- 3) View and edit an existing Load-Balancing Algorithm
- 4) Delete an existing Load-Balancing Algorithm

```
? ) help
```

```
q ) quit
```

```
Enter choice [1]: 2
```

3. Select an existing load-balancing algorithm to use as a template or select **n** to create a new load-balancing algorithm from scratch.

```
1) example-lb
```

```
n) new Load-Balancing Algorithm created from scratch
```

```
c) cancel
```

```
q) quit
```

```
Enter a choice [n]: n
```

4. Select the type of load-balancing algorithm that you want to create. Depending on type of algorithm you select, you will be guided through a series of configuration properties, such as providing a name and selecting an LDAP external server.

```
>>>> Select the type of Load Balancing Algorithm that you want to create:
```

```
1) Failover Load Balancing Algorithm
```

```
2) Fewest Operations Load Balancing Algorithm
```

- 3) Health Weighted Load Balancing Algorithm
- 4) Round Robin Load Balancing Algorithm
- 5) Single Server Load Balancing Algorithm
- 6) Weighted Load Balancing Algorithm

- ? ) help
- c ) cancel
- q ) quit

Enter choice [c]: 3

5. Review the configuration properties for your new load-balancing algorithm. If you are satisfied, enter f to finish.

## Configuring Proxy Transformations

The UnboundID Directory Proxy Server provides proxy transformations to alter the contents of client requests as they are sent from the client to the LDAP external server. Proxy transformations can also be used to alter the responses sent back from the server to the client, including altering or omitting search result entries. The directory proxy server provides the following types of data transformations:

- **Attribute mapping**  
This transformation rewrites client requests so that references to one attribute type may be replaced with an alternate attribute type. The proxy server can perform extensive replacements, including attribute names used in DN's and attribute names encoded in the values of a number of different controls and extended operations. For example, a client requests a `userid` attribute, which is replaced with `uid` before being forwarded on to the backend server. This mapping applies in reverse for the response returned to the client.
- **DN mapping**  
This transformation rewrites client requests so that references to entries below a specified DN will be mapped to appear below another DN. For example, references to entries below `o=example.com` could be rewritten so that they are below `dc=example,dc=com` instead. The mapping applies in reverse for the response returned to the client.
- **Suppress attribute**  
This proxy transformation allows you to exclude a specified attribute from search result entries. It also provides the ability to reject add, compare, modify, modify DN, or search requests if they attempt to reference the target attribute.
- **Suppress entry**  
This proxy transformation allows you to exclude any entries that match a specified filter from a set of search results. Search requests are transformed so that the original filter will be ANDed with a NOT filter containing the exclude filter. For example, if the suppression filter is `"(objectClass=secretEntry)"`, then a search request with a filter of

"(uid=john.doe)" will be transformed so that it has a filter of  
"(&(uid=john.doe) (! (objectClass=secretEntry)))".

- Simple to external bind  
This proxy transformation may be used to intercept a simple bind request and instead process the bind as a SASL EXTERNAL bind. If the SASL EXTERNAL bind fails, then the original simple bind request may or may not be processed, depending on how you configure the server.

## To Configure Data Transformations Using dsconfig

1. Use the `dsconfig` tool to create and configure a subtree view.

```
$ bin/dsconfig
```

Specify the hostname, connection method, port number, and bind DN as described in previous procedures.

2. In the Directory Proxy Server configuration console menu, enter the number associated with proxy transformations, in this example 11. Then enter 2 in the Proxy Transformation management menu to create a new proxy transformation.

```
Enter choice: 11
```

```
>>>> Proxy Transformation management menu
```

```
What would you like to do?
```

- 1) List existing Proxy Transformations
- 2) Create a new Proxy Transformation
- 3) View and edit an existing Proxy Transformation
- 4) Delete an existing Proxy Transformation
- b) back
- q) quit

```
Enter choice [1]: 2
```

3. Select the type of proxy transformation you want to create. This example configures an attribute mapping transformation.

```
>>>> Select the type of Proxy Transformation that you want to create:
```

- 1) Attribute Mapping Proxy Transformation
- 2) DN Mapping Proxy Transformation
- ? ) help
- c) cancel
- q) quit

```
Enter choice [c]: 1
```

4. Enter a name for the new transformation.

```
>>>> Enter a name for the Attribute Mapping Proxy Transformation that you
want to create: example_attribute
```

5. Indicate whether you want the transformation to be enabled by default.

```
Select a value for the 'enabled' property:
```

```
1) true
2) false

?) help
c) cancel
q) quit
```

```
Enter choice [c]: 1
```

6. Specify the name of the client attribute that you want to remap to a target attribute. Note that this attribute must not be equal to the target attribute.

```
Enter a value for the 'source-attribute' property: userid
```

7. Specify the name of the target attribute to which the client attribute should be mapped.

```
Enter a value for the 'target-attribute' property: uid
```

8. The properties of your new proxy transformation are displayed. If you want to make any further modifications, enter the number corresponding to the property. Enter f to finish the creation of the proxy transformation.

```
Enter choice [b]: f
```

## Configuring Request Processors

A request processor is responsible for handling operations within a portion of the DIT. You can create one of the following types of request processors:

- Proxying request processor. This request processor is used to forward requests for processing to an LDAP external server. Multiple servers may be configured to provide high availability and load balancing. Data transformations can be applied to requests and responses that are processed.
- Entry-balancing request processor. This request processor is used to distribute entries under a common parent entry among multiple backend sets. A backend set is a collection of replicated directory servers that contain identical portions of the data. This request processor uses multiple proxying request processors to process operations for the various backend LDAP servers.
- Failover request processor. This request processor performs ordered failover between other types of request processors, sometimes with different behavior for different types of opera-

tions. For example, you could use a failover request processor to achieve round-robin load balancing for read operations but failover load-balancing for writes

## To Configure a Request Processor Using `dsconfig`

1. Use the `dsconfig` tool to create and configure a request processor.

```
$ bin/dsconfig
```

Specify the hostname, connection method, port number, and bind DN as described in previous procedures.

2. In the Directory Proxy Server configuration console menu, enter the number associated with request processor configuration, in this example 14.

```
Enter choice: 14
```

```
>>>> Request Processor management menu
```

```
What would you like to do?
```

- 1) List existing Request Processors
- 2) Create a new Request Processor
- 3) View and edit an existing Request Processor
- 4) Delete an existing Request Processor
  
- b) back
- q) quit

```
Enter choice [b]: 2
```

3. Select an existing request processor to use as a template for creating a new one or enter `n` to create one from scratch.

```
Enter Choice [n]: n
```

4. Select the type of request processor that you want to create. In this example, we create a new proxying request processor.

Only select the entry-balancing request processor if you are configuring an entry balancing deployment of directory proxy server. When you create a new entry-balancing request processor, you will be prompted to select one or more proxying request processors that is uses to route operations as well as a entry balancing base DN.

```
>>>> Select the type of Request Processor that you want to create:
```

- 1) Entry-Balancing Request Processor
- 2) Proxying Request Processor

```
? ) help
c ) cancel
q ) quit
```

```
Enter choice [c]: 2
```

5. Choose a load-balancing algorithm used by the request processor to select the LDAP external server that will service the operation.

```
>>>>Select a Load-Balancing Algorithm for the 'load-balancing-algorithm' property:
```

```
1) example-lb
2) Create a new Load-Balancing Algorithm
```

```
? ) help
c ) cancel
```

```
Enter a choice [c]: 1
```

6. Review the configuration properties of the new request processor. If you are satisfied, enter f to finish.

## Passing LDAP Controls With the Proxying Request Processor

If your deployment does not use entry balancing and includes applications that use LDAP controls, such as the virtual list view (VLV) or LDAP join controls, you need to configure the directory proxy server to forward these controls correctly. You do so by configuring the **supported-control-oid** property to define the request and response OIDs of the LDAP control. The controls use the following OIDs:

- LDAP Join Control: 1.3.6.1.4.1.30221.2.5.9
- Virtual List View 2.16.840.1.113730.3.4.9 and 2.16.840.1.113730.3.4.10
- Server-Side Sort: 1.2.840.113556.1.4.473 and 1.2.840.113556.1.4.474
- Simple Paged Results: 1.2.840.113556.1.4.319

For example, using `dsconfig` tool, modify the **supported-control-oid** property as follows:

1. From the Request Processor management menu, enter 3 to view and edit an existing request processor.

```
>>>> >>>> Request Processor management menu
```

```
What would you like to do?
```

```
1) List existing Request Processors
2) Create a new Request Processor
3) View and edit an existing Request Processor
4) Delete an existing Request Processor
```

- b) back
- q) quit

Enter choice [b]: 3

2. Enter 8 to edit the **supported-control-oid** property.

>>>> >>>> Configure the properties of the Proxying Request Processor

Property	Value(s)
-----	
1) description	-
2) enabled	true
3) allowed-operation	abandon, add, bind, compare, delete, extended, modify, modify-dn, search
4) load-balancing-algorithm	dc_example_dc_com-round-robin
5) transformation	-
6) referral-behavior	pass-through
7) supported-control	account-usable, assertion, authorization-identity, get-authorization-entry, get-effective-rights, ignore-no-user-modification, intermediate-client, manage-dsa-it, matched-values, no-op, password-policy, post-read, pre-read, proxied-authorization-v1, proxied-authorization-v2, real-attributes-only, retain-identity, server-side-sort, subentries, subtree-delete, virtual-attributes-only
8) supported-control-oid	-
?) help	
f) finish	- apply any changes to the Proxying Request Processor
d) display	the equivalent dsconfig command lines to either re create this object or only to apply pending changes
b) back	
q) quit	

Enter choice [b]:

3. Next, enter 2 to add new values to the **supported-control-oid** property.

Do you want to modify the 'supported-control-oid' property?

- 1) Leave undefined
- 2) Add one or more values
- ?) help
- q) quit

Enter choice [1]: 2

4. Enter the request and response OIDs of the VLV control.

```
Enter a value for the 'supported-control-oid' property [con-
tinue]: \ 2.16.840.1.113730.3.4.9
```

```
Enter another value for the 'supported-control-oid' property \
[continue]: 2.16.840.1.113730.3.4.10
```

```
Enter another value for the 'supported-control-oid' property [con-
tinue]:
```

5. Enter 1 to update the `supported-control-oid` property to show the new OID values.

The `'supported-control-oid'` property has the following values:

- \*) 2.16.840.1.113730.3.4.9
- \*) 2.16.840.1.113730.3.4.10

Do you want to modify the `'supported-control-oid'` property?

- 1) Use these values
- 2) Add one or more values
- 3) Remove one or more values
- 4) Leave undefined
- 5) Revert changes
- ?) help
- q) quit

## Configuring Server Affinity

The directory proxy server supports server affinity. It provides the following algorithms:

- Client Connection.  
Requests from the same connection are consistently routed to the same backend server.
- Client IP  
Requests from all client connections from the same client IP address are routed to the same backend server.
- Bind DN  
Requests from all client connections authenticated as the same user are routed to the same backend server.

For each algorithm, you can specify the set of operations for which an affinity will be established, as well as the set of operations for which affinity will be used. Affinity assignments have a timeout value so that they are in effect for some period of time after the last operation that may cause the affinity to be set or updated.



## To Configure Server Affinity Using dsconfig

1. Use the `dsconfig` tool to configure server affinity.

```
$ bin/dsconfig
```

Specify the hostname, connection method, port number, and bind DN as described in previous procedures.

2. In the Directory Proxy Server configuration console menu, enter the number associated with server affinity provider configuration, in this example 15.

```
Enter choice: 15
```

```
>>>> Server Affinity Provider management menu
```

```
What would you like to do?
```

- 1) List existing Server Affinity Provider
- 2) Create a new Server Affinity Provider
- 3) View and edit an existing Server Affinity Provider
- 4) Delete an existing Server Affinity Provider
  
- b) back
- q) quit

```
Enter choice [b]: 2
```

3. We choose the type of server affinity provider. In this example, we create a bind DN server affinity provider.

```
Select the type of Server Affinity Provider that you want to create:
```

- 1) Bind DN Server Affinity Provider
- 2) Client Connection Server Affinity Provider
- 3) Client IP Server Affinity Provider
  
- ?) help
- c) cancel
- q) quit

```
Enter choice [c]:
```

4. Enter a name for your new server affinity provider.

```
>>>> Enter a name for the Bind DN Server Affinity Provider that you  
want to create: example
```

5. Indicate whether you want the server affinity provider to be enabled for use by the directory proxy server. In this example, we select 1 to enable to the server affinity provider.

```
Select a value for the 'enabled' property:
```

- 1) true
- 2) false

```
? ) help
c ) cancel
q ) quit
```

```
Enter choice [c]: 1
```

6. Next, configure the properties of the server affinity provider. For example, you can customize the types of operations for which affinity may be set and the types of operations for which affinity may be used, as well as the length of time for which the affinity should persist. This example illustrates the properties of the bind DN server affinity provider.

```
>>>> >>>> Configure the properties of the Bind DN Server Affinity Provider
```

	Property	Value(s)
1)	description	-
2)	enabled	true
3)	affinity-duration	30 s
4)	set-affinity-operation	add, delete, modify, modify-dn
5)	use-affinity-operation	add, bind, compare, delete, modify, modify-dn

```
? ) help
f ) finish - create the new Bind DN Server Affinity Provider
d ) display the equivalent dsconfig arguments to create this object
b ) back
q ) quit
```

```
Enter choice [b]:
```

7. Review the properties of the server affinity provider. If you are satisfied, enter **f** to finish.

## Configuring Subtree Views

You provide clients access to a specific portion of the DIT using subtree views. You can configure subtree views from the command line or using the Directory Proxy Server Management Console.

When you create a subtree view, you provide the following information to configure its properties:

- Subtree view name
- Base DN managed by the subtree view
- Request processor used by the subtree view to route requests. If one does not exist already, you will create a new one.

## To Configure a Subtree View Using dsconfig

1. Use the `dsconfig` tool to create and configure a subtree view.

```
$ bin/dsconfig
```

Specify the hostname, connection method, port number, and bind DN as described in previous procedures.

2. In the Directory Proxy Server configuration console menu, enter the number associated with subtree view configuration, in this example 15.

```
Enter choice: 15
```

```
>>>> Subtree View management menu
```

```
What would you like to do?
```

- 1) List existing Subtree Views
- 2) Create a new Subtree View
- 3) View and edit an existing Subtree View
- 4) Delete an existing Subtree View

```
b) back
```

```
q) quit
```

```
Enter choice [b]: 2
```

3. Enter a name for your new subtree view.

```
>>>> Enter a name for the Subtree View that you want to create:  
example.com
```

4. Enter the base DN of the subtree managed by this subtree view.

```
Enter a value for the 'base-dn' property:dc=example,dc=com
```

5. Select a request processor for this subtree view to route requests or make the appropriate selection to create a new one.

```
Select a Request Processor for the 'request-processor' property:
```

- 1) example-rp
- 2) Create a new Request Processor

```
? ) help
```

```
c) cancel
```

```
q) quit
```

```
Enter choice [c]: 1
```

6. Choose a load-balancing algorithm used by the request processor to select the LDAP external server that will service the operation.

```
>>>>Select a Load-Balancing Algorithm for the 'load-balancing-algo-  
rithm' property:
```

- 1) example-lb
- 2) Create a new Load-Balancing Algorithm

- ?) help
- c) cancel

```
Enter a choice [c]: 1
```

7. Select the LDAP external servers that this subtree view will use.

```
Select one or more LDAP External Servers for the 'backend-server'  
property:
```

- 1) east1
- 2) Create a new LDAP External Server

- ?) help
- c) cancel
- q) quit

```
Enter one or more choices separated by commas [c]:1
```

8. Review the properties of the subtree view. If you are satisfied, enter **f** to finish.

## Configuring Client Connection Policies

Client connection policies help distinguish what portions of the DIT the client can access. They also enforce restrictions on what clients can do in the server. A client connection policy specifies criteria for membership based on information about the client connection, including client address, protocol, communication security, and authentication state and identity. The client connection policy does not control membership based on the type of request being made.

Every client connection is associated with exactly one client connection policy at any given time. A client connection policy is assigned to the client when the connection is established. The choice of which client connection policy to use will be reevaluated when the client attempts a bind to change its authentication state or uses the StartTLS extended operation to convert an insecure connection to a secure one. Any changes you make to the client connection policy do not apply to existing connections. The changes only apply to new connections.

Client connections are always unauthenticated when they are first established. If you plan to configure a policy based on authentication, you must define at least one client connection policy with criteria that match unauthenticated connections.

Once a client has been assigned to a policy, you can determine what operations they can perform. For example, your policy might allow only SASL bind operations. Client connection policies are also associated with one or more subtree views, which determine the portions of the DIT a particular client can access. For example, you might configure a policy that prevents users connecting over the extranet from accessing configuration information. The client connection policy is evaluated in addition to access control, so even a root user connecting over the extranet would not have access to the configuration information.

## Understanding the Client Connection Policy

Client connection policies are based on two things:

- **Connection criteria.** The connection criteria are used in many areas within the server. They are used by the client connection policies, but they can also be used in other instances when the server needs to perform matching based on connection-level properties, such as filtered logging. A single connection can match multiple connection criteria definitions.
- **Evaluation order index.** If multiple client connection policies are defined in the server, then each of them must have a unique value for the **evaluation-order-index** property. The client connection policies are evaluated in order of ascending evaluation order index. If a client connection does not match the criteria for any defined client connection policy, then that connection will be terminated.

If the connection policy matches a connection, then the connection is assigned to that policy and no further evaluation occurs. If, after evaluating all of the defined client connection policies, no match is found, the connection is terminated.

## When a Client Connection Policy is Assigned

A client connection policy can be associated with a client connection at the following times:

- When the connection is initially established. This association occurs exactly once for each client connection.
- After completing processing for a StartTLS operation. This association occurs at most once for a client connection, because StartTLS can not be used more than once on a particular connection. You also may not stop using TLS while keeping the connection active.
- After completing processing for a bind operation. This association occurs zero or more times for a client connection, because the bind request can be processed many times on a given connection.

StartTLS and bind requests will be subject to whatever constraints are defined for the client connection policy that is associated with the client connection at the time that the request is received. Once they have completed, then subsequent operations will be subject to the constraints of the new client connection policy assigned to that client connection. This policy may or may not be the same client connection policy that was associated with the connection before the operation was processed.

All other types of operations will be subject to whatever constraints are defined for the client connection policy used by the client connection at the time that the request is received. The client connection policy assigned to a connection never changes as a result of processing any operation other than a bind or StartTLS. So, the server will not re-evaluate the client connection policy for the connection in the course of processing an operation. For example, the client connection policy will never be re-evaluated for a search operation.

## Setting Resource Limits

Your client connection policy can specify resource limits, helping to ensure that no single client monopolizes server resources. You can limit the total number of connections to a server from a particular client or from clients that match specified criteria. You can also limit the duration of the connection.

A client connection policy may only be used to enforce additional restrictions on a client connection. You can never use it to grant a client capabilities that it would not otherwise have. You can enforce limits on the number of concurrent connections that can be established in the following ways:

- Limit the total number of concurrent connections to the server.
- Limit the total number of concurrent connections from the same IP address.
- Limit the total number of concurrent connections authenticated as the same user.
- Limit the total number of concurrent connections associated with the same client connection policy.

The first three of these limits are global configuration properties and are enforced as server-wide restrictions. The fourth is defined in the client connection policy and restrictions vary across policies.

If a client connection exceeds any of these limits, then it is terminated. For example, the server has an overall limit of 5,000 connections but a per-user limit of 10,000 connections. Therefore, the server may never have more than 5,000 connections as the same user, because you can never have more than 5,000 connections total.

## Defining the Operation Rate

You can configure the maximum operation rate for individual client connections as well as collectively for all connections associated with a client connection policy. If the operation rate limit is exceeded, the directory proxy server may either reject the operation or terminate the connection. You can define multiple rate limit values, making it possible to fine tune limits for both a long term average operation rate and short term operation bursts. For example, you can define a limit of one thousand operations per second and one million operations per day, which works out to an average of less than twelve operations per second, but with bursts of up to one thousand operations per second.

Rate limit strings should be specified as a maximum count followed by a slash and a duration. The count portion must contain an integer, and may be followed by a multiplier of **k** (to indicate that the integer should be interpreted as thousands), **m** (to indicate that the integer should be interpreted as millions), or **g** (to indicate that the integer should be interpreted as billions). The duration portion must contain a time unit of milliseconds (**ms**), seconds (**s**), minutes (**m**), hours (**h**), days (**d**), or weeks (**w**), and may be preceded by an integer to specify a quantity for that unit.

For example, the following are valid rate limit strings:

- 1/s (no more than one operation over a one-second interval)
- 10K/5h (no more than ten thousand operations over a five-hour interval)
- 5 m / 2 d (no more than five million operations over a two-day interval)

You can provide time units in many different formats. For example, a unit of seconds can be signified using `s`, `sec`, `secs`, `second`, and `seconds`.

## Client Connection Policy Deployment Example

In this example scenario, we assume the following:

- Two external LDAP clients are allowed to bind to the directory proxy server.
- Client 1 should be allowed to open only 1 connection to the proxy server.
- Client 2 should be allowed to open up to 5 connections to the proxy server.

## Defining the Connection Policies

We need to set a per-client connection policy limit on the number of connections that may be associated with a particular client connection policy. We have to define at least two client connection policies, one for each of the two clients. Each policy must have different connection criteria for selecting the policy with which a given client connection should be associated.

Because the criteria is based on authentication, we must create a third client connection policy that applies to unauthenticated clients, because client connections are always unauthenticated as soon as they are established and before they have sent a bind request. Plus, clients are not required to send a bind request as their first operation.

Therefore, we define the following three client connection policies:

- Client 1 Connection Policy, which only allows client 1, with an evaluation order index of 1.
- Client 2 Connection Policy, which only allows client 2, with an evaluation order index of 2.
- Unauthenticated Connection Policy, which allows unauthenticated clients, with an evaluation order index of 3.

We define simple connection criteria for the Client 1 Connection Policy and the Client 2 Connection Policy with the following properties:

- The **user-auth-type** must not include **none**, so that it will only apply to authenticated client connections.
- The **included-user-base-dn** should match the bind DN for the target user. This DN may be full DN for the target user, or it may be the base DN for a branch that contains a number of users that you want treated in the same way.

To create more generic criteria that match more than one user, you could list the DNs of each of the users explicitly in the **included-user-base-dn** property. If there is a group that contains all of the pertinent users, then you could instead use the **[all|any|not-all|not-any]-included-user-group-dn** property to apply to all members of that group. If the entries for all of the users match a particular filter, then you could use the **[all|any|not-all|not-any]-included-user-filter** property to match them.

## How the Policy is Evaluated

Whenever a connection is established, the server associates the connection with exactly one client connection policy. The server does this by iterating across all of the defined client connection policies in ascending order of the evaluation-order-index. Policies with a lower evaluation-order-index value will be examined before those with a higher evaluation-order-index value. The first policy that the server finds whose criteria match the client connection will be associated with that connection. If no client connection policy is found with criteria matching the connection, then the connection will be terminated.

So, in our example, when a new connection is established, the server first checks the connection criteria associated with the Client 1 Connection Policy because it has the lowest evaluation-order-index value. If it finds that the criteria do not match the new connection, the server then checks the connection criteria associated with the Client 2 Connection Policy because it has the second lowest evaluation order index. If these criteria do not match, the server finally checks the connection criteria associated with the Unauthenticated Connection Policy because it has the third lowest evaluation order index. It finds a match, so the client connection is associated with the Unauthenticated Connection Policy.

After the client performs a bind operation to authenticate to the server, then the client connection policies will be re-evaluated. If client 2 performs the bind, then the Client 1 Connection Policy will not match but the Client 2 Connection Policy will, so the connection will be re-associated with that client connection policy. Whenever a connection is associated with a client connection policy, the server will check to see if the maximum number of client connections have already been associated with that policy. If so, then the newly-associated connection will be terminated.

For example, Client 1 opens a new connection. Because it is a new connection not yet associated with connection criteria, it is assigned to the Unauthorized Connection Policy. Client 1 then sends a bind request. The determination of whether the bind operation is allowed is made based on the constraints defined in the Unauthorized Connection Policy, because it is the client connection policy already assigned to the client connection. Once the bind has completed,



then the server will re-evaluate the client connection policy against the connection criteria associated with Client 1 Connection Policy, because it has the lowest evaluation order index. The associated connection criteria match, so processing stops and the client connection is assigned to the Client 1 Connection Policy.

Next, Client 2 opens a new connection. Because it is a new connection not yet associated with connection criteria, it is assigned to the Unauthorized Connection Policy. When Client 2 sends a bind request, the operation is allowed based on the constraints defined in the Unauthorized Connection Policy. Once the bind is complete, the client connection is evaluated against the connection criteria associated with Client 1 Connection Policy, because it has the lowest evaluation order index. The associated connection criteria do not match, so the client 2 connection is evaluated against the connection criteria associated with Client 2 Connection Policy, because it has the next lowest evaluation order index. The associated connection criteria match, so processing stops and the client connection is assigned to Client 2 Connection Policy.

Client 1 sends a search request. The Client 1 Connection Policy is used to determine whether the search operation should be allowed, because this is the client connection policy assigned to the client connection for client 1. The connection is not re-evaluated, before or after processing the search operation.

## **To Configure a Client Connection Policy Using dsconfig**

1. Use the `dsconfig` tool to create and configure a client connection policy.

```
$ bin/dsconfig
```

Specify the hostname, connection method, port number, and bind DN as described in previous procedures.

2. In the Directory Proxy Server configuration console menu, enter the number associated with client connection policy configuration, in this example 1. Then enter 2 to create a new client connection policy.

```
Enter choice: 1
```

```
>>>> Client connection policy management menu  
What would you like to do?
```

- ```
1) List existing client connection policies  
2) Create a new client connection policy  
3) View and edit an existing client connection policy  
4) Delete an existing client connection policy
```

```
b) back  
q) quit
```

```
Enter choice [b]: 2
```

3. Enter `n` to create a new client connection policy from scratch.

```
>>>> Select an existing Client Connection Policy to use as a tem-
```

plate for the new Client Connection Policy configuration or 'n' to create one from scratch:

1) default

n) new Client Connection Policy created from scratch

c) cancel

q) quit

4. Enter a name for the new client connection policy.

Enter the 'policy-id' for the Client Connection Policy that you want to create: new\_policy

5. Indicate whether you want the policy to be enabled by default.

Select a value for the 'enabled' property:

1) true

2) false

?) help

c) cancel

q) quit

Enter choice [c]: 1

6. Provide a value for the evaluation-order-index property. Client connection policies with a lower index will be evaluated before those with a higher index.

Enter a value for the 'evaluation-order-index' property: 9999

7. The properties of your new client connection policy are displayed. If you want to make any further modifications, enter the number corresponding to the property. Enter f to finish the creation of the client connection policy.

Enter choice [b]: f

|             |                                                                                                                                          |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Note</b> | Any changes that you make to the client connection policy do not apply to existing connections. They will only apply to new connections. |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------|

## Configuring Entry Balancing

With entry balancing, entries below a common parent are split beneath an entry balancing point among multiple sets of directory backend servers. For information about configuring entry balancing with the **create-initial-proxy-config** tool, see “To Configure an Entry Balancing Directory Proxy Server Deployment Using the create-initial-proxy-config Tool” on page 51.

The remainder of this section describes special configuration considerations when planning for and setting up an entry balancing directory proxy server deployment. If you plan to use access control with your entry balancing deployment, refer to “Configuring Access Control with Entry Balancing” on page 168

## Determining How to Balance Your Data

If a single directory server instance can hold all of your data, then we recommend storing your data on a single server and replicating for high availability, as this simplifies your deployment. If a single server can not hold all of your data, then you can spread it across multiple servers in several ways:

- If the data is already broken up by hierarchy and all of the clients understand how to access it that way, the number of top-level branches is small and a single directory server instance can hold all of the information within one or more branches. Configure the directory proxy server with multiple base DNs and use simple load-balancing rather than entry balancing to simplify your deployment.
- If simply breaking up the data using the existing hierarchy is not an option, for example if a large number of top-level branches must be configured, then consider using entry balancing. The contents of any single branch still must fit on a given server, because only entries that are immediate subordinates of the entry-balancing base DN may be spread across multiple servers. Any entries that are further subordinates have to be placed in the same directory instance as their parent.
- If one or more branches are so large that any single directory server instance can not hold all of the data, you need to use entry balancing within that branch to further subdivide the information. You may also need to change the way that the data is arranged in the server so that it uses as flat a DIT as possible, which is easier to use in an entry-balancing deployment.

## Configuring an Entry Balancing Placement Algorithm

The directory proxy server uses an entry balancing placement algorithm to determine on which backend server it creates a new entry. You can configure one of the following placement algorithms:

- Entry counter placement algorithm. This algorithm selects the backend set with the smallest number of entries or the smallest database size to distribute entries as equally amongst the backend servers as possible.
- Hash DN placement algorithm. This algorithm forwards LDAP add requests to backend sets based on an MD5 hash of the entry DN. This algorithm ensures that a given DN maps repeatedly to the same backend set, provided that the backend sets do not change.
- Round-robin placement algorithm. This algorithm forwards LDAP add requests to backends sets in a round-robin manner. You can configure it to prefer backend sets in the same

location as the directory proxy server, but to fail over to other locations should the local servers become unavailable.

- Single set placement algorithm. This algorithm forwards LDAP add requests to the same backend set.

If you want to change your placement algorithm, add your new algorithm and then delete the one you no longer want to use.

---

# 4 Managing Directory Proxy Server

Once you have configured the UnboundID® Directory Proxy Server, you can manage the day-to-day operations of your deployment using the monitoring and logging features. This chapter provides procedures to help you tune your logs and monitor your deployment.

This chapter includes the following sections:

- Managing Logs
- Managing the Global Index in Entry Balancing Configurations
- Setting Resource Limits
- Monitoring Directory Proxy Server
- Profiling Server Performance Using the Periodic Stats Logger
- Working with Account Status Notifications
- Working with Administrative Alert Handlers
- Working with Virtual Attributes

## Managing Logs

The directory proxy server provides a number of different types of log publishers that can be used to provide information about how the server is processing.

### About the Default Logs

You can view all logs in the `UnboundID-Proxy/logs` directory. This section provides information about the following default logs:

- Error Log
- server.out Log
- Debug Log
- Config Audit Log and the Configuration Archive
- Access Log
- Setup Log File
- Tool Log Files
- LDAP SDK Debug Log

## Error Log

By default, this log file is available at `logs/errors` below the server install root and it provides information about warnings, errors, and other significant events that occur within the server. A number of messages are written to this file on startup and shutdown, but while the server is running there is normally little information written to it. In the event that a problem does occur, however, the server writes information about that problem to this file.

The following is an example of a message that might be written to the error log:

```
[01/Jan/2008:01:02:03 -0400] category=EXTENSIONS severity=SEVERE_WARNING msgID=1880227903 msg=A pause of 12345 milliseconds was detected. This might be due to a misconfiguration of the JVM including garbage collection settings
```

The `category` field provides information about the area of the server from which the message was generated. Available categories include:

|            |                |                     |
|------------|----------------|---------------------|
| CORE       | PLUGIN         | REPLICATION         |
| EXTENSIONS | JEB            | VERSION             |
| PROTOCOL   | BACKEND        | QUICKSETUP          |
| CONFIG     | TOOLS          | ADMIN_TOOL          |
| LOG        | TASKS          | DSCONFIG            |
| UTIL       | ACCESS_CONTROL | RUNTIME_INFORMATION |
| SCHEMA     | ADMIN          |                     |

The `severity` field provides information about how severe the server considers the problem to be. Available severities include:

- **DEBUG** – Used for messages that provide verbose debugging information and do not indicate any kind of problem. Note that this severity level is rarely used for error logging, as the directory proxy server provides a separate debug logging facility as described below.
- **INFORMATION** – Used for informational messages that can be useful from time to time but are not normally something that administrators need to see.
- **MILD\_WARNING** – Used for problems that the server detects, which can indicate something unusual occurred, but the warning does not prevent the server from completing the task it was working on. These warnings are not normally something that should be of concern to administrators.
- **MILD\_ERROR** – Used for problems detected by the server that prevented it from completing some processing normally but that are not considered to be a significant problem requiring administrative action.
- **NOTICE** – Used for information messages about significant events that occur within the server and are considered important enough to warrant making available to administrators under normal conditions.

- **SEVERE\_WARNING** – Used for problem that the server detects that might lead to bigger problem in the future and should be addressed by administrators.
- **SEVERE\_ERROR** – Used for significant problems that have prevented the server from successfully completing processing and are considered important.
- **FATAL\_ERROR** – Used for critical problems that arise which might leave the server unable to continue processing operations normally.

The messages written to the error log may be filtered based on their severities in two ways. First, the error log publisher has a `default-severity` property, which may be used to specify the severity of messages logged regardless of their category. By default, this includes the `NOTICE`, `SEVERE_WARNING`, `SEVERE_ERROR`, and `FATAL_ERROR` severities.

You can override these severities on a per-category basis using the `override-severity` property. If this property is used, then each value should consist of a category name followed by an equal sign and a comma-delimited set of severities that should be logged for messages in that category. For example, the following override severity would enable logging at all severity levels in the `PROTOCOL` category:

```
protocol=debug,information,mild-warning,mild-  
error,notice,severe-warning,severe-error,fatal-error
```

Note that for the purposes of this configuration property, any underscores in category or severity names should be replaced with dashes. Also, severities are not inherently hierarchical, so enabling the `DEBUG` severity for a category will not automatically enable logging at the `INFORMATION`, `MILD_WARNING`, or `MILD_ERROR` severities.

The error log configuration may be altered on the fly using tools like `dsconfig`, the web administration console, or the LDIF connection handler, and changes will take effect immediately. You can configure multiple error logs that are active in the server at the same time, writing to different log files with different configurations. For example, a new error logger may be activated with a different set of default severities to debug a short-term problem, and then that logger may be removed once the problem is resolved, so that the normal error log does not contain any of the more verbose information.

## server.out Log

The `server.out` file holds any information written to standard output or standard error while the server is running. Normally, it includes a number of messages written at startup and shutdown, as well as information about any administrative alerts generated while the server is running. In most cases, this information is also written to the error log. However, the `server.out` file can also contain output generated by the JVM. For example, if garbage collection debugging is enabled, or if a stack trace is requested via "kill -QUIT" as described in a later section, then output is written to this file.

## Debug Log

The debug log provides a means of obtaining information that can be used for troubleshooting problems but is not necessary or desirable to have available while the server is functioning normally. As a result, the debug log is disabled by default, but it can be enabled and configured at any time.

Some of the most notable configuration properties for the debug log publisher include:

- **enabled** – Indicates whether debug logging should be enabled. By default, it is disabled.
- **log-file** – Specifies the path to the file to be written. By default, debug messages are written to the `logs/debug` file.
- **default-debug-level** – Specifies the minimum log level for debug messages that should be written. The default value is “**error**,” which only provides information about errors that occur during processing (for example, exception stack traces). Other supported debug levels include **warning**, **info**, and **verbose**. Note that unlike error log severities, the debug log levels are hierarchical. Configuring a specified debug level enables any debugging at any higher levels. For example, configuring the **info** debug level automatically enables the **warning** and **error** levels.
- **default-debug-category** – Specifies the categories for debug messages that should be written. Some of the most useful categories include **caught** (provides information and stack traces for any exceptions caught during processing), **database-access** (provides information about operations performed in the underlying database), **protocol** (provides information about ASN.1 and LDAP communication performed by the server), and **data** (provides information about raw data read from or written to clients).

As with the error and access logs, multiple debug loggers can be active in the server at any time with different configurations and log files in order to help isolate information that might be relevant to a particular problem.

## Config Audit Log and the Configuration Archive

The configuration audit log provides a record of any changes made to the server configuration while the server is online. This information is written to the `logs/config-audit.log` file and provides information about the configuration change in the form that may be used to perform the operation in a non-interactive manner with the `dsconfig` command. Other information written for each change includes:

- Time that the configuration change was made.
- Connection ID and operation ID for the corresponding change, which can be used to correlate it with information in the access log.
- DN of the user requesting the configuration change and the method by which that user authenticated to the server.
- Source and destination addresses of the client connection.
- Command that can be used to undo the change and revert to the previous configuration for the associated configuration object.



In addition to information about the individual changes that are made to the configuration, the directory proxy server maintains complete copies of all previous configurations. These configurations are provided in the `config/archived-configs` directory and are gzip-compressed copies of the `config/config.ldif` file in use before the configuration change was made. The filenames contain time stamps that indicate when that configuration was first used.

## Access Log

The access log provides information about operations processed within the server. The default access log file is written to `logs/access`, but multiple access loggers can be active at the same time, each writing to different log files and using different configurations.

By default, each operation on a directory proxy server generates three access log messages. The first is written when the server first starts processing the operation and provides information about that request. The second is written just before the server has forwarded the request to an LDAP external server. The third line is written after the directory proxy server has sent the response to the client (if the operation is one that should have a response) and provides information about the result of processing that operation.

Following is an example of the three access log messages from the directory proxy server for an anonymous base-level search to retrieve the `dc=example,dc=com` entry:

```
[19/May/2009:10:34:10 -0500] SEARCH REQUEST conn=2 op=0 msgID=1
base="dc=example,dc=com" scope=0
filter="(objectClass=*)" attrs="ALL"

[19/May/2009:10:34:10 -0500] SEARCH FORWARD conn=2 op=0 msgID=1
targetHost="localhost" targetPort=1389 targetProtocol="LDAP"

[19/May/2009:10:34:10 -0500] SEARCH RESULT conn=2 op=0 msgID=1
resultCode=0 etime=11.400 from="app='UnboundID-DS' sessionID='conn=4'
responseID='op=1'" entriesReturned=1
```

The first line provides details about the search request itself. The second line tells where the request was forwarded, and the third shows the result of the processing, and provides information that can be used to correlate the operation in the directory server access log.

Following are the corresponding access log messages from the directory server:

```
[19/May/2009:10:34:10 -0500] SEARCH REQUEST conn=4 op=1 msgID=2
via="app='UnboundID-Proxy' address='127.0.0.1' authzID='dn:'
sessionID='conn=2' requestID='op=0'" base="dc=example,dc=com"
scope=0 filter="(objectClass=*)" attrs="ALL"

[19/May/2009:10:34:10 -0500] SEARCH RESULT conn=4 op=1 msgID=2
resultCode=0 etime=6.616 entriesReturned=1 authzDN=""
```

The first line provides information about the request received from the directory proxy server, including information that can be used to track it back to the corresponding lines in its access log. The second line provides information about the result of the processing.

If you look at the log messages from the directory proxy server, you can see that they have **conn=2** and **op=0** elements, which match the **sessionID** and **requestID** values in the **via** element of the directory proxy server request log message. Similarly, the directory server access log messages have **conn=4** and **op=1** elements, which match the values of the **sessionID** and **responseID** values of the **from** element in the directory server result log message.

Each log message includes a timestamp indicating when it was written, followed by the operation type (in this case "SEARCH"), the connection ID (which is used for all operations processed on the same client connection), the operation ID (which can be used to correlate the request and response log messages for the operation), and the message ID used in LDAP messages for this operation.

The remaining content for access log messages varies based on the type of operation being processed, and whether it is a request or a result message. Request messages generally include the most pertinent information from the request, but generally omit information that is sensitive or not useful.

Messages forwarded by the directory proxy server include a "FORWARD" element that contains the address and port of the backend directory server to which the request was forwarded, as well as the target protocol.

Result messages include a "RESULT" element and they may include the following elements:

- **origin=replication** – Operation that was processed as a result of data synchronization (for example, replication) rather than a request received directly from a client.
- **message** – Text that was included in the **diagnosticMessage** field of the response sent to the client.
- **additionalInfo** – Additional information about the operation that was not included in the response sent back to the client.
- **authDN** – DN of the user that authenticated to the server (typically only included in bind result messages).
- **authzDN** – DN of an alternate authorization identify used when processing the operation (for example, if the proxied authorization control was included in the request).
- **authFailureID** – Unique identifier associated with the authentication failure reason (only included in non-successful bind result messages).
- **authFailureReason** – Information about the reason that a bind operation failed that might be useful to administrators but was not included in the response to the client for security reasons.
- **responseOID** – OID included in an extended response returned to the client.
- **entriesReturned** – Number of matching entries returned to the client for a search operation.

Note that this is not an exhaustive list, and elements that are not listed here may also be present in access log messages.

## Setup Log File

The `setup` tool writes a log file providing information about the processing that it performs. By default, this log file is written to `logs/setup.log` although a different name may be used if a file with that name already exists, because the `setup` tool has already been run. The full path to the setup log file is provided when the setup tool has completed.

## Tool Log Files

Many of the administrative tools provided with the directory proxy server (such as `backup` and `restore`) can take a significant length of time to complete write information to standard output or standard error or both while the tool is running. They also write additional output to files in the `logs/tools` directory. The information written to these log files can be useful for diagnosing problems encountered while they were running.

## LDAP SDK Debug Log

This log can be used to help examine the communication between the directory proxy server and the directory server. It contains information about exceptions that occur during processing, problems establishing and terminating network connections, and problems that occur during the reading and writing of LDAP messages and LDIF entries. You can configure the types of debugging that should be enabled, the debug level that should be used, and whether debug messages should include stack traces. As for other file-based loggers, you can also specify the rotation and retention policies.

## Types of Log Publishers

UnboundID Directory Proxy Server provides a number of differently types of loggers that can be used to get processing information about the server. There are three primary types of loggers:

- Access loggers provide information about operations processed within the server. They can be used for understanding the operations performed by clients and debugging problems with directory-enabled applications, and they can also be used for collecting usage information for performance and capacity planning purposes.
- Error loggers provide information about warnings, errors, or significant events that occur within the server.
- Debug loggers can provide detailed information about processing performed by the server, including any exceptions caught during processing, detailed information about data read from or written to clients, and accesses to the underlying database.

By default, the following log publishers are enabled on the system:

- File-based access logger
- File-based error logger

UnboundID Directory Proxy Server also provides a file-based debug logger that is disabled by default.

## Creating New Log Publishers

UnboundID Directory Proxy Server provides customization options to help you create your own log publishers with the `dsconfig` command. You can also configure the log retention and rotation policies associated with the log publisher.

### To Create a New Log Publisher

1. Use the `dsconfig` command in non-interactive mode to create and configure the new log publisher. This example shows how to create a logger that only logs disconnect operations.

```
$ bin/dsconfig create-log-publisher --type file-based-access \  
  --publisher-name "Disconnect Logger" --set enabled:true \  
  --set log-connects:false \  
  --set log-requests:false \  
  --set log-results:false \  
  --set log-file:logs/disconnect.log \  
  --set log-disconnects:true \  
  --bindDN "cn=Directory Manager" --bindPassword secret \  
  --port 1389 --no-prompt
```

2. View the Log Publishers.

```
$ bin/dsconfig list-log-publishers \  
  --bindDN "cn=Directory Manager" \  
  --bindPassword secret \  
  --port 1389 \  
  --no-prompt
```

| Log Publisher            | : Type              | : enabled |
|--------------------------|---------------------|-----------|
| Disconnect Logger        | : file-based-access | : true    |
| File-Based 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    |

### To Create a Log Publisher Using dsconfig Interactive Command-Line Mode

1. On the command line, type `bin/dsconfig`.

2. Authenticate to the server by following the prompts.
3. On the UnboundID Directory Proxy Server Configuration console main menu, select the option to configure the log publisher.
4. On the Log Publisher Management menu, select the option to create a new log publisher.
5. Select the Log Publisher type. In this case, select File Based Access Log Publisher.
6. Type a name for the log publisher.
7. Enable it.
8. Type the path to the log file, relative to the directory proxy server root. For example, logs/disconnect.log.
9. On the Log Publisher Properties menu, select the option for `log-connects:false`, `log-disconnects:true`, `log-requests:false`, and `log-results:false`.
10. Type `f` to apply the changes.

## Configuring Log Rotation

The directory proxy server allows you to configure the log rotation policy for the directory proxy server. When the rotation limits are reached, the directory proxy server rotates the current log and starts a new log. You can select the following parameters:

- **Time Limit Rotation Policy.** Rotates the log based on the length of time since the last rotation. Default implementations are provided for rotation every 24 hours and every 7 days.
- **Fixed Time Rotation Policy.** Rotates the logs every day at a specified time (based on 24-hour time). The default time is 2359.
- **Size Limit Rotation Policy.** Rotates the logs when the file reaches the maximum size for each log. The default size limit is 100 megabytes.

## To Configure the Log Rotation Policy

Use `dsconfig` to modify the log rotation policy for the access logger.

```
$ bin/dsconfig set-log-publisher-prop \  
  --publisher-name "File-Based Access Logger" \  
  --remove "rotation-policy:24 Hours Time Limit Rotation Policy" \  
  --add "rotation-policy:7 Days Time Limit Rotation Policy" \  
  --hostname host1 --port 1389 --bindDN "cn=Directory Manager" \  
  --bindPassword secret
```

## Configuring Log Retention

The directory proxy server allows you to configure the log retention policy for each log on the server. When the retention limits are reached, the directory proxy server removes the oldest archived log prior to creating a new log. Log retention is only effective if you have a log rotation policy in place.

- **File Count Retention Policy.** Sets the number of log files you want the directory to retain. The default file count is 10 logs. If the file count is set to 1, then the log will continue to grow indefinitely without being rotated.
- **Free Disk Space Retention Policy.** Sets the minimum amount of free disk space. The default free disk space is 500 megabytes.
- **Size Limit Retention Policy.** Sets the maximum size of the combined archived logs. The default size limit is 500 megabytes.

### To Configure the Log Retention Policy

Use `dsconfig` to modify the log retention policy for the access logger.

```
$ bin/dsconfig set-log-publisher-prop \  
  --publisher-name "File-Based Access Logger" \  
  --set "retention-policy:Free Disk Space Retention Policy" \  
  --hostname host1 --port 1389 --bindDN "cn=Directory Manager" \  
  --bindPassword secret
```

## Managing the Global Index in Entry Balancing Configurations

In an entry balancing configuration, the directory proxy server maintains a global index for DN's as well as key attributes that are frequently used in search operations. This global index allows the proxy server to select the correct backend server set for incoming operations, which avoids broadcasting operations to all backend sets.

This index is preloaded from peer proxies or the backend directory servers when the server starts up, and is updated by certain operations that come through the proxy. For instance, when a new entry is added, the DN of the new entry is added to the index of the proxy server performing the operation.

This section describes how to reload the global index, how to monitor its growth, and how to prime the global index from a peer at start-up.

## Reloading the Global Index

The Directory Proxy Server provides a tool, **reload-index**, which allows you to manually reload the directory proxy server global indexes. You might need to reload the index if the proxy fails to successfully load its global indexes on startup or if the integrity of the index is in question. You can use the tool to reload all configured indexes in the global index, including the RDN index and all attribute indexes, or to reload only those indexes you specify.

The tool schedules an operation to run within the directory proxy server's process. You must supply LDAP connection information so that the tool can communicate with the server through its task interface. Tasks can be scheduled to run immediately or at a later scheduled time. Once scheduled, you can manage the tasks using the **manage-tasks** tool.

For example, to reload all of the indexes within the scope of the **dc=example,dc=com** base DN, run the tool as follows. The task is performed as **cn=Directory Manager** on port 389 of the **localhost** server:

```
reload-index --bindPassword password \  
--baseDN "dc=example,dc=com"
```

The existing index contents are erased before reloading.

To reload the RDN and UID index in the background, so that the existing contents of these indexes can continue to be used, run the command as follows:

```
reload-index --bindPassword password \  
--baseDN "dc=example,dc=com" \  
--index rdn --index uid --background
```

You can force the proxy server to prime from the backend directory server only using the **--fromDS** option, overriding the configuration of the **prime-index-source** property. You can do this on a one off basis if the global index appears to be growing too large. For example, run the command as follows:

```
reload-index --bindPassword password --baseDN "dc=example,dc=com" \  
--fromDS
```

## Monitoring the Size of the Global Index

Over time, stale entries can build up in the global index because proxy servers do not communicate changes to the index with one another. The proxy continues to operate normally in this situation since the global index is only ever a hint at where to find entries.

The rate of this growth is typically very slow since in most environments the key attributes change infrequently. The global index itself is also very compact. However, if the global index starts to fill up the allocated memory, it might be necessary to flush it and reload it. The size of the global index can be monitored over LDAP using the following command:

```
ldapsearch -b "cn=monitor" -D "uid=admin,dc=example,dc=com" \  
-w password "(global-index-current-memory-percent=*)" \  
global-index-current-memory-percent
```

If the global index fills up, the proxy server will continue to operate normally, but it will need to start evicting entries from the index, which will lead to more broadcast searches, reducing the overall throughput of the proxy server.

To reload the index so that it no longer holds stale information, run the **reload-index** command with the **--fromDS** option so that data is loaded from backend directory servers. This should be done during off-peak hours.

## Sizing the Global Index

The directory proxy server includes a tool, **global-index-size**, to help you estimate the size in memory of your global indexes. You can estimate the size of more than one index in a single invocation by providing multiple sets of options. The tool creates its estimate using the following information:

- The number of keys in the index. For example, for the built-in RDN index, the number of keys is the total number of entries in the directory.
- The size of the key, in bytes. For attributes indexes, the key is simply the attribute value. For the built-in RDN index, the key is the RDN directly below the balancing base DN. For example, for the DN **uid=user.0,dc=example,dc=com** under the balancing base DN of **dc=example,dc=com**, the key size is 10 bytes (one byte for every digit in the RDN **uid=user.0**).
- The configured number of keys. This value corresponds to the maximum number of keys you expect in your directory. The configured number of keys is provided in the **index-size** configuration property of the **global-attribute-index** object when you configure an attribute index. For the built-in RDN index, the configured number of keys is provided in the **rdn-index-size** property. If you do not provide a value, the tool assumes that the configured number of keys is the same as the actual number of keys.

For example, run the command as follows to estimate the size of two separate indexes, both with 10,000,000 keys but with differing average key sizes. The configured number of keys is assumed to be equal to the actual number of keys:

```
./bin/global-index-size --numKeys 10000000 \  
--averageKeySize 11 --numKeys 10000000 \  
--averageKeySize 15
```

| Num Keys | : Cfg. Num Keys | : Avg. Key Size | : Est. Memory Size |
|----------|-----------------|-----------------|--------------------|
| -----    | :-----          | :-----          | :-----             |
| 10000000 | : 10000000      | : 11            | : 159 mb           |
| 10000000 | : 10000000      | : 15            | : 197 mb           |



## Priming the Global Index on Start Up

The directory proxy server can prime the global index on startup from the backend directory server or directory proxy server, which resides on the same LAN or subnet. When priming occurs locally, you can avoid WAN bandwidth consumption and reduce the processing load on the directory servers in the topology. You can specify the data sources for the index priming and the order in which priming from these sources occurs.

Use the **prime-index-source** property to specify the sources of data, either **ds**, **proxy** or some combination of the two. The order you specify is the order in which priming from these sources will be attempted. For example, if you specify **prime-index-source:ds,proxy**, priming will first be attempted on the directory server, then on the directory proxy server. In most cases, your directory proxy server should prime its index from an instance of a directory server instead of from a peer proxy server. No matter the server you choose, priming is most efficient if the peer server shares the same location as the proxy server itself.

In some circumstances, such as when the proxy server fronts a Sun Directory Server, you may want to prime the indexes from a peer directory proxy server instance. The **peer-proxy-server** property allows you to specify the set of directory proxy servers that are available to process index priming requests. The **prime-from-proxy-maximum-retry-count** property allows you to specify the times that priming is retried against an alternate peer proxy server.

During setup of an entry balancing topology, when you configure peer proxy servers, root accounts and external servers are automatically set up. The root DN account created is called **cn=IntraProxy User,cn=Root DNs,cn=config**. The external server name is **intra-proxy-<servername>**. If you have already configured peer proxy servers in a topology, use these existing **intra-proxy** accounts to specify the peer proxy server in the entry-balancing request processor.

For example, you can change the entry-balancing request processor so that it loads all indexes at startup from the peer proxy server. If the peer is unavailable, the request processor loads the indexes from the backend directory servers.

```
dsconfig set-request-processor-prop \
--processor-name dc_example_dc_com-eb-req-processor \
--set prime-all-indexes:true --set prime-index-source:proxy \
--set prime-index-source:ds \
--add peer-proxy-server:intra-proxy-host.example.com:3389
```

### Note

Setup creates an **intra-proxy** external server for the proxy server itself. This server should not be chosen as the peer server.

If you do not want to setup using a peer server, you can configure index priming manually by creating an external server, creating a global attribute index, and then changing the entry-balancing request processor to load indexes from this external server. For example, to create an external server of the type Alcatel-Lucent Proxy Server to represent a peer of the proxy server, you would use dsconfig as follows:

```
dsconfig create-external-server \
--server-name intra-proxy-host.example.com:3389 \
```

```
--type alcatel-lucent-proxy-server \  
--set server-host-name:intra-proxy-host \  
--set server-port:338 \  
--set "bind-dn:cn=Directory Manager" \  
--set "password:secret123"
```

Next, create a global attribute index on the uid attribute as follows:

```
dsconfig create-global-attribute-index \  
--processor-name dc_example+dc+com-eb-req-processor \  
--index-name uid --set guaranteed-unique:true
```

Finally, change the entry-balancing request processor to load the indexes at startup from the peer proxy server using **dsconfig set-request-processor-prop** as described above.

## Priming or Reloading the Global Index from Sun Directory Servers

When priming or reloading a global index based on a Sun Directory Server environment, the Sun servers may become overwhelmed and unresponsive because of their method of streaming data. To reduce the impact of priming on these server, you can use the **prime-search-entry-per-second** property. To reduce the impact of reloading these indexes, use the **--searchEntryPerSecond** property of the **reload-index** command. These properties control the rate at which the proxy server accepts search result entries from the backend directory servers.

The following example configures the entry-balancing request processor so that it loads all indexes at startup from the peer proxy server. If the peer is unavailable, the request processor loads the indexes from the backend Sun directory server at a rate of 1000 entries per second.

```
dsconfig set-request-processor-prop \  
--processor-name dc_example_dc_com-eb-req-processor \  
--set prime-all-indexes:true --set prime-index-source:proxy \  
--set prime-index-source:ds --add peer-proxy-server:localhost:238 \  
--set prime-search-entry-per-second 1000
```

To control the rate at which the proxy accepts search result entries from the backend directory server during index reload, you set the **--searchEntryPerSecond** property as follows:

```
reload-index --bindPassword password --baseDN "dc=example,dc=com" \  
--searchEntryPerSecond 1000
```

To find the optimum rate, we recommend starting low, for example, specifying a few thousand search entries per second, and then increasing as necessary.

## Setting Resource Limits

You can set resource limits for the directory proxy server using several global configuration properties. You can specify the following types of limits:

- Specify the maximum number of client connections that can be established at any given time using the **maximum-concurrent-connections** property. If the server already has the maximum number of connections established, then any new connection attempts from any clients will be rejected until an existing connection is closed. The default value of zero indicates that no limit is enforced.
- Specify the maximum number of client connections that can be established at any give time from the same client system using the **maximum-concurrent-connections-per-ip-address** property. If the server already has the maximum number of connections established from a given client, then any new connection attempts from that client will be rejected until an existing connection from that client is closed. The server may continue to accept connections from other clients that have not yet reached this limit. The default value of zero indicates that no limit is enforced.
- Specify the maximum number of client connections that can be established at any given time while authenticated as a particular user with the **maximum-concurrent-connections-per-bind-dn** property. This property applies after the connection is established, because the bind operation to authenticate the user happens after the connection is established rather than during the course of establishing the connection itself. If the maximum number of connections are authenticated as a given user, then any new attempt to authenticate as that user will cause the connection performing the bind to be terminated. Note that this limit applies only to authenticated connections, and will not be enforced for clients that have not authenticated or for clients that have authenticated as the anonymous user. The default value of zero indicates that no limit is enforced.

Any changes to the **maximum-concurrent-connections** and **maximum-concurrent-connections-per-ip-address** properties will take effect only for new connections established after the change is made. Any change to the **maximum-concurrent-connections-per-bind-dn** property will apply only to connections (including existing connections) which perform authentication after the change is made. Existing connections will be allowed to remain established even if that would cause the new limit to be exceeded.

## Monitoring Directory Proxy Server

While the directory proxy server is running, it generates a significant amount of information available through monitor entries. This section contains information about the following:

- Monitoring Server Status Using the status Tool
- About the Monitor Entries
- Using the Monitoring Interfaces
- Monitoring with JMX

## Monitoring Server Status Using the status Tool

UnboundID Directory Proxy Server provides a **status** tool that provides basic server status information, including version, connection handlers, a table of LDAP external servers, and the percent of the global index that is used.

For example, running the status tool returns the following information:

```
$ bin/status

>>>> Specify LDAP connection parameters

Administrator user bind DN [cn=Directory Manager]:
Password for user 'cn=Directory Manager':

--- Server Status ---

Server Run Status:    Started 07/Jan/2011:10:59:52.000 -0600
Operational Status:   Available
Open Connections:     4
Max Connections:      8
Total Connections:    25

--- Server Details ---

Host Name:            example
Administrative Users: cn=Directory Manager
Installation Path:    /path/to/UnboundID-Proxy
Version:              UnboundID-Directory Proxy Server 3.1.0.0
Java Version:         1.6.0_22

--- Connection Handlers ---

Address:Port : Protocol : State
-----:-----:-----
0.0.0.0:1689 : JMX       : Disabled
0.0.0.0:636  : LDAPS     : Disabled
0.0.0.0:9389 : LDAP      : Enabled

--- LDAP External Servers ---

Server          : Status      : Score : LB Algorithm
-----:-----:-----:-----
localhost:389   : Available : 10     : dc_example_dc_com-round-robin
localhost:1389  : Available : 10     : dc_example_dc_com-round-robin

--- LDAP External Server Op Counts ---

Server          : Add : Bind:Compare:Delete:Modify:Mod DN:Search : All
-----:-----:-----:-----:-----:-----:-----:-----
localhost:11389: 0   : 0   : 0       : 0       : 0       : 0       : 1249 : 1249
localhost:12389: 0   : 0   : 0       : 0       : 0       : 0       : 494  : 494

--- Entry Balancing Request Processors ---

Base DN          : Global Index % Used
-----:-----
ou=people,dc=example,dc=com : 33
```

```
--- Global Index Stats for ou=people,dc=example,dc=com ---
--- Global Index Stats for ou=people,dc=example,dc=com ---
Index : Total Bytes : Key Bytes : Keys      : Size (# Keys) : Inserted :
Removed : Replaced: Hits : Misses : Discarded : Duplicates
-----:-----:-----:-----:-----:-----:-----
rdn   : 30667304    : 14888906 : 1000001 : 3464494        : 1000001 :
0     : 0             : 0        : 0        : 0              : 0
uid   : 26523480    : 10888902 : 1000001 : 3464494        : 1000001 :
0     : 0             : 3583     : 0        : 0              : 0

--- Operation Processing Time ---
Op Type   : Total Ops : Avg Resp Time (ms)
-----:-----:-----
Add       : 0        : 0.0
Bind      : 0        : 0.0
Compare   : 0        : 0.0
Delete    : 0        : 0.0
Modify    : 0        : 0.0
Modify DN : 0        : 0.0
Search    : 3583     : 117.58
All       : 3583     : 117.58

--- Work Queue ---
      : Recent : Average : Maximum
-----:-----:-----:-----
Queue Size : 0      : 0       : 1
% Busy     : 0      : 1       : 19
```

## About the Monitor Entries

Monitor entries are available over LDAP in the `cn=monitor` subtree. The types of monitor entries that are available include:

- General Monitor Entry (**cn=monitor**) – Provides a basic set of general information about the server.
- Active Operations Monitor Entry (**cn=Active Operations,cn=monitor**) – Provides information about all operations currently in progress in the server.
- Backend Monitor Entries (**cn={id} Backend,cn=monitor**) – Provides information about the backend, including the number of entries, the base DN(s), and whether it is private.
- Client Connections Monitor Entry (**cn=Client Connections,cn=monitor**) – Provides information about all connections currently established to the server.

- Connection Handler Monitor Entry (**cn={name},cn=monitor**) – Provides information about the configuration of each connection handler and the client connections established to it.
- Disk Space Usage Monitor Entry (**cn=Disk Space Usage,cn=monitor**) – Provides information about the amount of usable disk space available to server components.
- Entry-Balancing Monitor Entry (**cn=Entry-Balancing Request Processor,cn=monitor**) – Provides information about the entry-balancing request processor, including information about the number of specific operations forwarded and processed as well as information about the memory used by the global index.
- JVM Memory Usage Monitor Entry (**cn=JVM Memory Usage,cn=monitor**) – Provides information about garbage collection activity, the amount of memory available to the server, and the amount of memory consumed by various server components.
- JVM Stack Trace Monitor Entry (**cn=JVM Stack Trace,cn=monitor**) – Provides a stack trace of all threads in the JVM.
- LDAP External Server Entry (**cn=LDAP external server {server\_name} for load-balancing algorithm {algorithm\_name}, cn=monitor**) – Provides information about the LDAP external server, including information about the server health, the operations it processed, the success and failure of each operation, and the common connection pool use.
- LDAP Statistics Monitor Entries (**cn={name} Statistics,cn=monitor**) – Provides information about the number of each type of operation requested and bytes transferred over the connection handler.
- Processing Time Histogram Monitor Entry (**cn=Processing Time Histogram,cn=monitor**) – Provides information about the number of percent of operations that completed in various response time categories.
- System Information Monitor Entry (**cn=System Information,cn=monitor**) – Provides information about the underlying JVM and system.
- Version Monitor Entry (**cn=Version,cn=monitor**) – Provides information about the directory proxy server version.
- Work Queue Monitor Entry (**cn=Work Queue,cn=monitor**) – Provides information about the state of the directory proxy server work queue, including the number of operations waiting on worker threads and the number of operations that have been rejected because the queue became full.

## Using the Monitoring Interfaces

UnboundID Directory Proxy Server exposes its monitoring information under the `cn=monitor` entry and provides interfaces through JMX, the Directory Proxy Server Management Console, and over LDAP.

## Monitoring with the Directory Proxy Server Management Console

UnboundID has developed a graphical web console for administrators to configure the directory proxy server. The console also provides a monitoring component that accesses the server's `cn=monitor` entry.

### To View the Monitor Dashboard

1. Start the directory proxy server.

```
$ /UnboundID-proxy/bin/start-proxy
```

2. Open a browser to `http://hostname:8080/proxyconsole/`. The application server listens on port 8080 for HTTP requests.
3. Type the root user DN (`cn=Directory Manager`) and password, and then click Login.
4. Click Monitor Dashboard.
5. View the monitoring information on the dashboard.

### Accessing the Server Monitors

The Directory Proxy Server Management Console accesses the monitor entry under `cn=monitor` and tracks the various system components.

1. On the Directory Proxy Server Management Console, click Server Monitors.
2. View the Server Monitors.

### Accessing the Processing Time Histogram

UnboundID Directory Proxy Server provides a processing time histogram that classifies operation response time into user-defined buckets. The histogram tracks the processing on a per operation basis and as a percentage of the overall processing time for all operations. It also provides statistics for each operation type (add, bind, compare, delete, modify, modify DN, search).

1. On the Directory Proxy Server Management Console, click Server Monitors.
2. Click Processing Time Histogram. Other monitor entries can be accessed in similar ways.

## Monitoring with JMX

UnboundID Directory Proxy Server supports monitoring the JVM through a Java Management Extensions (JMX) management agent, which can be accessed using JConsole. JConsole provides JVM performance and resource utilization information for applications running Java.

You can monitor generic metrics exposed by the JVM itself, including memory pools, threads, loaded classes, and MBeans.

## Running JConsole

Before you can access JConsole, you must configure and enable the JMX Connection Handler for the directory proxy server using the `dsconfig` tool.

To invoke the JConsole executable, type `jconsole` on the command line. If `JDK_HOME` is not set in your path, you can access JConsole in the `bin` directory of the `JDK_HOME` path.

### To Run JConsole

1. Use JConsole to open the Java Monitoring and Management Console. You can also run JConsole to monitor a specific process ID for your application: `jconsole PID`. Or you can run JConsole remotely using: `jconsole hostname:port`.

```
$ jconsole
```

2. On the Java Monitoring & Management Console, click Local Process, and then click the PID corresponding to the directory proxy server.
3. Review the resource monitoring information.

## Monitoring Directory Proxy Server Using JConsole

You can set up JConsole to monitor UnboundID Directory Proxy Server using a remote process. Make sure to enable the JMX Connection Handler and to assign JMX read, write, and notify privileges to a regular user account. Do not use a root user account, as this would pose a security risk.

### To Monitor the Directory Proxy Server using JConsole

1. Start the directory proxy server.

```
$ bin/start-proxy
```

2. On the Java Monitoring & Management Console, click Remote Process, and enter the following JMX URL using the host and port of your directory proxy server.

```
service:jmx:rmi:///jndi/rmi://localhost:1689/com.unboundid.directory.server.protocols.jmx.client-unknown
```

You may need to change `localhost` to reflect the address of the directory proxy server. The port number, in this example `1689`, corresponds to the port on which the JMX connection handler is listening.



3. In the Username and Password fields, type the bind DN and password for a user that has at least the `jmx-read` privilege. To receive JMX notifications, this user also needs the `jmx-notify` privilege. Click Connect.
4. Click `com.unboundid.directory.server` and expand the `rootdse` node of the `cn-monitor` sub-node.
5. Click a monitoring entry. For example, click the LDAP Connection Handler entry.

## Monitoring Over LDAP

UnboundID Directory Proxy Server exposes a majority of directory proxy server information under the `cn=monitor` entry. You can access these entries over LDAP using the `ldapsearch` tool.

```
$ bin/ldapsearch --hostname server1.example.com --port 1389 \  
  --bindDN "cn=Directory Manager" --bindPassword secret \  
  --baseDN "cn=monitor" "(objectclass=*)" "
```

For more information on each monitor entry, see “About the Monitor Entries” on page 117.

## Monitoring Using SNMP

The UnboundID Directory Proxy Server supports realtime monitoring using the Simple Network Management Protocol (SNMP). The directory proxy server provides an embedded SNMPv3 sub-agent plug-in that, when enabled, sets up the server as a managed device and exchanges monitoring information with a Master Agent based on the AgentX protocol.

### About SNMP

Established in 1988, SNMP is a widespread industry standard for realtime network monitoring. SNMP has a simple manager-agent model, in which a manager or client application sends information requests to an SNMP agent running as a daemon process on a managed device. A managed device can be a directory, proxy, synchronization, or host server plus any other networked component. The SNMP agent also issues a change-of-state notification, called a *trap*, that alerts the client about certain events or conditions that occur in the server. The manager’s trap handler can then respond to the trap by generating an admin alert, issuing an email, or taking some other specific action.

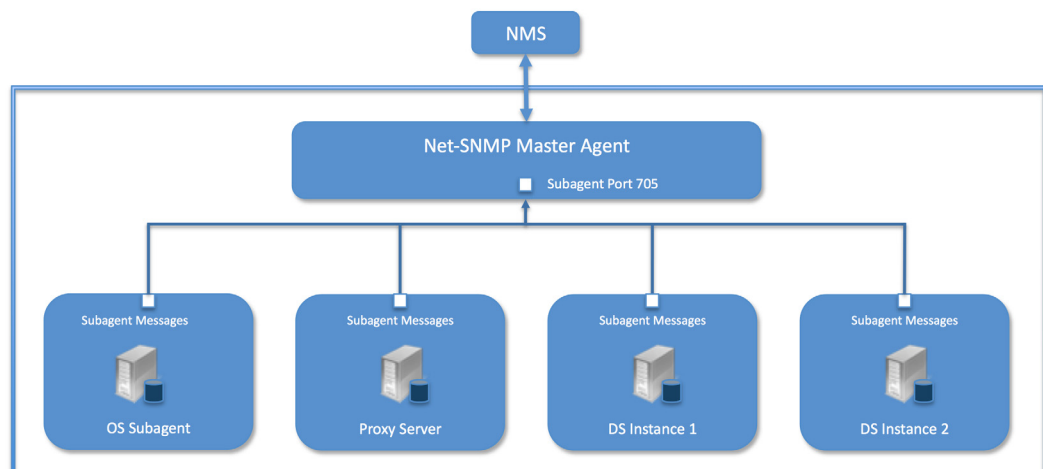
The manager and agent exchange information using a schema-like definition file called a Management Information Base (MIB). A MIB is a flat ASN.1-formatted text file that specifies the types of name-value definitions that can be exchanged between the manager and agent. MIBs contain an organized tree structure of variables, called *MIB objects*, which can be either a single scalar attribute or tabular objects that group similar variables. Each MIB attribute is uniquely identified by a hierarchically ordered number, separated by periods, called an *object identifier (OID)*. Agents and the management framework consult these MIBs to determine which variables can be retrieved for the SNMP request messages.

## Directory Proxy Server Implementation

In a typical SNMP deployment, many production environments use a Network Management System (NMS) for a unified monitoring and administrative view of all SNMP-enabled devices. The NMS communicates with a Master Agent, whose main responsibility is to translate the SNMP protocol messages and multiplex any request messages to the subagent on each managed device (for example, directory server instance, directory proxy server, or OS Sub-agent). The Master Agent also processes responses or traps from the agents. Many vendors provide commercial NMS systems, such as, Alcatel-Lucent (Omnivista EMS), HP (Open-View), IBM-Tivoli (Netview), Oracle-Sun (Solstice Enterprise Manager), and others. Specific discussion on integrating an SNMP deployment on an NMS system is beyond the scope of this chapter. Consult with your NMS system for specific information.

The UnboundID Directory Proxy Server contains an SNMP subagent plug-in that connects to a Net-SNMP master agent over TCP. The main configuration properties of the plug-in are the address and port of the master agent, which default to localhost and port 705, respectively. When the plugin is initialized, it creates an AgentX subagent and a Managed Object Server, and then registers as a MIBServer with the directory proxy server instance. Once the plug-in's startup method is called, it starts a session thread with the master agent. Whenever the connection is lost, the sub-agent automatically attempts to reconnect with the master agent. The directory proxy server's SNMP subagent plug-in only transmits read-only values for polling or trap purposes (INFORMS are not supported). Thus, SNMP management applications cannot perform actions on the server on its own or by means of an NMS system.

**FIGURE 6-1. Example SNMP Deployment**



One important note is that the UnboundID Directory Proxy Server was designed to interface with a Net-SNMP (version 5.3.2.2 or later) master agent implementation with AgentX over TCP enabled. Many operating systems provide their own Net-SNMP module, such as the System Management Agent (SMA) on Solaris or OpenSolaris. However, SMA disables some features present in the Net-SNMP package and only enables AgentX over UNIX Domain Sockets, which cannot be supported by Java. If your operating system has a native Net-SNMP

master agent that only enables UNIX Domain Sockets, you must download and install a separate Net-SNMP binary from its web site.

## Configuring SNMP

The UnboundID Directory Proxy Server requires a Net-SNMP (version 5.3.2.2 or later) implementation that supports AgentX over TCP, since UNIX Domain Sockets does not support Java. We recommend that you download the latest release of Net-SNMP. Also, follow your company's security policies to change the community names of your SNMP configuration files.

Because all server instances provide information for a common set of MIBs, each server instance provides its information under a unique SNMPv3 context name, equal to the server instance name. The server instance name is defined in the Global Configuration, and is constructed from the host name and the server LDAP port by default. Consequently, information must be requested using SNMPv3, specifying the context name that pertains to the desired server instance. This context name is limited to 30 characters or less. Any context name longer than 30 characters will result in an error message.

---

**Note**

The directory proxy server supports SNMPv3, and only SNMPv3 can access the MIBs. For systems that implement SNMP v1 and v2c, Net-SNMP provides a proxy function to route requests in one version of SNMP to an agent using a different SNMP version.

---

## To Configure SNMP

1. Enable the directory proxy server's SNMP plug-in using the **dsconfig** tool. Make sure to specify the address and port of the SNMP master agent. On each directory proxy server instance, enable the SNMP sub-agent. Note that the SNMPv3 context name is limited to 30 bytes maximum. If the default dynamically constructed instance name is greater than 30 bytes, there will be an error when attempting to enable the plug-in.

```
$ bin/dsconfig set-plugin-prop --plugin-name "SNMP Subagent" \
--set enabled:true --set agentx-address:localhost --set agentx-port:705 \
--set session-timeout:5s --set connect-retry-max-wait:10s
```

2. Set the SNMP Subagent Alert Handler so that the sub-agent will send traps.

```
$ bin/dsconfig set-alert-handler-prop \
--handler-name "SNMP Subagent Alert Handler" --set enabled:true
```

3. View the error log. You will see a message that the master agent is not connected, because it is not yet online.

```
The SNMP sub-agent was unable to connect to the master agent at
localhost/705: Timeout
```

4. Install Net-SNMP (version 5.3.2.2 or later) on your machine. This step will vary depending on the OS. On Linux, the SNMP install should be compiled to allow AgentX over TCP.

You can verify that it is compiled with TCP by installing the `net-snmp-devel` package and running this command:

```
$ net-snmp-config --configure-options
```

The output of that command should also NOT contain `--enable-agentx-dom-sock-only`.

5. Configure the SNMP agent configuration file, `snmpd.conf`, which is located in `/usr/local/share/snmp/snmpd.conf`. Add the directive to run the agent as an AgentX Master Agent:

```
master agentx
agentXSocket tcp:localhost:705
```

Note that the use of `localhost` means that only subagents running on the same host can connect to the master agent. This requirement is necessary since there are no security mechanisms in the AgentX protocol.

6. Add the trap directive to send SNMPv2 traps to localhost with the community name, *public*.

```
trap2sink localhost public
```

7. Create an SNMPv3 user.

```
$ net-snmp-config --create-snmpv3-user -A password snmpuser
```

8. Start the `snmpd` daemon and after a few seconds you should see the following message in the directory proxy server error log:

```
The SNMP sub-agent connected successfully to the master agent at
localhost:705. The SNMP context name is host.example.com:389
```

At this point SNMP is up and running.

9. Set up a trap client to see the alerts that are generated by the directory proxy server. Create a config file in `/tmp/snmptrapd.conf` and add the directive below to it. The directive specifies that the trap client can process traps using the `public` community string, and can log and trigger executable actions.

```
authcommunity log, execute public
```

10. Then, run the trap client using the `snmptrapd` command. The following example specifies that the command should not create a new process using `fork()` from the calling shell (`-f`), do not read any configuration files (`-C`) except the one specified with the `-c` option, print to standard output (`-Lo`), and then specify that debugging output should specify "No Such User" (`-Dusm`).

```
$ snmptrapd -f -C -c /tmp/snmptrapd.conf -Lo -Dusm
```

11. Install the MIB definitions for the Net-SNMP client tools.

```
$ cp resource/mib/* /usr/share/snmp/mibs
```

12. Run the Net-SNMP client tools to test the feature. The following options are required: `-v <SNMP version>`, `-u <user name>`, `-A <user password>`, `-l <security level>`, `-n <context name (instance name)>`. The `-m all` option loads all MIBs in the default MIB directory in `/usr/share/snmp/mibs` so that MIB names can be used in place of numeric OIDs.

```
$ snmpget -v 3 -u snmpuser -A password -l authNoPriv \  
-n host.example.com:389 -m all localhost localDBBackendCount.0
```

```
$ snmpwalk -v 3 -u snmpuser -A password -l authNoPriv \  
-n host.example.com:389 -m all localhost systemStatus
```

13. If you want alerts sent from the SNMP Subagent through the Net-SNMP Master Agent and onwards, you must enable the SNMP Subagent Alert Handler. The *SNMP Alert Handler* is used in deployments that do not enable the Subagent.

```
$ bin/dsconfig set-alert-handler-prop \  
--hostname host2 --port 1389 \  
--bindDN "uid=admin,dc=example,dc=com" --bindPassword secret \  
--handler-name "SNMP Subagent Alert Handler" \  
--set enabled:true \  
--set server-host-name:host2 \  
--set server-port:162 \  
--set community-name:public \  
--no-prompt
```

## MIBS

The directory proxy server provides SMIV2-compliant MIB definitions (RFC 2578, 2579, 2580) for distinct monitoring statistics. These MIB definitions are to be found in text files under `config/mib` in the server root directory. Each MIB provides managed object tables for each specific SNMP management information as follows:

- **LDAP Remote Server MIB.** Provides information related to the health and status of the LDAP servers that the proxy server connects to, and statistics about the operations invoked by the proxy server on those LDAP servers.
- **LDAP Statistics MIB.** Provides a collection of connection-oriented performance data that is based on a connection handler in the directory proxy server. A server typically contains only one connection handler and therefore supplies only one table entry.
- **Local DB Backend MIB.** Provides key metrics related to the state of the local database backends contained in the server.
- **Processing Time MIB.** Provides a collection of key performance data related to the processing time of operations broken down by several criteria but reported as a single aggregated data set.
- **Replication MIB.** Provides key metrics related to the current state of replication, which can help diagnose how much outstanding work replication may have to do.

- **System Status MIB.** Provides a set of critical metrics for determining the status and health of the system in relation to its work load.

## MIB Table Reference

The following tables summarize the available monitoring statistics for each MIB available on the directory server and the directory proxy server:

**TABLE 6-1. LDAP Remote Server MIB for the Directory Proxy Server**

| Name                  | Description                                                       |
|-----------------------|-------------------------------------------------------------------|
| ldapRemoteServerCount | Scalar. Number of LDAP servers that the proxy server connects to. |

**TABLE 6-2. LDAP Remote Server Entry Table for the Directory Proxy Server**

| Name                             | Description                                                                                                                                                                                                                                                                  |
|----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| remoteServerID                   | LDAP remote server ID.                                                                                                                                                                                                                                                       |
| serverAddress                    | Address of the LDAP remote server.                                                                                                                                                                                                                                           |
| serverPort                       | Port of the LDAP remote server.                                                                                                                                                                                                                                              |
| healthState                      | Health state of the LDAP remote server. Possible values are 'available', 'degraded' and 'unavailable'.                                                                                                                                                                       |
| healthStateDynamicallyDetermined | Provides information about how the health state of the LDAP remote server has been determined. The value 'true' indicates that the health state was dynamically determined. The value 'false' indicates that the health state was explicitly configured in the proxy server. |
| totalSuccessfulOperations        | Total number of successful operations invoked by the proxy server against the LDAP remote server.                                                                                                                                                                            |
| totalUnsuccessfulOperations      | Total number of failed operations invoked by the proxy server against the LDAP remote server                                                                                                                                                                                 |

**TABLE 6-3. LDAP Statistics MIB**

| Name                   | Description                                          |
|------------------------|------------------------------------------------------|
| connectionHandlerCount | Scalar. Number of connection handlers in the server. |

**TABLE 6-4. LDAP Statistics MIB Entry Table**

| Name                   | Description                                                                            |
|------------------------|----------------------------------------------------------------------------------------|
| connectionHandlerID    | Unique connection handler ID.                                                          |
| connectionsEstablished | Number of connections established since the associated connection handler was started. |
| connectionsClosed      | Number of connections closed since the associated connection handler was started.      |

**TABLE 6-4. LDAP Statistics MIB Entry Table**

| <b>Name</b>         | <b>Description</b>                                                                              |
|---------------------|-------------------------------------------------------------------------------------------------|
| bytesRead           | Number of bytes read from clients since the associated connection handler was started.          |
| bytesWritten        | Number of bytes written to clients since the associated connection handler was started.         |
| ldapMessagesRead    | Number of LDAP messages read from clients since the associated connection handler was started.  |
| ldapMessagesWritten | Number of LDAP messages written to clients since the associated connection handler was started. |
| operationsAbandoned | Number of operations abandoned since the associated connection handler was started.             |
| operationsInitiated | Number of operations initiated since the associated connection handler was started.             |
| operationsCompleted | Number of operations completed since the associated connection handler was started.             |
| abandonRequests     | Number of abandon requests from clients since the associated connection handler was started.    |
| addRequests         | Number of add requests from clients since the associated connection handler was started.        |
| addResponses        | Number of add responses to clients since the associated connection handler was started.         |
| bindRequests        | Number of bind requests from clients since the associated connection handler was started.       |
| bindResponses       | Number of bind responses to clients since the associated connection handler was started.        |
| compareRequests     | Number of compare requests from clients since the associated connection handler was started.    |
| compareResponses    | Number of compare responses to clients since the associated connection handler was started.     |
| deleteRequests      | Number of delete requests from clients since the associated connection handler was started.     |
| deleteResponses     | Number of delete responses to clients since the associated connection handler was started.      |
| extendedRequests    | Number of extended requests from clients since the associated connection handler was started.   |
| extendedResponses   | Number of extended responses to clients since the associated connection handler was started.    |
| modifyRequests      | Number of modify requests from clients since the associated connection handler was started.     |
| modifyResponses     | Number of modify responses to clients since the associated connection handler was started.      |
| modifyDNRequests    | Number of modify DN requests from clients since the associated connection handler was started.  |
| modifyDNResponses   | Number of modify DN responses to clients since the associated connection handler was started.   |
| searchRequests      | Number of search requests from clients since the associated connection handler was started.     |

**TABLE 6-4. LDAP Statistics MIB Entry Table**

| Name                   | Description                                                                                             |
|------------------------|---------------------------------------------------------------------------------------------------------|
| searchResultEntries    | Number of search result entries sent to clients since the associated connection handler was started.    |
| searchResultReferences | Number of search result references sent to clients since the associated connection handler was started. |
| searchResultsDones     | Number of search result done responses to clients since the associated connection handler was started.  |
| unbindRequests         | Number of unbind requests from clients since the associated connection handler was started.             |

**TABLE 6-5. Local DB Backend MIB**

| Name                | Description                                                  |
|---------------------|--------------------------------------------------------------|
| localDBBackendCount | Scalar. Number of local DB backends contained in the server. |

**TABLE 6-6. Local DB Backend Entry Table**

| Name             | Description                                                                                  |
|------------------|----------------------------------------------------------------------------------------------|
| backendID        | Unique ID for the associated backend. Typically, the ID is "userRoot" for user data entries. |
| numberOfEntries  | Total number of entries in the associated backend represented as a string.                   |
| percentCacheFull | Percentage of the maximum database cache size that is currently in use.                      |

**TABLE 6-7. Processing Time Entry Table**

| Name                      | Description                                                                              |
|---------------------------|------------------------------------------------------------------------------------------|
| allCompletedOps           | Total number of all completed operations processed. Syntax: counter64.                   |
| allTotalOpTime            | Total time (milliseconds) spent processing operations of any type in the server.         |
| allAverageResponseTime    | Average response time (milliseconds) of operations of all types performed in the server. |
| searchCompletedOps        | Total number of search operations that have been performed in the server.                |
| searchTotalOpTime         | Total time (milliseconds) spent processing search operations in the server.              |
| searchAverageResponseTime | Average response time (milliseconds) of search operations performed in the server.       |
| modifyCompletedOps        | Total number of modify operations that have been performed in the server..               |
| modifyTotalOpTime         | Total time (milliseconds) spent processing modify operations in the server.              |
| modifyAverageResponseTime | Average response time (milliseconds) of modify operations performed in the server.       |
| addCompletedOps           | Total number of add operations that have been performed in the server.                   |
| addTotalOpTime            | Total time (milliseconds) spent processing add operations in the server.                 |



**TABLE 6-7. Processing Time Entry Table**

| Name                        | Description                                                                         |
|-----------------------------|-------------------------------------------------------------------------------------|
| addAverageResponseTime      | Average response time (milliseconds) of add operations performed in the server.     |
| deleteCompletedOps          | Total number of delete operations that have been performed in the server.           |
| deleteTotalOpTime           | Total time (milliseconds) spent processing delete operations in the server.         |
| deleteAverageResponseTime   | Average response time (milliseconds) of delete operations performed in the server.  |
| modifyDNCompletedOps        | Total number of modDN operations that have been performed in the server.            |
| modifyDNTotalOpTime         | Total time (milliseconds) spent processing modDN operations in the server.          |
| modifyDNAverageReponse-Time | Average response time (milliseconds) of modDN operations performed in the server.   |
| bindCompletedOps            | Total number of bind operations that have been performed in the server..            |
| bindTotalOpTime             | Total time (milliseconds) spent processing bind operations in the server.           |
| bindAverageResponseTime     | Average response time (milliseconds) of bind operations performed in the server.    |
| compareCompletedOps         | Total number of compare operations that have been performed in the server.          |
| compareTotalOpTime          | Total time (milliseconds) spent processing compare operations in the server.        |
| compareAverageResponse-Time | Average response time (milliseconds) of compare operations performed in the server. |

**TABLE 6-8. Replication MIB Table**

| Name                    | Description                                                                                                                                                                                       |
|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| replicaCount            | Scalar. Number of replicas contained in the server.                                                                                                                                               |
| replicationDomainID     | Replication domain ID for the replica.                                                                                                                                                            |
| oldestMissingChangeDate | The date of the oldest missing change for the replica in Generalized Time form (yyyyMMddHHmmss.SSS'Z'), or the empty string if this information is not available or there are no missing changes. |
| numberOfMissingChanges  | Number of missing changes for the replica, or -1 if this information is not available                                                                                                             |

**TABLE 6-9. System Status Entry Table**

| Name                            | Description                                                                                               |
|---------------------------------|-----------------------------------------------------------------------------------------------------------|
| totalConnections                | Total number of client connections that have been established since startup.                              |
| currentNumConnections           | Number of client connections currently established to the server.                                         |
| maxWorkQueueSize                | Maximum number of operations observed in the directory proxy server work queue at any time since startup. |
| averageWorkQueueSize            | Average number of operations observed in the directory proxy server work queue since startup.             |
| maxWorkerThreadPercent-Busy     | Maximum percentage of the time over any interval that the worker threads have spent processing requests.  |
| averageWorkerThreadPercent-Busy | Average percentage of the time since startup that the worker threads have spent processing requests.      |

## SNMP Traps

The directory proxy server generates an extensive set of SNMP traps for event monitoring. The traps display the severity, description, name, OID, and summary as follows:

**TABLE 6-10. SNMP Traps**

| SNMP Trap OID               | SNMP Traps                               | Description                                                                                                                                                                   |
|-----------------------------|------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.3.6.1.4.1.30221.2.10.3.1  | ACCESS_CONTROL_DISABLED                  | Indicates that the access control evaluation has been disabled.                                                                                                               |
| 1.3.6.1.4.1.30221.2.10.3.2  | ACCESS_CONTROL_ENABLED                   | Indicates that the access control evaluation has been enabled.                                                                                                                |
| 1.3.6.1.4.1.30221.2.10.3.3  | ACCESS_CONTROL_PARSE_FAILURE             | Indicates that an access control rule could not be parsed.                                                                                                                    |
| 1.3.6.1.4.1.30221.2.10.3.4  | CANNOT_ACQUIRE_SHARED_BACKEND_LOCK       | Indicates that the server cannot acquire a shared lock for a backend.                                                                                                         |
| 1.3.6.1.4.1.30221.2.10.3.5  | CANNOT_COPY_SCHEMA_FILES                 | Indicates that the server cannot copy schema files when making a schema update.                                                                                               |
| 1.3.6.1.4.1.30221.2.10.3.6  | CANNOT_DECODE_ENTRY                      | Indicates that the server cannot decode an entry stored in a backend.                                                                                                         |
| 1.3.6.1.4.1.30221.2.10.3.7  | CANNOT_FIND_RECURRING_TASK               | Indicates that the server cannot schedule the next iteration of a recurring task because the definition could not be found.                                                   |
| 1.3.6.1.4.1.30221.2.10.3.8  | CANNOT_REGISTER_BACKEND                  | Indicates that the server cannot register a backend.                                                                                                                          |
| 1.3.6.1.4.1.30221.2.10.3.9  | CANNOT_RELEASE_SHARED_BACKEND_LOCK       | Indicates that the server cannot release a shared lock for a backend.                                                                                                         |
| 1.3.6.1.4.1.30221.2.10.3.10 | CANNOT_RENAME_CURRENT_TASK_FILE          | Indicates that the server cannot rename the current task backing file.                                                                                                        |
| 1.3.6.1.4.1.30221.2.10.3.11 | CANNOT_RENAME_NEW_TASK_FILE              | Indicates that the server cannot rename the new task backing file.                                                                                                            |
| 1.3.6.1.4.1.30221.2.10.3.12 | CANNOT_RESTORE_BACKUP                    | Indicates that the server cannot restore a backup.                                                                                                                            |
| 1.3.6.1.4.1.30221.2.10.3.13 | CANNOT_SCHEDULE_RECURRING_TASK_ITERATION | Indicates that the server cannot schedule a new iteration of a recurring task.                                                                                                |
| 1.3.6.1.4.1.30221.2.10.3.14 | CANNOT_WRITE_CONFIGURATION               | Indicates that the server cannot write its configuration to disk.                                                                                                             |
| 1.3.6.1.4.1.30221.2.10.3.15 | CANNOT_WRITE_NEW_SCHEMA_FILES            | Indicates that the server cannot write new schema files when making a schema update.                                                                                          |
| 1.3.6.1.4.1.30221.2.10.3.16 | CANNOT_WRITE_TASK_BACKING_FILE           | Indicates that the server cannot write its task backing file.                                                                                                                 |
| 1.3.6.1.4.1.30221.2.10.3.17 | DUPLICATE_ALERTS_SUPPRESSED              | Indicates that the server has suppressed one or more duplicate alert messages.                                                                                                |
| 1.3.6.1.4.1.30221.2.10.3.18 | ENTERING_LOCKDOWN_MODE                   | Indicates that the server is entering lockdown mode.                                                                                                                          |
| 1.3.6.1.4.1.30221.2.10.3.19 | EXTERNAL_CONFIG_FILE_EDIT_HANDLED        | Indicates that the server has detected an external modification to its configuration file with the server online but that change was successfully copied to another location. |
| 1.3.6.1.4.1.30221.2.10.3.20 | EXTERNAL_CONFIG_FILE_EDIT_LOST           | Indicates that the server has detected an external modification to its configuration file but that change was lost.                                                           |
| 1.3.6.1.4.1.30221.2.10.3.21 | INVALID_PRIVILEGE                        | Indicates that a user has been configured with an invalid privilege.                                                                                                          |

TABLE 6-10. SNMP Traps

| SNMP Trap OID               | SNMP Traps                                   | Description                                                                                                                                       |
|-----------------------------|----------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.3.6.1.4.1.30221.2.10.3.22 | JE_RECOVERY_REQUIRED                         | Indicates that a Oracle Berkeley DB JE backend may need to be restarted or run recovery.                                                          |
| 1.3.6.1.4.1.30221.2.10.3.23 | LDAP_CONNECTION_HANDLER_CANNOT_LISTEN        | Indicates that an LDAP connection handler has been disabled because an error occurred while attempting to begin listening for client connections. |
| 1.3.6.1.4.1.30221.2.10.3.24 | LDAP_CONNECTION_HANDLER_CONSECUTIVE_FAILURES | Indicates that an LDAP connection handler has been disabled due to multiple consecutive failures.                                                 |
| 1.3.6.1.4.1.30221.2.10.3.25 | LDAP_CONNECTION_HANDLER_UNCAUGHT_ERROR       | Indicates that an LDAP connection handler has been disabled due to an uncaught error.                                                             |
| 1.3.6.1.4.1.30221.2.10.3.26 | LDIF_BACKEND_CANNOT_WRITE                    | Indicates that an LDIF backend was unable to write an updated copy of its data to disk.                                                           |
| 1.3.6.1.4.1.30221.2.10.3.27 | LDIF_CONNECTION_HANDLER_PARSE_ERROR          | Indicates that an LDIF connection handler encountered an error while reading the contents of the LDIF file.                                       |
| 1.3.6.1.4.1.30221.2.10.3.28 | LDIF_CONNECTION_HANDLER_IO_ERROR             | Indicates that an LDIF connection handler encountered an I/O error that prevented it from performing its processing.                              |
| 1.3.6.1.4.1.30221.2.10.3.29 | LEAVING_LOCKDOWN_MODE                        | Indicates that the server is leaving lockdown mode.                                                                                               |
| 1.3.6.1.4.1.30221.2.10.3.30 | LOGGING_ERROR                                | Indicates that the server has encountered an error while attempting to log a message                                                              |
| 1.3.6.1.4.1.30221.2.10.3.31 | REPLICATION_UNRESOLVED_CONFLICT              | Indicates that the server has detected a replication conflict that could not be resolved.                                                         |
| 1.3.6.1.4.1.30221.2.10.3.32 | SERVER_SHUTTING_DOWN                         | Indicates the the server has begun the shutdown process.                                                                                          |
| 1.3.6.1.4.1.30221.2.10.3.32 | SERVER_STARTED                               | Indicates that the server has been started.                                                                                                       |
| 1.3.6.1.4.1.30221.2.10.3.34 | UNCAUGHT_EXCEPTION                           | Indicates that the server has detected an uncaught exception.                                                                                     |
| 1.3.6.1.4.1.30221.2.10.3.35 | UNIQUE_ATTRIBUTE_SYNC_CONFLICT               | Indicates that a unique attribute conflict has been generated through synchronization.                                                            |
| 1.3.6.1.4.1.30221.2.10.3.36 | UNIQUE_ATTRIBUTE_SYNC_ERROR                  | Indicates that an error occurred while attempting to detect unique attribute conflicts during synchronization.                                    |
| 1.3.6.1.4.1.30221.2.10.3.37 | USER_DEFINED_ERROR                           | Indicates that an error alert notification was generated from third-party code.                                                                   |
| 1.3.6.1.4.1.30221.2.10.3.38 | USER_DEFINED_FATAL                           | Indicates that a fatal error alert notification was generated from third-party code..                                                             |
| 1.3.6.1.4.1.30221.2.10.3.39 | USER_DEFINED_INFO                            | Indicates that an information alert notification was generated from third-party code.                                                             |
| 1.3.6.1.4.1.30221.2.10.3.40 | USER_DEFINED_WARNING                         | Indicates that a warning alert notification was generated from third-party code.                                                                  |
| 1.3.6.1.4.1.30221.2.10.3.41 | WORKER_THREAD_CAUGHT_ERROR                   | Indicates that a worker thread caught an unexpected error.                                                                                        |
| 1.3.6.1.4.1.30221.2.10.3.42 | WORK_QUEUE_FULL                              | Indicates that the work queue has reached its maximum capacity and has begun rejecting requests.                                                  |
| 1.3.6.1.4.1.30221.2.10.3.43 | WORK_QUEUE_NO_THREADS_REMAINING              | Indicates that the work queue does not have any threads left.                                                                                     |
| 1.3.6.1.4.1.30221.2.10.3.43 | Unused                                       |                                                                                                                                                   |

TABLE 6-10. SNMP Traps

| SNMP Trap OID               | SNMP Traps                                                | Description                                                                                                                                                                                |
|-----------------------------|-----------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.3.6.1.4.1.30221.2.10.3.45 | LOW_DISK_SPACE_WARNING                                    | Indicates that the amount of usable disk space has dropped below the low space warning threshold.                                                                                          |
| 1.3.6.1.4.1.30221.2.10.3.46 | LOW_DISK_SPACE_ERROR                                      | Indicates that the amount of usable disk space has dropped below the low space error threshold.                                                                                            |
| 1.3.6.1.4.1.30221.2.10.3.47 | OUT_OF_DISK_SPACE_ERROR                                   | Indicates that the amount of usable disk space has dropped below the out-of-space error threshold.                                                                                         |
| 1.3.6.1.4.1.30221.2.10.3.48 | CANNOT_WRITE_SERVER_STATE_FILE                            | Indicates that the server cannot write the server state file.                                                                                                                              |
| 1.3.6.1.4.1.30221.2.10.3.49 | DEADLOCK-DETECTED                                         | Indicates that a deadlock has been detected in the JVM in which the server is running.                                                                                                     |
| 1.3.6.1.4.1.30221.2.10.3.50 | WORK_QUEUE_BACKLOGGED                                     | Indicates that the work queue has become significantly backlogged and operations have been required to wait a significant length of time to be processed.                                  |
| 1.3.6.1.4.1.30221.2.10.3.51 | THREAD_EXIT_HOLDING_LOCK                                  | Indicates that a thread has exited while holding one or more locks.                                                                                                                        |
| 1.3.6.1.4.1.30221.2.10.3.52 | REPLICATION_MISSING_CHANGES                               | Indicates that the replication server may have missed some replication changes.                                                                                                            |
| 1.3.6.1.4.1.30221.2.10.3.53 | REPLICATION_PLUGIN_MESSAGE_SERIALIZATION_FAILURE          | Indicates that the replication plugin failed to serialize or deserialize a replication message.                                                                                            |
| 1.3.6.1.4.1.30221.2.10.3.54 | REPLICATION_SERVER_CHANGELOG_FAILURE                      | Indicates that the replication server encountered an error while accessing the replication changelog database.                                                                             |
| 1.3.6.1.4.1.30221.2.10.3.55 | HEALTH_CHECK_AVAILABLE_TO_DEGRADED                        | Indicates that the health of an LDAP external server has been reclassified from available to degraded.                                                                                     |
| 1.3.6.1.4.1.30221.2.10.3.56 | HEALTH_CHECK_AVAILABLE_TO_UNAVAILABLE                     | Indicates that the health of an LDAP external server has been reclassified from available to unavailable.                                                                                  |
| 1.3.6.1.4.1.30221.2.10.3.57 | HEALTH_CHECK_DEGRADED_TO_AVAILABLE                        | Indicates that the health of an LDAP external server has been reclassified from degraded to available.                                                                                     |
| 1.3.6.1.4.1.30221.2.10.3.58 | HEALTH_CHECK_DEGRADED_TO_UNAVAILABLE                      | Indicates that the health of an LDAP external server has been reclassified from degraded to unavailable.                                                                                   |
| 1.3.6.1.4.1.30221.2.10.3.59 | HEALTH_CHECK_UNAVAILABLE_TO_AVAILABLE                     | Indicates that the health of an LDAP external server has been reclassified from unavailable to available.                                                                                  |
| 1.3.6.1.4.1.30221.2.10.3.60 | HEALTH_CHECK_UNAVAILABLE_TO_DEGRADED                      | Indicates that the health of an LDAP external server has been reclassified from unavailable to degraded.                                                                                   |
| 1.3.6.1.4.1.30221.2.10.3.61 | PROXY_ENTRY_BALANCING_OPERATION_FAILURE                   | Indicates that an entry balancing request processor attempted to execute an operation on all configured backend sets, but the operation failed on one or more sets.                        |
| 1.3.6.1.4.1.30221.2.10.3.62 | PROXY_ENTRY_BALANCING_ERROR_MULTIPLE_OPERATIONS_SUCCEEDED | Indicates that an entry balancing request processor successfully executed an operation on multiple backend sets. Only one backend set was expected to succeed with the operation.          |
| 1.3.6.1.4.1.30221.2.10.3.63 | LARGE_ATTRIBUTE_UPDATE_FAILURE                            | Indicates that an error occurred while updating large attribute information in the directory proxy server, and the large attribute information may be out-of-sync with the entry contents. |
| 1.3.6.1.4.1.30221.2.10.3.64 | ACCESS_LOG_CRITERIA_MATCHED                               | Indicates that an access log message matched the criteria for the admin alert access log publisher.                                                                                        |

TABLE 6-10. SNMP Traps

| SNMP Trap OID               | SNMP Traps                         | Description                                                                                                                                                         |
|-----------------------------|------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.3.6.1.4.1.30221.2.10.3.65 | REPLICATION_BACKLOGGED             | Indicates that the replication backlog has exceeded the replication backlog count alert threshold for longer than the replication backlog duration alert threshold. |
| 1.3.6.1.4.1.30221.2.10.3.66 | REPLICATION_MONITOR_DATA_AVAILABLE | Indicates that the replication monitor data is unavailable from <b>cn=monitor</b> .                                                                                 |
| 1.3.6.1.4.1.30221.2.10.3.67 | LBA_NO_AVAILABLE_SERVERS           | Indicates that a load-balancing algorithm instance does not have any available servers.                                                                             |
| 1.3.6.1.4.1.30221.2.10.3.68 | FORCE_GC_COMPLETE                  | Indicates that the server has completed a forced garbage collection.                                                                                                |
| 1.3.6.1.4.1.30221.2.10.3.69 | FORCE_GC_STARTING                  | Indicates that the server is about to force a synchronous garbage collection.                                                                                       |
| 1.3.6.1.4.1.30221.2.10.3.70 | CONFIG_CHANGE                      | Indicates that a configuration change has been made in the server.                                                                                                  |
| 1.3.6.1.4.1.30221.2.10.3.71 | SYNC_RESOURCE_CONNECTION_ERROR     | Indicates that the Sync Server could not connect to an external resource.                                                                                           |
| 1.3.6.1.4.1.30221.2.10.3.72 | SYNC_RESOURCE_OPERATION_ERROR      | Indicates that the Sync Server had an operation error at the external server.                                                                                       |
| 1.3.6.1.4.1.30221.2.10.3.73 | ACCESS_CONTROL_CHANGE              | Indicates that a change has been made to the access control configuration.                                                                                          |
| 1.3.6.1.4.1.30221.2.10.3.74 | INDEX_DEGRADED                     | Indicates that a backend has an index in a degraded state which must be rebuilt before that index may be used.                                                      |
| 1.3.6.1.4.1.30221.2.10.3.75 | INDEX_REBUILD_IN_PROGRESS          | Indicates that a backend has an index in a degraded state which must be rebuilt before that index may be used.                                                      |
| 1.3.6.1.4.1.30221.2.10.3.76 | INDEX_REBUILD_COMPLETED            | Indicates that a backend has completed the process of rebuilding an index.                                                                                          |
| 1.3.6.1.4.1.30221.2.10.3.77 | SYSTEM_NANOTIME_STOPPED            | Indicates that Java's System.nanoTime() has stopped returning updated values.                                                                                       |
| 1.3.6.1.4.1.30221.2.10.3.78 | BACKEND_INITIALIZATION_FAILED      | Indicates that an error occurred while attempting to initialize a directory proxy server backend.                                                                   |

## Profiling Server Performance Using the Periodic Stats Logger

The Directory Proxy Server ships with a built-in Periodic Stats Logger that is useful for profiling server performance for a given configuration. At specified intervals, the periodic stats logger writes server statistics to a log file in a comma-separated format (.csv), which can be ready by most spreadsheet applications. The logger has a negligible impact on server performance unless the `log-interval` property is set to a very small value.

You can control what statistics are logged and their verbosity. We recommend that you see what stats are available by experimenting with configurations or values. For example, you can make a configuration change in a test environment and see what stats are logged in the file.

## To Enable the Periodic Stats Logger

By default, the directory proxy server ships with the periodic stats logger disabled. To enable it using the `dsconfig` tool or the web console, go to Plugins menu (available on the Advanced object menu), and then, select "Stats Logger."

1. Run `dsconfig` in interactive mode. Enter the LDAP or LDAPS connection parameters when prompted.
2. Enter "o" to change to the Advanced Objects menu.
3. On the Directory Server configuration console menu, enter the number for Plugins. In this example, enter 33.
4. On the Plugin management menu, enter 3 to view and edit an existing plugin.
5. On the Plugin selection list, enter 9 for Stats Logger.
6. On the Periodic Stats Logger Plugin menu, enter 2 to set the `enabled` property to `TRUE`. When done, enter `f` to save and apply the configuration. The default logger will log information about the server every second to `<server-root>/logs/dsstats.csv`. If the server is idle, nothing will be logged, but this can be changed by setting the `suppress-if-idle` property to `FALSE` (`suppress-if-idle=false`).

>>>> Configure the properties of the Periodic Stats Logger Plugin

|     | Property                                                                                                       | Value(s)                                                             |
|-----|----------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------|
| 1)  | description                                                                                                    | Logs performance stats to a log file periodically.                   |
| 2)  | enabled                                                                                                        | true                                                                 |
| 3)  | log-interval                                                                                                   | 1 s                                                                  |
| 4)  | collection-interval                                                                                            | 200 ms                                                               |
| 5)  | suppress-if-idle                                                                                               | true                                                                 |
| 6)  | header-prefix-per-column                                                                                       | false                                                                |
| 7)  | empty-instead-of-zero                                                                                          | true                                                                 |
| 8)  | lines-between-header                                                                                           | 50                                                                   |
| 9)  | included-ldap-stat                                                                                             | active-operations, num-connections, op-count-and-latency, work-queue |
| 10) | included-resource-stat                                                                                         | memory-utilization                                                   |
| 11) | histogram-format                                                                                               | count                                                                |
| 12) | histogram-op-type                                                                                              | all                                                                  |
| 13) | local-db-backend-info                                                                                          | basic                                                                |
| 14) | replication-info                                                                                               | basic                                                                |
| 15) | log-file                                                                                                       | logs/dsstats.csv                                                     |
| 16) | log-file-permissions                                                                                           | 640                                                                  |
| 17) | append                                                                                                         | true                                                                 |
| 18) | rotation-policy                                                                                                | Fixed Time Rotation Policy, Size Limit Rotation Policy               |
| 19) | retention-policy                                                                                               | File Count Retention Policy                                          |
| ?)  | help                                                                                                           |                                                                      |
| f)  | finish - apply any changes to the Periodic Stats Logger Plugin                                                 |                                                                      |
| a)  | hide advanced properties of the Periodic Stats Logger Plugin                                                   |                                                                      |
| d)  | display the equivalent dsconfig command lines to either re-create this object or only to apply pending changes |                                                                      |
| b)  | back                                                                                                           |                                                                      |
| q)  | quit                                                                                                           |                                                                      |

Enter choice [b]: f

7. Run the directory proxy server. For example, if you are running in a test environment, you can run the `search-and-mod-rate` tool to apply some searches and modifications to the server. You can run `search-and-mod-rate --help` to see an example command.
8. View the Stats log output in `<server-root>/logs/dsstats.csv`. You can open the file in a spreadsheet.

## To Configure Multiple Periodic Stats Loggers

Multiple periodic stats loggers can be created to log different stats or to create a log at different intervals (such as logging cumulative operations statistics every hour). To create a new log, use the existing stats logger as a template to get reasonable settings, including rotation and retention policy.

1. Run `dsconfig` by repeating steps 1–3 in “To Enable the Periodic Stats Logger” on page 134 (the previous procedure).
2. From the Plugin management menu, enter 2 to create a new plugin.
3. From the How to Create a New Plugin menu, enter `t` to use an existing plugin as a template.
4. Enter 9 to select the existing stats logger as a template.
5. Next, enter a descriptive name for the new stats logger. For this example, type `Stats Logger-10s`.
6. Enter the log file path to the file. For this example, type `logs/dsstats2.csv`.
7. On the Periodic Stats Logger Plugin menu, make any other changes to your logger. For this example, change the `log-interval` to `10s`, and the `suppress-if-idle` to `false`. When finished, enter `f` to save and apply the configuration.
8. You should now see two loggers `dsstats.csv` and `dsstats2.csv` in the logs directory.

## Configuring a Custom Periodic Stats Logger

You can add custom statistics based on any attribute in an entry under `cn=monitor` using the Custom Logged Stats object. This configuration object provides powerful controls for how monitor attributes are written to the log. For example, you can extract a value from a monitor attribute using a regular expression.

Besides allowing a straight passthrough of the values using the 'raw' statistic-type, you can configure attributes to be treated as a *counter* (where the interval includes the difference in the value since the last interval), an *average*, a *minimum*, or a *maximum* value held by the attribute

during the interval specified. The value of an attribute can also be scaled by a fixed value or by the value of another monitor attribute.

---

**Note**

Custom third-party server extensions that were written using the UnboundID Server SDK can also expose interval statistics using the Periodic Stats Logger. The extension must first implement the SDK's MonitorProvider interface and register with the server. The monitor attributes produced by this custom MonitorProvider are then available to be referenced by a Custom Logged Stats object.

---

To illustrate how to configure a Custom Period Stats Logger, the following procedure reproduces the built-in "Consumer Total GB" column that shows up in the output when the `included-resource-stat` property is set to `memory-utilization` on the Periodic Stats Logger. The column is derived from the `total-bytes-used-by-memory-consumers` attribute of the `cn=JVM Memory Usage,cn=monitor` entry as follows:

```
dn: cn=JVM Memory Usage,cn=monitor
objectClass: top
objectClass: ds-monitor-entry
objectClass: ds-memory-usage-monitor-entry
objectClass: extensibleObject
cn: JVM Memory Usage
...
total-bytes-used-by-memory-consumers: 3250017037
```

## To Configure a Custom Periodic Stats Logger Using dsconfig Interactive

1. Run `dsconfig` and enter the LDAP/LDAPS connection parameters when prompted.

```
$ bin/dsconfig
```

2. On the Directory Proxy Server configuration main menu (Advanced Objects menu), select `Custom Logged Stats`.

3. On the Custom Logged Stats menu, enter 2 to create a new custom logged stats.

```
>>>> Custom Logged Stats management menu

What would you like to do?

1) List existing Custom Logged Stats
2) Create a new Custom Logged Stats
3) View and edit an existing Custom Logged Stats
4) Delete an existing Custom Logged Stats

b) back
q) quit

Enter choice [b]: 2
```

4. Select the Periodic Stats Logger Plugin from the list if more than one is present on the system. If you only have one stats logger, press Enter to confirm that you want to use the existing plugin.

```
>>>> There is only one Periodic Stats Logger Plugin: 'Stats Logger'. Are you sure
that this is the correct one? (yes / no) [yes]:
```



5. Enter a descriptive name for the Custom Logged Stats. For this example, enter **Memory Usage**.

```
>>>> Enter a name for the Custom Logged Stats that you want to create: Memory Usage
```

6. From the monitor-objectclass property menu, enter the objectclass attribute to monitor. For this example, enter **ds-memory-usage-monitor-entry**. You can run **ldapsearch** using the base DN "**cn=JVM Memory Usage,cn=monitor**" entry to view the entry.

```
>>>> Configuring the 'monitor-objectclass' property
>>>> via creating 'Memory Usage' Custom Logged Stats
```

The objectclass name of the monitor entries to examine for generating these statistics.

Syntax: monitor-entry-objectclass - The objectclass name of the monitor entries to examine for generating these statistics.

```
Enter a value for the 'monitor-objectclass' property: ds-memory-usage-monitor-entry
```

7. Next, specify the attributes of the monitor entry that you want to log in the stats logger. In this example, enter **total-bytes-used-by-memory-consumers**, and then press Enter again to continue.

```
>>>> Configuring the 'attribute-to-log' property
>>>> via creating 'Memory Usage' Custom Logged Stats
```

Specifies the attributes on the monitor entries that should be included in the output.

The full column name, which appears in the header, is built from the header-prefix, the header-prefix-attribute, and the column name, which is either determined automatically from the monitor entry attribute name or is specified explicitly in the column-name property. Both the header-prefix, and the header-prefix-attribute column names are optional.

Syntax: monitor-entry-objectclass - The name of the attribute to log.

```
Enter a value for the 'attribute-to-log' property: total-bytes-used-by-memory-consumers
```

```
Enter another value for the 'attribute-to-log' property [continue]:
```

8. Next, specify the type of statistics for the monitored attribute that will appear in the log file. In this example, enter **5** for **raw** statistics as recorded by the logger.

```
>>>> Configuring the 'statistic-type' property
>>>> via creating 'Memory Usage' Custom Logged Stats
```

Specifies the type of statistic to include in the output for each monitored attribute.

The statistics listed here will be applied to each attribute listed in the attribute-to-log list. The statistic is computed after all other optional processing such as regular expression manipulation and value scaling.

If a single monitor entry includes attributes of different statistic-types, then multiple Custom Logged Stats objects must be specified.

```
Select one or more values for the 'statistic-type' property:
```

- 1) average
- 2) counter
- 3) maximum
- 4) minimum
- 5) raw
- ? ) help
- c) cancel
- q) quit

```
Enter one or more choices separated by commas [c]: 5
```

9. In the Custom Logged Stats menu, review the configuration. At this point, we want to set up a column name that lists the Memory Usage. Enter 6 to change the `column-name` property.

```
>>>> Configure the properties of the Custom Logged Stats
>>>> via creating 'Memory Usage' Custom Logged Stats
```

|     | Property                  | Value(s)                             |
|-----|---------------------------|--------------------------------------|
| 1)  | description               | -                                    |
| 2)  | enabled                   | true                                 |
| 3)  | monitor-objectclass       | ds-memory-usage-monitor-entry        |
| 4)  | include-filter            | -                                    |
| 5)  | attribute-to-log          | total-bytes-used-by-memory-consumers |
| 6)  | column-name               | -                                    |
| 7)  | statistic-type            | raw                                  |
| 8)  | header-prefix             | -                                    |
| 9)  | header-prefix-attribute   | -                                    |
| 10) | regex-pattern             | -                                    |
| 11) | regex-replacement         | -                                    |
| 12) | divide-value-by           | -                                    |
| 13) | divide-value-by-attribute | -                                    |
| 14) | decimal-format            | ###                                  |
| 15) | non-zero-implies-not-idle | false                                |

```
? ) help
f ) finish - create the new Custom Logged Stats
a ) hide advanced properties of the Custom Logged Stats
d ) display the equivalent dsconfig arguments to create this object
b ) back
q ) quit

Enter choice [b]: 6
```

10. Next, we want to add a specific label for the column name. Enter 2 to add a value, and then enter **Memory Consumer Total (GB)**, and press Enter again to continue.

```
>>>> Configuring the 'column-name' property
>>>> via creating 'Memory Usage' Custom Logged Stats
```

Optionally, specifies an explicit name for each column header instead of having these names automatically generated from the monitored attribute name.

The full column name, which appears in the header, is built from the header-prefix, the header-prefix-attribute, and the column name, which is either determined automatically from the monitor entry attribute name or is specified explicitly in the column-name property. The column-name, header-prefix, and header-prefix-attribute properties are optional.

The number of items in this property must line up with the attribute-to-log property. That is, the first value in this property will be used as the column name for the first value in the attribute-to-log property.

Syntax: STRING

```
Do you want to modify the 'column-name' property?

1) Leave undefined
2) Add one or more values

? ) help
q ) quit

Enter choice [1]: 2

Enter a value for the 'column-name' property [continue]: Memory Consumer Total (GB)

Enter another value for the 'column-name' property [continue]:
```

11. Confirm that you want to use the column-name value that you entered in the previous step, and then press Enter to use the value.

```
>>>> Configuring the 'column-name' property (Continued)

Do you want to modify the 'column-name' property?
```

```

1) Use the value: Memory Consumer Total (GB)
2) Add one or more values
3) Remove one or more values
4) Leave undefined
5) Revert changes

?) help
q) quit

```

Enter choice [1]:

12. Next, we want to scale the Memory Consumer Totals by one gigabyte. On the Custom Logged Stats menu, enter 12 to change the `divide-value-by` property.

```

>>>> Configure the properties of the Custom Logged Stats
>>>> via creating 'Memory Usage' Custom Logged Stats

```

| Property                      | Value(s)                             |
|-------------------------------|--------------------------------------|
| 1) description                | -                                    |
| 2) enabled                    | true                                 |
| 3) monitor-objectclass        | ds-memory-usage-monitor-entry        |
| 4) include-filter             | -                                    |
| 5) attribute-to-log           | total-bytes-used-by-memory-consumers |
| 6) column-name                | Memory Consumer Total (GB)           |
| 7) statistic-type             | raw                                  |
| 8) header-prefix              | -                                    |
| 9) header-prefix-attribute    | -                                    |
| 10) regex-pattern             | -                                    |
| 11) regex-replacement         | -                                    |
| 12) divide-value-by           | -                                    |
| 13) divide-value-by-attribute | -                                    |
| 14) decimal-format            | ###                                  |
| 15) non-zero-implies-not-idle | false                                |

```

?) help
f) finish - create the new Custom Logged Stats
a) hide advanced properties of the Custom Logged Stats
d) display the equivalent dsconfig arguments to create this object
b) back
q) quit

```

Enter choice [b]: 12

13. On the "divide-value-by" property menu, enter 2 to change the value, and then enter 1073741824 (i.e., 1073741824 bytes = 1 gigabytes).

```

>>>> Configuring the 'divide-value-by' property
>>>> via creating 'Memory Usage' Custom Logged Stats

```

An optional floating point value that can be used to scale the resulting value.

This is used to scale the resulting value by a fixed amount. This is typically used to reduce large values to a more manageable length. For instance, free disk space could be represented in GB instead of just bytes. If a value for this property is specified, then it will be applied to the monitored attribute after any regular expression replacement is done and before it is scaled by the divide-value-by-attribute.

Syntax: `divide-by-amount` - A floating point value used to scale the result.

Do you want to modify the 'divide-value-by' property?

```

1) Leave undefined
2) Change the value

?) help
q) quit

```

Enter choice [1]: 2

```
Enter a value for the 'divide-value-by' property [continue]: 1073741824
```

14. On the Custom Logged Stats menu, review your configuration. When finished, enter **f** to save and apply the settings.

```
>>>> Configure the properties of the Custom Logged Stats
>>>> via creating 'Memory Usage' Custom Logged Stats
```

|     | Property                                                        | Value(s)                             |
|-----|-----------------------------------------------------------------|--------------------------------------|
|     | -----                                                           | -----                                |
| 1)  | description                                                     | -                                    |
| 2)  | enabled                                                         | true                                 |
| 3)  | monitor-objectclass                                             | ds-memory-usage-monitor-entry        |
| 4)  | include-filter                                                  | -                                    |
| 5)  | attribute-to-log                                                | total-bytes-used-by-memory-consumers |
| 6)  | column-name                                                     | Memory Consumer Total (GB)           |
| 7)  | statistic-type                                                  | raw                                  |
| 8)  | header-prefix                                                   | -                                    |
| 9)  | header-prefix-attribute                                         | -                                    |
| 10) | regex-pattern                                                   | -                                    |
| 11) | regex-replacement                                               | -                                    |
| 12) | divide-value-by                                                 | 1073741824                           |
| 13) | divide-value-by-attribute                                       | -                                    |
| 14) | decimal-format                                                  | ###                                  |
| 15) | non-zero-implies-not-idle                                       | false                                |
| ?)  | help                                                            |                                      |
| f)  | finish - create the new Custom Logged Stats                     |                                      |
| a)  | hide advanced properties of the Custom Logged Stats             |                                      |
| d)  | display the equivalent dsconfig arguments to create this object |                                      |
| b)  | back                                                            |                                      |
| q)  | quit                                                            |                                      |

```
Enter choice [b]:f
```

```
The Custom Logged Stats was created successfully
```

## To Configure a Custom Periodic Stats Logger Using dsconfig Interactive

Use the **dsconfig** non-interactive command-line equivalent to create your custom stats logger. The following one-line command replicates the procedure in the previous section.

This command produces a column named "Memory Consumer Total (GB)" that contains the value of the **total-bytes-used-by-memory-consumers** attribute pulled from the entry with the **ds-memory-usage-monitor-entry** objectclass. This value is scaled by 1073741824 to get to a value represented in GBs.

```
$ bin/dsconfig create-custom-logged-stats --plugin-name "Stats Logger" \
--stats-name "Memory Usage" --type custom \
--set monitor-objectclass:ds-memory-usage-monitor-entry \
--set attribute-to-log:total-bytes-used-by-memory-consumers \
--set "column-name:Memory Consumer Total (GB)" --set statistic-type:raw \
--set divide-value-by:1073741824
```

## Working with Account Status Notifications

The UnboundID Directory Proxy Server supports notification handlers that can be used to notify users and/or administrators of significant changes related to password policy state for user entries. The following two notification handlers are available:

- **Error Log Account Status Notification Handler.** Enabled by default. The handlers send alerts to the error log when an account event occurs.
- **SMTP Account Status Notification Handler.** Disabled by default. You can enable the SMTP Handler with the `dsconfig` command to send notifications to designated email addresses.

The handlers send alerts when one of the account status events described in the following table occur during password policy processing.

**TABLE 6-11. Account Status Notification Types**

| Account Status Notification Type | Description                                                                                                                                                                                               |
|----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| account-disabled                 | Generates a notification whenever a user account is disabled by an administrator.                                                                                                                         |
| account-enabled                  | Generates a notification whenever a user account is enabled by an administrator.                                                                                                                          |
| account-expired                  | Generates a notification whenever a user authentication attempt fails because the account has expired.                                                                                                    |
| account-idle-locked              | Generates a notification whenever a user authentication attempt fails because the account has been locked after idling for too long.                                                                      |
| account-permanently-locked       | Generates a notification whenever a user account is permanently locked (requiring administrative action to unlock the account) after too many failed attempts.                                            |
| account-reset-locked             | Generates a notification whenever an authentication attempt fails because the user account is locked because the user failed to change a password reset by an administrator within the required interval. |
| account-temporarily-locked       | Generates a notification whenever a user account is temporarily locked after too many failed attempts.                                                                                                    |
| account-unlocked                 | Generates a notification whenever a user account is unlocked by an administrator.                                                                                                                         |
| password-changed                 | Generates a notification whenever a user changes his or her own password.                                                                                                                                 |
| password-expired                 | Generates a notification whenever a user authentication fails because the password has expired.                                                                                                           |
| password-expiring                | Generates a notification the first time that a password expiration warning is encountered for a user password.                                                                                            |
| password-reset                   | Generates a notification whenever a user's password is reset by an administrator.                                                                                                                         |

## Working with the Error Log Account Status Notification Handler

The Error Log Account Status Notification Handler is enabled by default and sends alerts when one of the account status events occur.

### To Disable the Error Log Account Status Notification Handler

Use the `dsconfig` tool to disable the Error Log Handler. You can view the log at `logs/error`.

```
$ bin/dsconfig set-account-status-notification-handler-prop \  
  --hostname host1 --port 1389 --bindDN "uid=admin,dc=example,dc=com" \  
  --bindPassword secret --handler-name "Error Log Handler" --set  
enabled:false
```

### To Remove a Notification Type from the Error Log Handler

While not recommended, if you want to remove an account status notification type, use the `dsconfig` tool with the `--remove` option.

```
$ bin/dsconfig set-account-status-notification-handler-prop \  
  --hostname host1 --port 1389 \  
  --bindDN "uid=admin,dc=example,dc=com" --bindPassword secret \  
  --handler-name "Error Log Handler" \  
  --remove account-status-notification-type: password-reset
```

## Working with the SMTP Account Status Notification Handler

You can enable account status notifications to be sent to designated email addresses of end users, administrators, or both through an outgoing SMTP server. The email message is automatically generated from template files that contain the text to use in the message body. For example, the message subject for the `account-disabled` event is:

```
account-disabled: Your directory account has been disabled.
```

The message templates are located in the `config/messages` directory. The typical message body template is as follows:

```
Your directory account has been disabled.
```

```
For further assistance, please contact a server administrator.
```

By default, the sender address is `notifications@example.com`, but you can configure your own address.

Before you enable the SMTP Account Status Notification Handler, you must configure the directory proxy server to use at least one mail server as shown below. You can configure an SMTP server using `dsconfig` and the `set-global-configuration-prop` option.

### To Configure the SMTP Server

Use `dsconfig` to configure an SMTP server.

```
$ bin/dsconfig set-global-configuration-prop \  
  --hostname host1 --port 1389 --bindDN "uid=admin,dc=example,dc=com" \  
  --bindPassword secret --set smtp-server:mailserver.example.com
```

## To Enable the SMTP Account Status Notification Handler

Use `dsconfig` to enable the SMTP server.

```
$ bin/dsconfig set-account-status-notification-handler-prop \
  --hostname host1 --port 1389 --bindDN "uid=admin,dc=example,dc=com" \
  --bindPassword secret --handler-name "SMTP Server" \
  --set enabled:true \
  --set "recipient-address:admin@example.com" \
  --set "sender-address:acct-status-notifications@example.com"
```

## To View the Account Status Notification Handlers

After you have enabled the SMTP server, view the list of account status notification handlers using `dsconfig`.

```
$ bin/dsconfig list-account-status-notification-handlers \
  --hostname localhost --port 1389 \
  --bindDN "uid=admin,dc=example,dc=com" --bindPassword secret --no-prompt
```

|                                     |   |           |   |         |
|-------------------------------------|---|-----------|---|---------|
| Account Status Notification Handler | : | Type      | : | enabled |
| -----:-----:-----                   |   |           |   |         |
| Error Log Handler                   | : | error-log | : | true    |
| SMTP Handler                        | : | smtp      | : | true    |

## Associating Account Status Notification Handlers with Password Policies

In order to generate notifications whenever appropriate password policy state changes occur in the server, the password policy that governs the entry being updated must be configured to use one or more account status notification handlers. By default, password policies are not configured with any such handlers, and therefore, no account status notifications will be generated.

The set of account status notification handlers that should be in use for a password policy is controlled by the `account-status-notification-handler` property for that password policy. It can be configured using `dsconfig` or the web administration console. For example, the following change updates the default password policy, so that the error log account status notification handler will be invoked for any appropriate password policy state changes for entries governed by the default password policy:

```
$ bin/dsconfig set-password-policy-prop --hostname host1 --port 1389 \
  --bindDN "uid=admin,dc=example,dc=com" --bindPassword secret \
  --policy-name "Default Password Policy" \
  --set "account-status-notification-handler:Error Log Handler"
```

## Working with Administrative Alert Handlers

UnboundID Directory Proxy Server provides mechanisms to send alert notifications to administrators when significant problems or events occur during processing, such as problems during server startup or shutdown. The directory proxy server provides four types of alert handlers:

- **Admin Alert Log Publisher.** Sends administrative alerts to a log publisher. Specifically, the Admin Alert Access Log Publisher generates administrative alerts when certain criteria appear in the File-Based Access Log.
- **JMX Alert Handler.** Sends administrative alerts to clients using the Java Management Extensions (JMX) protocol. UnboundID uses JMX for monitoring entries and requires that the JMX connection handler be enabled.
- **SMTP Alert Handler.** Sends administrative alerts to clients via email using Simple Mail Transfer Protocol (SMTP). The server requires that one or more SMTP servers be defined in the global configuration.
- **SNMP Alert Handler.** Sends administrative alerts to clients using Simple Network Monitoring Protocol (SNMP). The server must have an SNMP agent capable of communicating via SNMP 2c.

### Configuring the JMX Connection Handler and Alert Handler

You can configure the JMX connection handler and alert handler respectively using the `dsconfig` tool. Any user allowed to receive JMX notifications must have the `jmx-read`, `jmx-write`, and `jmx-notify` privileges. By default, these privileges are not granted to any users (including root users or global administrators). For security reasons, we recommend that you create a separate user account that does not have any other privileges but the ones listed below. You can also configure the JMX connection handler and alert handler using `dsconfig` in interactive command-line mode, which is visible on the "Standard" object complexity menu.

#### To Configure the JMX Connection Handler

1. Use `dsconfig` to enable the JMX Connection Handler.

```
$ bin/dsconfig set-connection-handler-prop \  
  --handler-name "JMX Connection Handler" \  
  --set enabled:true --set listen-port:1689 \  
  --hostname host1 --port 1389 \  
  --bindDN "cn=Directory Manager" \  
  --bindPassword secret \  
  --no-prompt
```

2. Add a new non-root user account with the `jmx-read`, `jmx-write`, and `jmx-notify` privileges. This account can be added using the `ldapmodify` tool using an LDIF representation like:



```
dn: uid=jmx-user,dc=example,dc=com
changetype: add
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
uid: jmx-user
givenName: JMX
sn: User
cn: JMX User
userPassword: password
ds-privilege-name: jmx-read
ds-privilege-name: jmx-write
ds-privilege-name: jmx-notify
```

## To Configure the JMX Alert Handler

Use `dsconfig` to configure the JMX Alert Handler.

```
$ bin/dsconfig --no-prompt set-alert-handler-prop \
--hostname host1 --port 1389 \
--bindDN "cn=Directory Manager" --bindPassword secret \
--handler-name "JMX Alert Handler" --set enabled:true
```

## Configuring the SMTP Alert Handler

By default, there is no configuration entry for an SMTP alert handler. To create a new instance of an SMTP Alert Handler, use the `dsconfig` tool.

## To Configure the SMTP Alert Handler

By default, there is no configuration entry for an SMTP alert handler. To create a new instance of an SMTP Alert Handler, use the `dsconfig` tool.

```
$ bin/dsconfig create-alert-handler \
--hostname host1 \
--port 1389 \
--bindDN "cn=Directory Manager" \
--bindPassword secret \
--no-prompt \
--handler-name "SMTP Alert Handler" \
--type smtp \
--set enabled:true \
--set "sender-address:alerts@example.com" \
--set "recipient-address:administrators@example.com" \
--set "message-subject:Directory Proxy Admin Alert %\%alert-type%\%" \
--set "message-body:Administrative alert:\\n\\%\%alert-message%\%"
```

## Configuring the SNMP Alert Handler

You can configure the SNMP alert handler using the `dsconfig` tool, which is visible at the "Standard" object complexity. Before you begin, you need an SNMP agent capable of communicating via SNMP 2c.

### To Configure the SNMP Alert Handler

Use `dsconfig` to configure the SNMP Alert Handler. The `server-host-name` is the address of the system running the SNMP agent. The `server-port` is the port number on which the agent is running (by default, agents listen on port 162 for SNMP traps). The `community-name` is the name of the SNMP community that is used for the traps.

```
$ bin/dsconfig \
  set-alert-handler-prop \
    --hostname host2 \
    --port 1389 \
    --bindDN "cn=Directory Manager" \
    --bindPassword secret \
    --no-prompt \
    --handler-name "SNMP Alert Handler" \
    --set enabled:true \
    --set server-host-name:host2 \
    --set server-port:162 \
    --set community-name:admin

firstchangenumber: 1
lastchangenumber: 8
lastpurgedchangenumber: 0
firstReplicaChange: 16225:0000011D0205237F3F6100000001:5
firstReplicaChange: 16531:0000011CFF334C60409300000002:1
lastReplicaChange: 16225:0000011D02054E8B3F6100000002:7
lastReplicaChange: 16531:0000011CFF334C60409300000002:1
oldest-change-time: 20081015063104Z
...(more data)...
```

## Working with Virtual Attributes

The UnboundID Proxy Server provides dynamically generated attributes called virtual attributes for local proxy server data. The proxy virtual attributes apply to a local proxy backend, such as `cn=config` or the Root DSE. If you want to use virtual attributes in proxied requests, use proxy transformations as described in “Configuring Proxy Transformations” on page 82.

For example, you can define a virtual attribute and assign it to the Root DSE as follows:

```
$ bin/dsconfig create-virtual-attribute \
  --name defineDescriptionOnRootDSE --type user-defined \
  --set enabled:true --set attribute-type:description \
  --set filter:objectclass=ds-root-dse --set value:PrimaryProxy
```

If you search the Root DSE using the following LDAP search, you see that the description attribute now has the value **PrimaryProxy**.

```
$ bin/ldapsearch --baseDN "" --searchScope base --bindDN "" \  
--bindPassword "" --port 5389 -- hostname localhost \  
"objectclass=*" description
```

```
dn:  
description:PrimaryProxy
```



---

# 5

## Managing Password Policies

The UnboundID® Directory Proxy Server provides a flexible password policy system to assign, manage, or remove user passwords for root and non-root users, as well as groups of users. The password policy contains configurable properties for password expiration, failed login attempts, account lockout and other aspects of password maintenance on the directory proxy server. The directory proxy server also provides a global configuration option and a per-user privilege feature that disables parts of the password policy evaluation for production environments that do not require a password policy.

This chapter presents the following topics:

- [Viewing Password Policies](#)
- [Managing the Default Password Policy Properties](#)
- [Modifying the Default Password Policy](#)
- [Creating a New Password Policy](#)
- [Modifying a User's Password](#)
- [Managing Users Accounts](#)
- [Disabling Password Policy Properties](#)

### Viewing Password Policies

Password policies enforce a set of rules that ensure that access to directory data is not compromised through negligent password practices. The UnboundID Directory Proxy Server provides mechanisms to create and maintain password policies that determine whether passwords should expire or not, whether users are allowed to modify their own passwords, or whether failed authentication attempts should result in an account lockout. Many other options are available to fully configure a proprietary password policy for your directory services.

By default, the directory proxy server has a Root Password Policy that governs passwords for root users and a Default Password Policy for other non-root users. Both password policies are similar, except that the Root Password Policy uses a stronger password storage scheme (Salted-SHA512) to that of the Default Password Policy (Salted-SHA1) in addition to some other small differences. The directory proxy server allows you to use the Default Password Policy as a template to create multiple password policies, which can be applied to specific

users, groups of users, or both. These default password policies can be modified but not removed from the system.

## To View Password Policies

You can view the list of password policies configured on the directory proxy server using the `dsconfig` tool. Use `dsconfig` to view the directory proxy server's password policies.

```
$ bin/dsconfig --port 1389 --bindDN "cn=Directory Manager" \
  --bindPassword secret --no-prompt list-password-policies
```

| Password Policy         | Type    | password-attribute | default-password-storage-scheme |
|-------------------------|---------|--------------------|---------------------------------|
| Default Password Policy | generic | userpassword       | Salted SHA-1                    |
| Root Password Policy    | generic | userpassword       | Salted SHA-512                  |

## To View a Specific Password Policy

Use `dsconfig` to view a specific password policy. In this example, view the Default Password Policy that applies to all non-root users.

```
$ bin/dsconfig --port 1389 --bindDN "cn=Directory Manager" \
  --bindPassword secret get-password-policy-prop \
  --policy-name "Default Password Policy" --no-prompt
```

| Property                                  | Value(s)                  |
|-------------------------------------------|---------------------------|
| description                               | -                         |
| password-attribute                        | userpassword              |
| default-password-storage-scheme           | Salted SHA-1              |
| deprecated-password-storage-scheme        | -                         |
| password-validator                        | -                         |
| account-status-notification-handler       | -                         |
| allow-user-password-changes               | true                      |
| password-change-requires-current-password | false                     |
| force-change-on-add                       | false                     |
| force-change-on-reset                     | false                     |
| password-generator                        | Random Password Generator |
| require-secure-authentication             | false                     |
| require-secure-password-changes           | false                     |
| min-password-age                          | 0 s                       |
| max-password-age                          | 0 s                       |
| max-password-reset-age                    | 0 s                       |
| password-expiration-warning-interval      | 5 d                       |
| expire-passwords-without-warning          | false                     |
| allow-expired-password-changes            | false                     |
| grace-login-count                         | 0                         |
| lockout-failure-count                     | 0                         |
| lockout-duration                          | 0 s                       |
| lockout-failure-expiration-interval       | 0 s                       |
| require-change-by-time                    | -                         |
| last-login-time-attribute                 | ds-pwp-last-login-time    |
| last-login-time-format                    | -                         |
| previous-last-login-time-format           | -                         |
| idle-lockout-interval                     | 0 s                       |
| password-history-count                    | 0                         |
| password-history-duration                 | 0 s                       |

## Managing the Default Password Policy Properties

The Default Password Policy provides a number of configurable properties for custom and proprietary password policies for your directory. The following properties are commonly used:

- **allow-user-password-changes**. Specifies whether users can change their own passwords. The system checks this property as well as evaluates the access control instruction that allows the user to modify their own passwords.
- **password-change-requires-current-password**. Specifies whether users must include their current password when changing their password.
- **force-change-on-add**. Specifies whether users are required to change their passwords upon first authenticating to the directory proxy s
- **server** after their account has been created. This property is used in First Login Password policy, which you can easily configure on the Directory Proxy Server.
- **force-change-on-reset**. Specifies whether users are required to change their passwords after they have been reset by the administrator. An administrator is a user who has the **password-reset** privilege and the appropriate access control instruction to allow modification of other users' passwords.
- **max-password-age**. Specifies the maximum length of time that a user can continue the same password before he or she is required to change it. The value can be expressed in seconds (s), minutes (m), hours (h), days (d), or weeks (w).
- **password-expiration-warning-interval**. Specifies the maximum length of time prior to a password expiration date when the server begins sending notifications to users to change their passwords. The value can be expressed in seconds (s), minutes (m), hours (h), days (d), or weeks (w).
- **lockout-failure-count**. Specifies the maximum number of authentication, or login, failures that a user is allowed before the account is locked. If the value is set to 3, then the user is locked out on the fourth failed attempt.
- **password-history-count**. Specifies the maximum number of former passwords to maintain in the password history. Any newly proposed password is checked against this history.

**Note**

To view a description of each of the password policy properties, see the *UnboundID Directory Server Configuration Reference*. The reference is an HTML document that is bundled with the UnboundID Directory Server build.

## Modifying the Default Password Policy

You can modify the Default Password Policy to suit your company's requirements. When creating new password policies, the Default Password Policy serves as a template for the new policy. Any properties that are not explicitly set will take their values from the Default Password Policy, except for the `password-attribute` and `default-password-storage-scheme` properties, which you must explicitly set with every new policy.

### To Modify the Default Password Policy

Use `dsconfig` to modify the Directory Proxy Server's Default Password Policy. The following example sets some of the properties presented in the previous section.

```
$ bin/dsconfig --port 1389 --bindDN "cn=Directory Manager" \
--bindPassword secret --no-prompt set-password-policy-prop \
--policy-name "Default Password Policy" \
--set "max-password-age:90d" \
--set "password-expiration-warning-interval:14d" \
--set "lockout-failure-count:3" \
--set "password-history-count:6" \
--no-prompt
```

## Creating a New Password Policy

You can create any number of password policies using the Default Password Policy as a template. You can add only those specific password policy properties needed for your policy, the other properties will be taken from the Default Password Policy. After you have created the password policy, you can assign the policy to specific entries or to a group of entries.

### To Create a New Password Policy

You can use the Default Password Policy as a template to create a new password policy. Properties that are not explicitly listed in the `dsconfig` command are inherited from the Default Password Policy. However, you must explicitly specify the `password-attribute` and `default-password-storage-scheme` properties.

```
$ bin/dsconfig --no-prompt create-password-policy \
--policy-name "Temp Password Policy" \
--set "password-attribute:userpassword" \
--set "default-password-storage-scheme:Salted SHA-1" \
--set "force-change-on-add:true" --set "force-change-on-reset:true" \
--set "password-expiration-warning-interval:2w" \
--set "max-password-age:90d" --set "lockout-duration:24h" \
--set "lockout-failure-count:3" \
--set "password-change-requires-current-password:true" \
--port 1389 --bindDN "cn=Directory Manager" --bindPassword secret
```



## To Assign a Password Policy to an Individual Account

1. Create a file (`assign.ldif`) with the following contents:

```
dn: uid=user.1,ou=People,dc=example,dc=com
changetype: modify
add: ds-pwp-password-policy-dn
ds-pwp-password-policy-dn: cn=Temp Password Policy,cn=Password Poli-
cies,cn=config
```

2. Use `ldapsearch` to verify the user's password policy assignment.

```
$ bin/ldapmodify --port 1389 --bindDN "cn=Directory Manager" \
--bindPassword secret --filename assign.ldif
```

3. Use `ldapsearch` to verify the user's password policy assignment.

```
$ bin/ldapsearch --port 1389 --bindDN "cn=Directory Manager" \
--bindPassword secret --baseDN dc=example,dc=com "(uid=user.1)" \
ds-pwp-password-policy-dn

dn: uid=user.1,ou=People,dc=example,dc=com
ds-pwp-password-policy-dn: cn=Temp Password Policy,cn=Password Policies,cn=config
```

## To Prevent User Modification of a Password Policy

1. Create a file (`prevent.ldif`) with the following contents.

```
dn: dc=example,dc=com
changetype: modify
add: aci
aci:(targetattr != "passwordPolicySubentry")(version 3.1; acl "Allow self
modification except for passwordPolicySubentry"; allow (write)(userdn =
"ldap:///self");)
```

2. Use `ldapmodify` to prevent users from modifying their own password policy.

```
$ bin/ldapmodify --port 1389 --bindDN "cn=Directory Manager" \
--bindPassword secret --filename prevent.ldif
```

## To Assign a Password Policy to a Group of Users

1. Create some groups in a file (`groups.ldif`) and save the file.

```
dn: ou=Groups,dc=example,dc=com
objectClass: organizationalunit
objectClass: top
ou: Groups

dn: cn=Engineering Managers,ou=groups,dc=example,dc=com
objectClass: groupOfUniqueNames
objectClass: top
cn: Engineering Managers
```

```
uniqueMember: uid=user.0,ou=People,dc=example,dc=com
ou: groups
```

2. Apply the groups using `ldapmodify`.

```
$ bin/ldapmodify --port 1389 --bindDN "cn=Directory Manager" \
  --bindPassword secret --defaultAdd --filename groups.ldif
```

3. Use `dsconfig` to add a password policy to a group of users. Create a virtual attribute that assigns the password policy to the `cn=Engineering Managers` entry. Use the entry file as applied in the previous step.

```
$ bin/dsconfig create-virtual-attribute \
  --name "Eng Mgrs Password Policy" \
  --type user-defined \
  --set "description:Eng Mgrs Grp PWPolicy" \
  --set enabled:true \
  --set attribute-type:ds-pwp-password-policy-dn \
  --set "value:cn=Temp Password Policy,cn=Password Policies,cn=config" \
  --set "group-dn:cn=Engineering Managers,ou=Groups,dc=example,dc=com" \
  --port 1389 --bindDN "cn=Directory Manager" --bindPassword secret \
  --no-prompt
```

4. Use `ldapsearch` to verify that a user in the group contains the assigned password policy DN.

```
$ bin/ldapsearch --port 1389 --bindDN "cn=Directory Manager" \
  --bindPassword secret --baseDN dc=example,dc=com "(uid=user.0)" \
  ds-pwp-password-policy-dn

dn: uid=user.0,ou=People,dc=example,dc=com
ds-pwp-password-policy-dn: cn=Temp Password Policy,cn=Password Policies,cn=config
```

## To Delete a Password Policy

Use `dsconfig` to delete a password policy, and type **YES** to confirm that you want to delete the password policy.

```
$ bin/dsconfig delete-password-policy \
  --policy-name "Temp Password Policy" \
  --port 1389 --bindDN "cn=Directory Manager" --bindPassword secret
```

### Note

If you have assigned a password policy to a user's account and later delete the password policy, the password policy will be removed from the user's account.

## Modifying a User's Password

Use `ldappasswordmodify` to modify a user's password. If you do not specify a new password option, the tool randomly generates a password. Otherwise, you can provide a specific temporary password that the user can change at a later time.

### To Automatically Generate a Temporary Password

Use `ldappasswordmodify` to modify a password and generate temporary password.

```
$ bin/ldappasswordmodify --port 1389 \  
  --bindDN "cn=Directory Manager" \  
  --bindPassword secret --authzID "u:user.1"
```

```
The LDAP password modify operation was successful  
Generated Password: fbi27oqy
```

### To Modify a User's Password

Use `ldappasswordmodify` to modify a user's password with a custom password that you specify.

```
$ bin/ldappasswordmodify --port 1389 \  
  --bindDN "cn=Directory Manager" --bindPassword secret \  
  --authzID "uid=user.1,ou=People,dc=example,dc=com" \  
  --newPassword tempPassword
```

## Managing Users Accounts

The Directory Proxy Server provides a user management utility, the `manage-account` tool, that provides a means to quickly view the password policy properties for a user or a group of users. The utility also allows you to enable or disable a user's account if no longer active. The utility also can be used to view the password policy information and the account state for a user's account.

### To Get the Account State

Use `manage-account` to get the account's state.

```
$ bin/manage-account get-account-is-disabled \  
  --targetDN uid=user.1,ou=People,dc=example,dc=com \  
  --port 1389 --bindDN "cn=Directory Manager" --bindPassword secret
```

```
Account Is Disabled: true
```

## To Disable an Account

Use `manage-account` to disable a user's account.

```
$ bin/manage-account set-account-is-disabled \  
  --operationValue true --targetDN uid=user.1,ou=People,dc=example,dc=com \  
  --port 1389 --bindDN "cn=Directory Manager" --bindPassword secret
```

```
Account Is Disabled: true
```

## To Enable a Disabled Account

Use `manage-account` to enable an account that is in a disabled state.

```
$ bin/manage-account clear-account-is-disabled \  
  --targetDN uid=user.1,ou=People,dc=example,dc=com \  
  --port 1389 --bindDN "cn=Directory Manager" --bindPassword secret
```

```
Account Is Disabled: false
```

## To Return the Password Policy State Information

Use `manage-account` to get information about the account's password policy.

```
$ bin/manage-account get-all \  
  --targetDN uid=user.1,ou=People,dc=example,dc=com \  
  --port 1389 --bindDN "cn=Directory Manager" --bindPassword secret
```

```
Password Policy DN: cn=Temp 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: 1209600  
Seconds Until Password Expiration Warning: 0  
Authentication Failure Times:  
Seconds Until Authentication Failure Unlock:  
Remaining Authentication Failure Count: 3  
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:  
Password History:
```

# Disabling Password Policy Properties

The Directory Proxy Server provides a global configuration property (`disable-password-policy-evaluation`) and a similar privilege (`bypass-pw-policy`) that bypasses password policy evaluation. These features are convenient for those production environments that do not require password policy support. If the `disable password policy` property is set, passwords will still be encoded and evaluated. However, only account expiration and account disabling will be in effect. All password policy properties, such as password expiration, lockout, force change on add or reset, are ignored.

The `bypass-pw-policy` privilege is applied to an individual entry and is intended for cases where external synchronization might require passwords that violate the password validation rules. The privilege is not evaluated for bind operations, so password policy evaluation still takes effect, unless the `disable-password-policy-evaluation` property is set.

## To Globally Bypass Password Policy Evaluation

Use `dsconfig` to set the `disable-password-policy-evaluation` property globally for the Directory Proxy Server.

```
$ bin/dsconfig set-global-configuration-prop \  
  --set "disable-password-policy-evaluation:false" \  
  --port 1389 --bindDN "cn=Directory Manager" \  
  --bindPassword secret --no-prompt
```

## To Bypass Password Policy Evaluation on a Per-User Basis

Use `ldapmodify` to set the `bypass-pw-policy` privilege for a user.

```
$ bin/ldapmodify --port 1389 --bindDN "cn=Directory Manager" \  
  --bindPassword secret  
dn: uid=user.1,ou=People,dc=example,dc=com  
changetype: modify  
add: ds-privilege-name  
ds-privilege-name: bypass-pw-policy
```

# Managing Password Validators

A password validator is a password policy component and is used with an associated password storage scheme to determine if a new password is acceptable or not based on specific validators. You can use all or any combination of the default password validators in a password policy. You also can create new password validators based on the default password validators on the system.

The UnboundID Directory Proxy Server comes with a set of default password validators, shown in the following table.

**TABLE 7-1. Password Validators**

| Password Validators                 | Description                                                                                                                                                                                                                                                                                                                                                                                       |
|-------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Attribute Value                     | Determines if the proposed password is contained in any attribute within the user's entry. The validator can be configured to look in all attributes or in a subset of attributes.                                                                                                                                                                                                                |
| Character Set                       | Determines if the proposed password 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 lower-case letter, one uppercase letter, one digit, and one symbol.                                                                                                                      |
| Dictionary                          | Determines if the proposed password has a value that appears in a provided dictionary file. A large dictionary file is provided with the server, but the administrator can supply an alternate dictionary. In this case, then the dictionary must be a plain-text file with one word per line.                                                                                                    |
| Length-Based Password Validator     | Determines if the proposed password has a character length that falls within an acceptable range of values. Both upper and lower bounds may be defined.                                                                                                                                                                                                                                           |
| Regular Expression Validator        | Determines if the proposed password has a value that matches a defined regular expression. If the password matches the provided regular expression, the Directory Proxy Server can either accept or reject it.                                                                                                                                                                                    |
| Repeated Characters                 | Determines if the proposed password does not contain any characters that repeat or appear consecutively in a password. The validator ensures that user passwords do not contain strings of the same character repeated N-times, such as, "aaaa" or "aaabbb". You can also specify that certain characters must be part of a defined character set.                                                |
| Similarity-Based Password Validator | Determines if the proposed password is similar to the user's current password.<br><br>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 <b>password-change-requires-current-password</b> attribute in the password policy configuration must also be set to true. |
| Unique Characters                   | Determines if the proposed password contains a specific number of unique characters. The validator ensures that user passwords do not contain repeated strings of the same character, such as, "aaaa" or "aaabbb".                                                                                                                                                                                |

## Managing Password Validators

You can use the **dsconfig** configuration tool to configure or modify any password validators. Once you have defined your password validators, you can add them to an existing password policy. The following example procedures show the **dsconfig** non-interactive commands necessary to carry out the tasks. If you use **dsconfig** in interactive command-line mode, you can access the Password Validator menu in the Basic Objects menu. For more details on the password validator properties, see the *UnboundID Directory Server Reference Guide*.

## To View the List of Default Password Validators

Use `dsconfig` to view the list of default password validators on the system. If you require specialized password validators, contact UnboundID for more assistance. The Directory Proxy Server comes with a flexible API for coding any specialized validators.

```
$ bin/dsconfig list-password-validators --port 1389 --bindDN "cn=Directory
Manager" \
--bindPassword secret --no-prompt
```

## Configuring the Attribute Value Password Validator

The Attribute Value Password Validator determines if a proposed password is acceptable or not by determining whether that password is contained in any attribute within the user's entry. The validator is enabled by default but must be assigned to a password policy. By default, the validator checks every attribute in the user's entry and checks for reverse passwords.

### To Configure the Attribute Value Password Validator

1. Use `dsconfig` to edit the existing default configuration for the Attribute Value Password Validator.

```
$ bin/dsconfig set-password-validator-prop \
--validator-name "Attribute Value" \
--set match-attribute:cn \
--set match-attribute:sn \
--set match-attribute:telephonenumber \
--set match-attribute:uid \
--port 1389 --bindDN "cn=Directory Manager" \
--bindPassword secret --no-prompt
```

2. Update an existing password policy to use the Attribute Value Password Validator.

```
$ bin/dsconfig set-password-policy-prop \
--policy-name "Temp Password Policy" \
--set "password-validator:Attribute Value" \
--port 1389 --bindDN "cn=Directory Manager" \
--bindPassword secret --no-prompt
```

3. Test the Attribute Value Password Validator by submitting a password that is identical to one of the configured attributes (`cn`, `sn`, `telephonenumber`, `uid`).

```
$ bin/ldappasswordmodify --port 1389 --bindDN "cn=Directory Manager" \
--bindPassword secret --authzID "uid=user.0,ou=People,dc=example,dc=com"
\
--newPassword user.0
```

```
The LDAP password modify operation failed with result code 53
Error Message: The provided new password failed the validation checks
```

defined in the server: The provided password was found in another attribute in the user entry

## Configuring the Character Set Password Validator

The Character Set Password Validator determines if a proposed password is acceptable or not by determining whether that password contains a sufficient number of characters from one or more user-defined character sets. By default, the validator checks that the password has at least one lowercase letter, one uppercase letter, one digit, and one special character.

### To Configure the Character Set Password Validator

1. Use `dsconfig` to edit the existing default configuration. In this example, we remove the requirement that special characters be required in a password, and add a requirement that two numbers must be included in the password. Thus, in this example, all newly created passwords must have at least one lowercase letter, one uppercase letter, and two numbers.

```
$ bin/dsconfig set-password-validator-prop \
  --validator-name "Character Set" \
  --remove character-set:1:0123456789 \
  --remove "character-set:1:~\!@#\$%\^&*()-_+=[]{}|;:,.<>/?" \
  --add character-set:2:0123456789 \
  --set allow-unclassified-characters:false \
  --port 1389 --bindDN "cn=Directory Manager" \
  --bindPassword secret --no-prompt
```

2. Update an existing password policy to use the Character Set Password Validator.

```
$ bin/dsconfig set-password-policy-prop \
  --policy-name "Temp Password Policy" \
  --set "password-validator:Character Set" \
  --port 1389 --bindDN "cn=Directory Manager" \
  --bindPassword secret --no-prompt
```

3. Test the Character Set Password Validator by submitting a password that meets the requirements (one lowercase letter, one uppercase letter, two digits).

```
$ bin/ldappasswordmodify --port 1389 \
  --bindDN "cn=Directory Manager" \
  --bindPassword secret \
  --authzID "uid=user.0,ou=People,dc=example,dc=com" \
  --newPassword abC12
```

## Configuring the Dictionary Password Validator

The Dictionary Password Validator determines if a proposed password is acceptable or not by determining whether that password appears in a provided dictionary file. By default, the validator is disabled. When the validator is enabled and assigned to a password policy, the system



checks the `config/wordlist.txt` (UNIX, Linux) or `config\wordlist.txt` (Microsoft Windows) file to see if the case insensitive password is in the file.

## To Configure the Dictionary Password Validator

1. Use `dsconfig` to edit the existing default configuration. In this example, we enable the validator and use the default dictionary file located at `config/wordlist.txt`. If you want to specify a specific file, use the `--set dictionary-file:/path/to/file` command.

```
$ bin/dsconfig set-password-validator-prop \  
  --validator-name "Dictionary" --set enabled:true \  
  --port 1389 --bindDN "cn=Directory Manager" \  
  --bindPassword secret --no-prompt
```

2. Update an existing password policy to use the Dictionary Password Validator.

```
$ bin/dsconfig set-password-policy-prop \  
  --policy-name "Temp Password Policy" \  
  --set "password-validator:Dictionary" \  
  --port 1389 --bindDN "cn=Directory Manager" \  
  --bindPassword secret --no-prompt
```

3. Test the Dictionary Password Validator by submitting a password that is identical to one of words in the dictionary file.

```
$ bin/ldappasswordmodify --port 1389 \  
  --bindDN "cn=Directory Manager" --bindPassword secret \  
  --authzID "uid=user.0,ou=People,dc=example,dc=com" \  
  --newPassword Zyrian
```

```
The LDAP password modify operation failed with result code 53  
Error Message: The provided new password failed the validation checks  
defined in the server: The provided password was found in the server's  
dictionary
```

## Configuring the Length-Based Password Validator

The Length-Based Password Validator determines if a proposed password is acceptable or not by determining whether that password contains a specific number of characters. By default, the validator has maximum and minimum length of zero, which indicates that there is no upper and lower bound, respectively, for password length. If both values are defined, the minimum length must be less than or equal to the maximum length.

## To Manage the Length-Based Password Validator

1. Use `dsconfig` to edit the existing default configuration. In this example, we set the required minimum number of characters in a password to five.

```
$ bin/dsconfig create-password-validator \  
--validator-name "Length-Based Password Validator" \  
--set max-password-length:5 --set min-password-length:5 \  
--port 1389 --bindDN "cn=Directory Manager" \  
--bindPassword secret --no-prompt
```

2. Update an existing password policy to use the Length-Based Password Validator.

```
$ bin/dsconfig set-password-policy-prop \  
--policy-name "Temp Password Policy" \  
--set "password-validator:Length-Based Password Policy" \  
--port 1389 --bindDN "cn=Directory Manager" \  
--bindPassword secret --no-prompt
```

3. Test the Length-Based Password Validator by submitting a password that has less than the number of minimum required characters.

```
$ bin/ldappasswordmodify --port 1389 \  
--bindDN "cn=Directory Manager" --bindPassword secret \  
--authzID "uid=user.0,ou=People,dc=example,dc=com" \  
--newPassword abcd
```

```
The LDAP password modify operation failed with result code 53  
Error Message: The provided new password failed the validation checks  
defined in the server: The provided password is shorter than the minimum  
required length of 5 characters
```

## Configuring the Regular Expression Password Validator

The Regular Expression Password Validator determines if a proposed password is acceptable or not by determining whether that password matches a specified regular expression. By default, the validator is disabled and must be created using the **dsconfig** tool. You also have the option to specify if the password should be accepted or rejected if it matches the regular expression.

### To Configure the Regular Expression Password Validator

1. Use **dsconfig** to create the Regular Expression password validator. The following password validator checks that the password contains at least one number, one lowercase letter, and one uppercase letter with no restrictions on password length. If the password matches the regular expression, then it will be accepted. When using the following command, remember to include the LDAP/LDAPS connection parameters (host name and port), bind DN, and bind password.

```
$ bin/dsconfig create-password-validator \  
--validator-name "Regular Expression" \  
--type regular-expression --set enabled:true \  
--set "match-pattern:^\w*(?=\w*\d)(?=\w*[a-z])(?=\w*[A-Z])\w*$" \  
--set match-behavior:require-match
```

2. Update an existing password policy to use the Regular Expression validator.

```
$ bin/dsconfig set-password-policy-prop \  
  --policy-name "Temp Password Policy" \  
  --set "password-validator:Regular Expression"
```

3. Test the Regular Expression Validator by submitting a password that meets the requirements (contains one number, one lowercase letter, and one uppercase letter), then run it again with a password that does not meet it.

```
$ bin/ldappasswordmodify \  
  --port 1389 --bindDN "cn=Directory Manager" --bindPassword secret \  
  --authzID "uid=user.0,ou=People,dc=example,dc=com" --newPassword baaA1  
The LDAP password modify operation was successful
```

Try another password. The following password should fail, because no uppercase letter is present.

```
$ bin/ldappasswordmodify \  
  --port 1389 --bindDN "cn=Directory Manager" --bindPassword secret \  
  --authzID "uid=user.0,ou=People,dc=example,dc=com" --newPassword baaa1
```

Error Message: The provided new password failed the validation checks defined in the server:  
The provided password is not acceptable because it does not match regular expression pattern  
'^\w\*(?=\w\*\d)(?=\w\*[a-z])(?=\w\*[A-Z])\w\*\$'

## Configuring the Repeated Character Password Validator

The Repeated Character Password Validator determines if a proposed password is acceptable or not by determining whether that password contains a specific number of repeated characters. By default, the validator checks if the password has at least two (2) repeated, case insensitive characters in a row.

Options are available to set the maximum number of repeated characters, case insensitivity, and the character set. For example, you can configure the validator to reject any password with more than three digits in a row.

### To Configure the Repeated Character Password Validator

1. Use **dsconfig** to edit the existing default configuration. In this example, we set the maximum consecutive length of any character to 3. For example, the following validator rejects any passwords, such as, "baaaal" or "4eeeb", etc.

```
$ bin/dsconfig set-password-validator-prop \  
  --validator-name "Repeated Characters" \  
  --set max-consecutive-length:3 \  
  --port 1389 --bindDN "cn=Directory Manager" \  
  --bindPassword secret --no-prompt
```

Or, you can create a validator that rejects any character from a pre-defined character set that appears more than the default number of times in a row (2). You can also specify more than one character set. For example, the following validator defines two characters sets:

[abc] and [123]. It rejects any passwords with more than 2 characters from a character set in a row. Thus, "aaa", "bbb", "ccc", "abc", or "123" and so on fails, but "12a3" is okay.

```
$ bin/dsconfig set-password-validator-prop \  
  --validator-name "Repeated Characters" \  
  --set character-set:123 --set character-set:abc \  
  --port 1389 --bindDN "cn=Directory Manager" \  
  --bindPassword secret --no-prompt
```

2. Update an existing password policy to use the Repeated Character Password Validator.

```
$ bin/dsconfig set-password-policy-prop \  
  --policy-name "Temp Password Policy" \  
  --set "password-validator:Repeated Characters" \  
  --port 1389 --bindDN "cn=Directory Manager" \  
  --bindPassword secret --no-prompt
```

3. Test the Repeated Character Validator by submitting a password that has more than the maximum allowable length of consecutive characters.

```
$ bin/ldapmodify --port 1389 \  
  --bindDN "cn=Directory Manager" \  
  --bindPassword secret \  
  --authzID "uid=user.0,ou=People,dc=example,dc=com" \  
  --newPassword baaal
```

```
The LDAP password modify operation failed with result code 53  
Error Message: The provided new password failed the validation checks defined in the server:  
The provided password contained too many instances of the same character appearing consecu-  
tively. The maximum number of times the same character may appear consecutively in a password  
is 2
```

## Configuring the Similarity-Based Password Validator

The Similarity-Based Password Validator determines if a proposed password is acceptable or not by determining how similar it is to the current password. By default, the validator checks if there are at least three (3) password changes (inserting, deleting, or replacing a character) between the proposed password and the current password.

### To Configure the Similarity-Based Password Validator

1. Use `dsconfig` to edit the existing default configuration. In this example, we set the minimum number of differences to 2.

```
$ bin/dsconfig set-password-validator-prop \  
  --validator-name "Similarity-Based Password Validator" \  
  --set min-password-difference:2 \  
  --port 1389 --bindDN "cn=Directory Manager" \  
  --bindPassword secret --no-prompt
```

2. Update an existing password policy to use the Similarity-Based Password Validator. The `password-change-requires-current-password` property must be set to `TRUE`, so that the password policy can gain access to the current user's password.

```
$ bin/dsconfig set-password-policy-prop \  
  --policy-name "Temp Password Policy" \  
  --set "password-validator:Similarity-Based Password Validator" \  
  --set password-change-requires-current-password:true \  
  --port 1389 --bindDN "cn=Directory Manager" \  
  --bindPassword secret --no-prompt
```

3. Test the Similarity-Based Password Validator by submitting a password that has less than the number of changes (e.g., 2). The `ldappasswordmodify` command requires the `--currentPassword` option when testing the Similarity-Based Password Validator.

```
$ bin/ldappasswordmodify --port 1389 \  
  --bindDN "cn=Directory Manager" \  
  --bindPassword secret \  
  --authzID "uid=user.0,ou=People,dc=example,dc=com" \  
  --currentPassword abcde --newPassword abcd
```

The LDAP password modify operation failed with result code 49

## Configuring the Unique Characters Password Validator

The Unique Characters Password Validator determines if a proposed password is acceptable or not by determining if it has the minimum number of unique characters. By default, the validator checks if there are a minimum of five (5) unique characters (case insensitive).

### To Configure the Unique Characters Password Validator

1. Use `dsconfig` to edit the existing default configuration. In this example, we set the minimum number of unique characters that a password is allowed to contain to 2.

```
$ bin/dsconfig set-password-validator-prop \  
  --validator-name "Similarity-Based" \  
  --set min-unique-characters:2 \  
  --port 1389 --bindDN "cn=Directory Manager" \  
  --bindPassword secret --no-prompt
```

2. Update an existing password policy to use the Unique Characters Password Validator.

```
$ bin/dsconfig set-password-policy-prop \  
  --policy-name "Temp Password Policy" \  
  --set "password-validator:Unique Characters" \  
  --port 1389 --bindDN "cn=Directory Manager" \  
  --bindPassword secret --no-prompt
```

3. Test the Unique Characters Password Validator by submitting a password that has less than the minimum number of unique characters (e.g., 2).

```
$ bin/ldappasswordmodify --port 1389 \  
  --bindDN "cn=Directory Manager" \  
  --bindPassword secret \  
  --authzID "uid=user.0,ou=People,dc=example,dc=com" \  
  --newPassword aaaaa
```

The LDAP password modify operation failed with result code 53  
Error Message: The provided new password failed the validation checks  
defined in the server: The provided password does not contain enough  
unique characters. The minimum number of unique characters that may appear  
in a user password is 2

---

# 6

## Managing Access Control

The UnboundID® Directory Proxy Server provides a fine-grained access control model to ensure that users are able to access the information they need, but are prevented from accessing information that they should not be allowed to see. It also includes a privilege subsystem that provides even greater flexibility and protection in many key areas.

This chapter presents the access control model and how it applies to the directory proxy server.

- [Working with Access Control in the Proxy Server](#)
- [Configuring Access Control with Entry Balancing](#)
- [Configuring Access Control](#)
- [Working with Privileges](#)

### Working with Access Control in the Proxy Server

Access controls enforce access wherever data is stored. For accesses through the proxy server to backend directory server data, access control is enforced on the directory server level. If you configure global access controls in the directory proxy server for data stored on backend directory servers, they will not be enforced. However, you can configure global access controls in the directory proxy server to control data stored on the directory proxy servers, including schema, configuration, monitoring and alert information, and administrative data. You can configure what subtree view clients can see and the operations they can perform at a coarse level using client connection policies. For more information about client connection policies, see “Configuring Client Connection Policies” on page 92.

The access control model uses access control instructions (ACIs), which are stored in the `aci` operational attribute, to determine what a user or a group of users can do with a set of entries, down to the attribute level. The operational attribute can appear on any entry and will affect the entry or any subentries within that branch of the directory information tree (DIT).

## Configuring Access Control with Entry Balancing

The directory servers in each backend server set should contain a copy of all the access control rules applicable to the branch of the tree under the entry balancing point, or the entry balancing base entry. These access controls should be stored in ACI attributes at or above the entry balancing point. The directory proxy server ensures that any changes to entries within the scope of the entry balancing request processor, but outside the balancing point, are applied to all backend server sets. This means, for example, that any ACI stored at the entry balancing point will be kept in sync if changes are made through the directory proxy server.

Entry balancing introduces a possible issue when clients to the directory proxy server authenticate as users whose entries are among those balanced. The access control rules may require access to the entry contents of the user issuing the request. By its nature, entry balancing dictates that servers in only one of the backend sets contain the entry for a given user. If the server processing a request does not contain the issuing user's entry, then the access control cannot be evaluated. As a solution to this problem, the directory proxy server allows an alternate authorization identity to be defined for any user, where the alternate identity is the DN of an entry that exists in all directory servers in all backend sets. The alternate authorization identity is used when the directory proxy server observes that the directory server processing a request does not contain the entry of the user issuing the request.

To use alternate authorization identities, specify the name of an attribute in user entries to hold the alternate DN. This attribute is specified by the **authz-attribute** property of the entry balancing request processor configuration. This advanced property has a default value of **ds-authz-map-to-dn**, an attribute reserved for this purpose.

Next, define one or more generic users to be used as alternate authorization identities. For example, suppose the entry balancing point is the **ou=people** entry under **dc=example,dc=com**. You could create the following users:

```
uid=normal-user,dc=example,dc=com
uid=server-admin,dc=example,dc=com
uid=password-admin,dc=example,dc=com
```

Note that there must be a copy of each generic user in all of the backend directory servers. Make sure to give these generic users the set of rights that should be granted to those different classes of user.

Finally, every user among the balanced entries should be assigned one of these generic users as their alternate authorization identity. The directory server schema must allow the chosen attribute for alternate authorization to be present in user entries. For example, you could create an auxiliary object class containing **ds-authz-map-to-dn** as an allowed attribute, and add this object class value to all user entries. Then, you could explicitly add the following attribute value to a **server-admin** user:

```
ds-authz-map-to-dn: uid=server-admin,dc=example,dc=com
```

Alternatively, the attribute could be supplied as a virtual attribute definition.



If any user among the balanced entries does not have an alternate authorization identity defined, the proxy server will instead use the value of the `authz-dn` property of the entry balancing request processor configuration, if provided. For example, the following command could be used to specify that users are “normal” users by default (and thus avoid having to explicitly assign each of them an alternate identity).

```
dsconfig set-request-processor-prop
--processor-name dc_example_dc_com-eb-req-processor
--set "authz-dn:uid=normal-user,dc=example,dc=com"
```

Suppose you have set things up such that `uid=user.0,ou=people,dc=example,dc=com` is a normal user. If an operation performed by this user is routed to a directory server containing this user entry, then you will see

`authzDN="uid=user.0,ou=People,dc=example,dc=com"` in the access log for the operation in that directory server. The sample output has been wrapped for readability.

```
ldapsearch -D "uid=user.0,ou=people,dc=example,dc=com" -w password
-b uid=user.0,ou=people,dc=example,dc=com "(objectclass=*)" "
```

```
[15/Jun/2009:11:51:23 -0500] BIND REQUEST conn=171 op=1 msgID=2
version="3" dn="uid=user.0,ou=people,dc=example,dc=com"
authType="SIMPLE"
```

```
[15/Jun/2009:11:51:23 -0500] BIND RESULT conn=171 op=1 msgID=2
resultCode=0 etime=1.826 authDN="uid=user.0,ou=People,
dc=example,dc=com"
```

```
[15/Jun/2009:11:51:23 -0500] SEARCH REQUEST conn=172 op=1 msgID=2
via="app='UnboundID-Proxy' address='127.0.0.1'
authzID='dn:uid=user.0,ou=people,dc=example,dc=com'
sessionID='conn=1' requestID='op=1'" base="uid=user.0,
ou=people,dc=example,dc=com" scope=2 filter="(objectclass=*)" "
attrs="ALL"
```

```
[15/Jun/2009:11:51:23 -0500] SEARCH RESULT conn=172 op=1 msgID=2
resultCode=0 etime=3.101 entriesReturned=1
authzDN="uid=user.0,ou=People,dc=example,dc=com"
```

However, if an operation performed while bound as that same user is routed to a directory server that does not contain the user entry, then you will see `authzID='dn:uid=normal-user,dc=example,dc=com'` in the directory server log, indicating that the alternate authorization identity was used. In this example, it is assumed that `user.1` is in a different backend set from `user.0`:

```
ldapsearch -D "uid=user.0,ou=people,dc=example,dc=com" -w password
-b uid=user.1,ou=people,dc=example,dc=com "(objectclass=*)" "
```

```
[15/Jun/2009:11:54:35 -0500] SEARCH REQUEST conn=153 op=1 msgID=2
via="app='UnboundID-Proxy' address='127.0.0.1'
authzID='dn:uid=normal-user,dc=example,dc=com' sessionID='conn=2'
requestID='op=1'" base="uid=user.1,ou=people,dc=example,dc=com"
scope=2 filter="(objectclass=*)" attrs="ALL"
```

```
[15/Jun/2009:11:54:35 -0500] SEARCH RESULT conn=153 op=1 msgID=2  
resultCode=0 etime=2.038 entriesReturned=1  
authzDN="uid=normal-user,dc=example,dc=com"
```

In the above output, you do not see the bind operation in the log because that operation was performed on the other server.

## Configuring Access Control

This section describes the key features of access controls and the keywords supported.

### Key Access Control Features

The UnboundID Directory Proxy Server provides important access control features that provide added security for the directory proxy server's entries.

### Improved Validation and Security

The directory proxy server provides an access control model with strong validation to help ensure that invalid ACIs are not allowed into the server. For example, the directory proxy server ensures that all access control rules added over LDAP are valid and can be fully parsed. Any operation that attempts to store one or more invalid ACIs are rejected. The same validation is applied to ACIs contained in data imported from an LDIF file. Any entry containing a malformed ACI value will be rejected.

As an additional level of security, the directory proxy server examines and validates all ACIs stored in the data whenever a backend is brought online. If any malformed ACIs are found in the backend, then the server generates an administrative alert to notify administrators of the problem and places itself in lockdown mode. While in lockdown mode, the server only allows requests from root users and rejects any requests received from non-root users. This action allows administrators to correct the malformed ACI while ensuring that no sensitive data is inadvertently exposed due to an access control instruction not being enforced. When the problem has been corrected, the administrator can use the `leave-lockdown-mode` tool or restart the server to allow it to resume normal operation.

### Global ACIs

Global ACIs (also called default ACIs) are a set of ACIs that can apply to entries anywhere in the directory (although they can also be in scope so that they only apply to a specific set of entries). They work in conjunction with access control rules stored in user data and provide a convenient way to define ACIs that span disparate portions of the DIT.

In the UnboundID Directory Proxy Server, global ACIs are defined within the server configuration, in the `global-aci` property of configuration object for the access control handler. They

can be viewed and managed using configuration tools like **dsconfig** and the web administration console.

The global ACIs available by default in the UnboundID Directory Proxy Server include:

- Allow anyone (including unauthenticated users) to access key attributes of the root DSE, including **namingContexts**, **subschemaSubentry**, **supportedAuthPasswordSchemes**, **supportedControl**, **supportedExtension**, **supportedFeatures**, **supportedLDAPVersion**, **supportedSASLMechanisms**, **vendorName**, and **vendorVersion**.
- Allow anyone (including unauthenticated users) to access key attributes of the subschema subentry, including **attributeTypes**, **dITContentRules**, **dITStructureRules**, **ldapSyntaxes**, **matchingRules**, **matchingRuleUse**, **nameForms**, and **objectClasses**.
- Allow anyone (including unauthenticated users) to include the following controls in requests made to the server: authorization identity request, manage DSA IT, password policy, real attributes only, and virtual attributes only.
- Allow anyone (including unauthenticated users) to request the following extended operations: get symmetric key, password modify request, password policy state, StartTLS, and Who Am I?

## Target Attributes

The target attributes for an access control instruction are provided using the **targetattr** keyword. There are three general forms that it can take in the UnboundID Directory Proxy Server:

- **(targetattr="\*)**. Indicates that the access control rule applies to all attributes.
- **(targetattr="attr1 | attr2 | attr3 | ... | attrN")**. Indicates that the access control rule applies only to the named set of attributes.
- **(targetattr!="attr1 | attr2 | attr3 | ... | attrN")**. Indicates that the access control rule applies to all attributes except the named set of attributes.

The targeted attributes can be classified as *user attributes* and *operational attributes*. User attributes define the actual data for that entry, while operational attributes provide additional metadata about the entry that can be used for informational purposes (for example, when the entry was created or last modified and by whom) or operational purposes (for example, specifying which password policy applies to the user, or overriding default constraints like size limit, time limit, or look-through limit for that user).

The UnboundID Directory Proxy Server distinguishes between these two types of attributes in its implementation of access control. The directory proxy server does not automatically grant any access at all to operational attributes. For example, a clause like:

```
(targetattr="*)
```

applies only to user attributes and not to operational attributes, which is almost always the intended effect. If there is a legitimate need for a user to be able to access an operational attribute, then it can be specifically included in the values of the `targetattr` clause, as follows:

```
(targetattr="ds-rlim-size-limit") or  
(targetattr="* | ds-rlim-size-limit")
```

You can use the "+" symbol to indicate that the rule should apply to all operational attributes, as follows:

```
(targetattr="+")  
or  
(targetattr="* | +")
```

The implications of the directory proxy server not distinguishing between user attributes and operational attributes can be outlined in an example. It can be easy to inadvertently create an access control instruction that grants far more capabilities to a user than originally intended. Consider the following example:

```
aci: (targetattr!="uid | employeeNumber") (version 3.1; acl "Allow  
users to update their own entries"; allow (write) userdn="ldap:///self";)
```

This instruction is intended to allow a user to update any attribute in his or her own entry with the exception of `uid` and `employeeNumber`. This ACI is a very common type of rule and seems relatively harmless on the surface, but it has very serious consequences for a directory proxy server that does not distinguish between user attributes and operational attributes. It allows users to update operational attributes in their own entries, and could be used for a number of malicious purposes, including:

- A user could alter password policy state attributes to become exempt from password policy restrictions.
- A user could alter resource limit attributes and bypass size limit, time limit, and look-through-limit constraints.
- A user could add access control rules to his or her own entry, which could allow them to make their entry completely invisible to all other users (including administrators, but excluding the root user), allow them to edit any other attributes in their own entry (including those excluded by rules like `uid` and `employeeNumber` in the example above), or add, modify, or delete any entries below his or her own entry.

## Target Scope

The `targetscope` keyword can be used to restrict the scope of an access control rule. By default, an ACI is applied to the target entry (either as defined by the `target` clause of the ACI, or the entry in which the ACI is defined, if it does not include a `target`) and all entries below it. However, adding the `targetscope` element into an access control rule can restrict the set of entries to which it applies. The following `targetscope` keyword values are allowed:

- **base.** Indicates that the access control rule should apply only to the target entry and not to any of its subordinate.
- **onelevel.** Indicates that the access control rule should apply only to entries that are immediate children of the target entry, but not to any of their subordinates and to the target entry itself.
- **subtree.** Indicates that the access control rule should apply to the target entry and all of its subordinates. This is the default behavior if no targetscope is specified.
- **subordinate.** Indicates that the access control rule should apply to all entries below the target entry but not the target entry itself.

## Authentication Method

The `authmethod` keyword can be used to indicate that an access control rule should be applied based on the way in which the user authenticated to the server. The `authmethod` keyword allows the following values:

- **none**. Indicates that the rule should apply to clients that have not attempted to authenticate, that performed an anonymous bind (for example, using simple authentication with a zero-length DN and password), or whose last bind attempt was not successful.
- **simple**. Indicates that the rule should apply to clients that authenticated using simple authentication with a non-empty DN and password.
- **ssl**. Indicates that the rule should apply to clients that authenticated using a client certificate via the SASL EXTERNAL mechanism.
- **sasl <sasl\_mechanism>**. Indicates that the rule should apply to clients that authenticated using any SASL mechanism. You can restrict the SASL authentication to a particular mechanism by including its name. For example, “sasl DIGEST-MD5” indicates that the rule should only apply to clients that authenticated using the DIGEST-MD5 SASL mechanism.

## Use of Controls and Extended Operations

The UnboundID Directory Proxy Server adds new keywords that supports access control to request controls and extended operations. The `targetcontrol` keyword can be used to indicate whether a given request control is used. The `extop` keyword can be used to indicate whether a given extended operation can be used. For both keywords, the value is the object identifier for the associated control or extended operation. Multiple OIDs can be provided by separating them with the two pipe characters (optionally surrounded by spaces). For example:

- `(targetcontrol="2.16.840.1.113730.3.4.2")`
- `(extop="1.3.6.1.4.1.4203.1.11.1 || 1.3.6.1.4.1.1466.20037")`

---

|             |                                                                                                                                                                                                                        |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Note</b> | When assigning ACIs to a user using the <code>targetcontrol</code> or <code>extop</code> keywords, you must define a specific ACI for any of the controls or extended operations, and then assign an ACI for the user. |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

## Groups

The UnboundID Directory Proxy Server provides enhanced support for groups. A separate role implementation is not required with a specific keyword.

## Summary of Access Control Keywords

This section provides an overview of the keywords supported for use in the UnboundID Directory Proxy Server access control implementation.

### Target Keywords

The following keywords are supported for use in the target portion of ACIs to indicate the set of entries and/or operations to which an access control rule applies:

- **extop**. Specifies the object identifiers of one or more extended requests.
- **target**. Specifies one or more LDAP URLs for a set of entries.
- **targattrfilters**. Specifies a set of filters that can be used to indicate whether particular attribute values are added to or removed from an entry.
- **targetattr**. Specifies one or more attributes.
- **targetcontrol**. Specifies the object identifiers of one or more request controls.
- **targetfilter**. Specifies one or more search filters to indicate a set of entries.
- **targetscope**. Specifies the scope for a set of entries.

### Permissions

The following keywords are supported for use in the permissions portion of ACIs to indicate the type of operations to which an access control rule might apply:

- **add**. Indicates that the access control should apply to add operations.
- **compare**. Indicates that the access control should apply to compare operations.
- **delete**. Indicates that the access control should apply to delete operations.
- **proxy**. Indicates that the access control rule should apply to operations that attempt to use an alternate authorization identity (for example, operations that include a proxied authorization request control).
- **read**. Indicates that the access control rule should apply to search result entries returned by the server.
- **search**. Indicates that the access control rule should apply to search operations.
- **selfwrite**. Indicates that the access control rule should apply to operations in which a user attempts to add his or her own DN to the set of members for a static group.

- **write**. Indicates that the access control rule should apply to modify and modify DN operations.
- **all**. An aggregate permission that includes all other permissions except “proxy”. This is equivalent to providing a permission of “add, compare, delete, read, search, selfwrite, write”.

## Bind Rule Keywords

The following keywords are supported for use in the bind rule portion of ACIs to indicate whether an access control rule should apply to a given requester:

- **authmethod**. Indicates that the requester’s authentication method should be taken into account when determining whether the access control rule should apply to an operation. The keyword’s syntax is as follows:

**authmethod** = *method*

where:

*method* = none, simple, ssl, sasl {sasl\_mechanism}

Wildcards are not allowed in this expression.

- **dayofweek**. Indicates that the day of the week should be taken into account when determining whether the access control rule should apply to an operation. The keyword’s syntax is as follows:

**dayofweek** = *day*

where:

*day* = sun, mon, tues, wed, thu, fri, sat

Wildcards are not allowed in this expression.

- **dns**. Indicates that the requester’s DNS-resolvable hostname should be taken into account when determining whether the access control rule should apply to an operation. The keyword’s syntax is as follows:

**dns** = *dns-host-name*

Wildcards are allowed in this expression.

- **groupdn**. Indicates that the requester’s group membership should be taken into account when determining whether the access control rule should apply to any operation. The keyword’s syntax is as follows:

**groupdn** [ = || != ] “ldap:///groupdn [ || ldap:///groupdn ] ...”

Wildcards are not allowed in this expression.



- **ip**. Indicates that the requester's IP address should be taken into account when determining whether the access control rule should apply to an operation. The keyword's syntax is as follows:

**ip** [ = | | != ] *ipAddressList*

where:

*ipAddressList* is one of the following representations:

- A specific IPv4 address: 127.0.0.1
- An IPv4 address with wildcards to specify a subnetwork: 127.0.0.\*
- An IPv4 address or subnetwork with subnetwork mask: 123.4.5.\*+255.255.255.192
- An IPv6 address as defined by RFC 2373.

- **timeofday**. Indicates that the time of day should be taken into account when determining whether the access control rule should apply to an operation. The keyword's syntax is as follows:

**timeofday** [ = || != || > || >= || < ] *time*

where:

*time* = 4-digit 24-hour time format (0 to 2359)

Wildcards are not allowed in this expression.

- **userattr**. Indicates that the requester's relation to the value of the specified attribute should be taken into account when determining whether the access control rule should apply to an operation. The keyword's syntax is as follows:

**userattr** = "*attrName*# [ *bindType* || *attrValue* ]"

where:

*attrName* = name of the attribute for matching

*bindType* = **USERDN**, **GROUPDN**, **LDAPURL**

*attrValue* = an attribute value

Wildcards are not allowed in this expression.

- **userdn**. Indicates that the user's DN should be taken into account when determining whether the access control rule should apply to an operation. The keyword's syntax is as follows:

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

Wildcards are allowed in the expression.

## Working with Privileges

In addition to the access control implementation, the UnboundID Directory Proxy Server includes a privilege subsystem that can also be used to control what users are allowed to do.

The privilege subsystem works in conjunction with the access control subsystem so that privileged operations are only allowed if they are allowed by the access control configuration and the user has all of the necessary privileges.

Privileges can be used to grant normal users the ability to perform certain tasks that, in most other directories, would only be allowed for the root user. In fact, the capabilities extended to root users in the UnboundID Directory Proxy Server are all granted through privileges, so you can create a normal user account with the ability to perform all of the same actions as root users. At present, the only special treatment that root users get as compared with a non-root user who has the equivalent set of privileges is that only root users can interact with the server when it is in lockdown mode.

Administrators can remove privileges from root users so that they are unable to perform certain types of operations. Multiple root users can be defined in the server with different sets of privileges so that the capabilities that they have are restricted to only the tasks that they need to be able to perform.

## Available Privileges

The following privileges have been defined in the UnboundID Directory Proxy Server.

**TABLE 8-1. Summary of Privileges**

| Privilege        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| backend-backup   | This privilege is required to initiate an online backup through the tasks interface. The server's access control configuration must also allow the user to add the corresponding entry in the tasks backend.                                                                                                                                                                                                                                                                                                                                                     |
| backend-restore  | This privilege is required to initiate an online restore through the tasks interface. The server's access control configuration must also allow the user to add the corresponding entry in the tasks backend.                                                                                                                                                                                                                                                                                                                                                    |
| bypass-acl       | This privilege allows a user to bypass access control evaluation. For a user with this privilege, any access control determination made by the server immediately returns that the operation is allowed. Note, however, that this does not bypass privilege evaluation, so the user must have the appropriate set of additional privileges to be able to perform any privileged operation (for example, a user with the <code>bypass-acl</code> privilege but without the <code>config-read</code> privilege is not allowed to access the server configuration). |
| bypass-pw-policy | This privilege allows a user entry to bypass password policy evaluation. This privilege is intended for cases where external synchronization might require passwords that violate the password validation rules. The privilege is not evaluated for bind operations so that password policy evaluation will still occur.                                                                                                                                                                                                                                         |
| bypass-read-acl  | This privilege allows the associated user to bypass access control checks performed by the server for bind, search, and compare operations. Access control evaluation may still be enforced for other types of operations.                                                                                                                                                                                                                                                                                                                                       |
| config-read      | This privilege is required for a user to access the server configuration. Access control evaluation is still performed and can be used to restrict the set of configuration objects that the user is allowed to see.                                                                                                                                                                                                                                                                                                                                             |

TABLE 8-1. Summary of Privileges

| Privilege         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| config-write      | This privilege is required for a user to alter the server configuration. The user is also required to have the <code>config-read</code> privilege. Access control evaluation is still performed and can be used to restrict the set of configuration objects that the user is allowed to alter.                                                                                                                                                          |
| disconnect-client | This privilege is required for a user to request that an existing client connection be terminated. The connection is terminated through the disconnect client task. The server's access control configuration must also allow the user to add the corresponding entry to the tasks backend.                                                                                                                                                              |
| jmx-notify        | This privilege is required for a user to subscribe to JMX notifications generated by the directory proxy server. The user is also required to have the <code>jmx-read</code> privilege.                                                                                                                                                                                                                                                                  |
| jmx-read          | This privilege is required for a user to access any information provided by the directory proxy server via the Java Management Extensions (JMX).                                                                                                                                                                                                                                                                                                         |
| jmx-write         | This privilege is required for a user to update any information exposed by the directory proxy server via the Java Management Extensions (JMX). The user is also required to have the <code>jmx-read</code> privilege. Note that currently all of the information exposed by the server over JMX is read-only.                                                                                                                                           |
| ldif-export       | This privilege is required to initiate an online LDIF export through the tasks interface. The server's access control configuration must also allow the user to add the corresponding entry in the tasks backend.                                                                                                                                                                                                                                        |
| ldif-import       | This privilege is required to initiate an online LDIF import through the tasks interface. The server's access control configuration must also allow the user to add the corresponding entry in the tasks backend.                                                                                                                                                                                                                                        |
| lockdown-mode     | This privilege allows the associated user to request that the server enter or leave lockdown mode, or to perform operations while the server is in lockdown mode.                                                                                                                                                                                                                                                                                        |
| modify-acl        | This privilege is required for a user to add, modify, or remove access control rules defined in the server. The server's access control configuration must also allow the user to make the corresponding change to the <code>aci</code> operational attribute.                                                                                                                                                                                           |
| password-reset    | This privilege is required for a user to change another user's password. The server's access control configuration must also allow the corresponding change to the attribute containing the user's password.                                                                                                                                                                                                                                             |
| privilege-change  | This privilege is required for a user to change a user's set of privileges. The server's access control configuration must also allow the user to make the corresponding change to the <code>ds-privilege-name</code> operational attribute.                                                                                                                                                                                                             |
| proxied-auth      | This privilege is required for a user to request that an operation be performed with an alternate authorization identity. This privilege applies to operations that include the proxied authorization V1 or V2 control, operations that include the intermediate client request control with a value set for the client identity field, or for SASL bind requests that can include an authorization identity different from the authentication identity. |
| server-restart    | This privilege is required to initiate a server restart through the tasks interface. The server's access control configuration must also allow the user to add the corresponding entry in the tasks backend.                                                                                                                                                                                                                                             |
| server-shutdown   | This privilege is required to initiate a server shutdown through the tasks interface. The server's access control configuration must also allow the user to add the corresponding entry in the tasks backend.                                                                                                                                                                                                                                            |

TABLE 8-1. Summary of Privileges

| Privilege        | Description                                                                                                                                                                                                                                                                                                                                                           |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| stream-values    | This privilege is required for a user to perform a stream values extended operation, which obtains all entry DNs and/or all values for one or more attributes for a specified portion of the DIT.                                                                                                                                                                     |
| unindexed-search | This privilege is required for a user to be able to perform a search operation in which a reasonable set of candidate entries cannot be determined using the defined es. Instead, a significant portion of the database needs to be traversed to identify matching entries. The server's access control configuration must also allow the user to request the search. |
| update-schema    | This privilege is required for a user to modify the server schema. The server's access control configuration must allow the user to update the operational attributes that contain the schema elements.                                                                                                                                                               |

## Privileges Automatically Granted to Root Users

As previously mentioned, the special abilities that root users have are generally granted through privileges. Privileges can be assigned to root users in two ways:

- All root users are automatically granted a default set of privileges
- Individual root users can have additional privileges granted to them, or they can have default privileges removed.

The set of privileges that are automatically granted to root users is controlled by the `default-root-privilege-name` property of the `Root` DN configuration object. By default, this set of privileges includes:

- backend-backup
- backend-restore
- bypass-acl
- config-read
- config-write
- disconnect-client
- ldif-export
- ldif-import
- lockdown-mode
- modify-acl
- password-reset
- privilege-change
- server-restart
- server-shutdown
- stream-values
- unindexed-search
- update-schema

The privileges not granted to root users by default includes:

- bypass-read-acl
- jmx-read

- `jmx-write`
- `jmx-notify`
- `proxied-auth`
- `bypass-pw-policy`

The set of default root privileges can be altered to add or remove values as necessary. Doing so will require the `config-read`, `config-write`, and `privilege-change` privileges, as well as either the `bypass-acl` privilege or sufficient permission granted by the access control configuration to make the change to the server's configuration.

## Assigning Privileges to Normal Users and Individual Root Users

Privileges can be granted to normal users on an individual basis. This can be accomplished by adding the `ds-privilege-name` operational attribute to that user's entry with the names of the desired privileges. For example, the following change will grant the `proxied-auth` privilege to the `uid=proxy,dc=example,dc=com` account:

```
dn: uid=proxy,dc=example,dc=com
changetype: modify
add: ds-privilege-name
ds-privilege-name: proxied-auth
```

The user making this change will be required to have the `privilege-change` privilege, and the server's access control configuration must also allow the requester to write to the `ds-privilege-name` attribute in the target user's entry.

This same method can be used to grant privileges to root users that they would not otherwise have through the set of default root privileges. You can also remove default root privileges from root users by prefixing the name of the privilege to remove with a minus sign. For example, the following change grants a root user the `jmx-read` privilege in addition to the set of default root privileges, and removes the `server-restart` and `server-shutdown` privileges:

```
dn: cn=Sync Root User,cn=Root DNs,cn=config
changetype: modify
add: ds-privilege-name
ds-privilege-name: jmx-read
ds-privilege-name: -server-restart
ds-privilege-name: -server-shutdown
```

Note that because root user entries exist in the configuration, this update requires the `config-read` and `config-write` privileges in addition to the `privilege-change` privilege.

## Disabling Privileges

Although the privilege subsystem in the UnboundID Directory Proxy Server is a very powerful feature, it might break some applications if they expect to perform some operation that requires a privilege that they do not have. In the vast majority of these cases, you can work around the problem by simply assigning the necessary privilege manually to the account used by that application. However, if this workaround is not sufficient, or if you need to remove a

particular privilege (for example, to allow anyone to access information via JMX without requiring the `jmx-read` privilege), then privileges can be disabled on an individual basis.

The set of disabled privileges is controlled by the `disabled-privilege` property in the global configuration object. By default, no privileges are disabled. If a privilege is disabled, then the server behaves as if all users have that privilege.

---

# 7

## Managing Replication in an Entry Balancing Deployment

In a standard, replicated UnboundID Directory Server deployment, multiple directories are fully connected and replicating with one another. However, in an entry-balancing deployment, the data is not shared on all servers. Data above the balancing point is replicated across all servers, and the data below the entry balancing point is present only on certain servers.

This chapter presents the following information about replication in an entry-balanced environment:

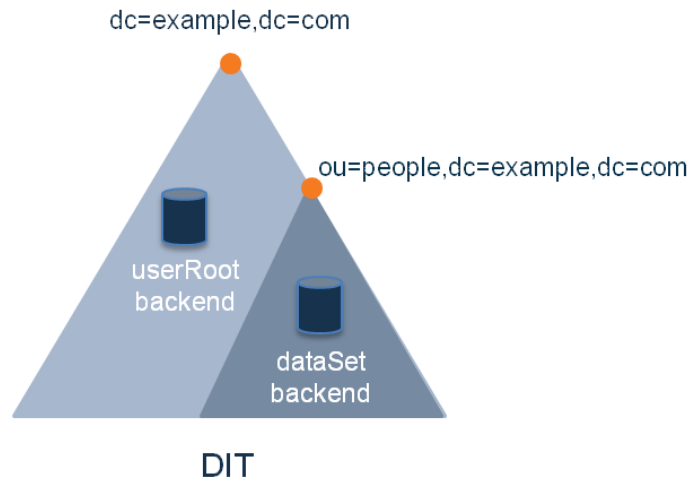
- [Overview of Replicating in an Entry Balanced Environment](#)
- [Replication Prerequisites in an Entry-Balanced Deployment](#)
- [Example of Configuring Entry-Balancing Replication](#)
- [Detaching a Replication Set from a Topology](#)

### Overview of Replicating in an Entry Balanced Environment

In an entry-balanced deployment, some data is replicated everywhere, such as the schema, the server registry, and other shared data, and some data is replicated only on certain servers. A replication domain contains all of the servers in a replicated topology and shares a schema. The replication domain is associated with the base DN and must be a base DN of a backend.

By default, replication propagates updates to all replication servers in the topology. Updates to data under the entry-balancing point, however, must be replicated only among server instances in the same data set. Replication requires that, in such deployments, the directory server is configured with a replication set name global configuration property, and two backends. One backend has a base DN that is replicated globally (such as `dc=example,dc=com`)

and the second backend has a base DN associated with the entry-balancing point (such as `ou=people,dc=example,dc=com`).



If a data set name is not defined when you set up the proxy server, one will be provided by default. The proper configuration of an entry-balanced environment requires coordination between the directory server and proxy server. Once replication is enabled, the replication domain may be designated as the domain participating in entry balancing.

## Replication Prerequisites in an Entry-Balanced Deployment

Replication in an entry-balanced deployment requires the following:

- **Multiple local DB backends.** When you set up the directory instances, you need two backends, a global backend for globally replicated data, such as `userRoot`, and a restricted backend for the balancing point base DN, `dataSet`. Both backends, the global and the restricted, need to be initialized separately. Depending on the size of the dataset, you can initialize the backends using the `dsreplication` tool, the `import-ldif` tool, or binary copy. See the UnboundID Directory Server Administration Guide for more information about initializing replication data.
- **Replication set name.** Every directory server in your replicated topology must have a replication set name. This replication set name coordinates the proxy server and the directory server. The restricted domain is only replicated within instances using the same replication set name.
- **Multiple proxy server subtree views.** The entry-balanced proxy configuration relies on multiple subtree views, one for the globally replicated base DN and one for the entry-balancing point base DN. The globally replicated base DN will have a proxying request processor associated with it. The restricted base DN will have an entry-balancing request processor associated with it. This configuration is best achieved using the `create-initial-proxy-config` tool after running `setup`.



## About the --restricted Argument of the dsreplication Command-Line Tool

When enabling replication for a server that takes part in an entry balanced environment it is recommended that the multiple domains involved are enabled at the same time. There is a global domain, and a restricted domain, where the restricted domain represents the entry-balancing point. Each base DN is defined in a separate Local DB Backend. The dsreplication CLI tool has a --restricted argument that is used to specify which base DN is considered an entry balancing point. The following example displays how you can enable replication between two servers with entry balancing. Notice that the command includes multiple occurrences of the base DN argument.

```
$ bin/dsreplication enable --host1 host1.example.com \  
--port1 1389 --bindDN1 "cn=Directory Manager" \  
--bindPassword1 secret --replicationPort1 8989 \  
-- host2 host2.example.com --port2 2389 \  
--bindDN2 "cn=Directory Manager" --bindPassword2 secret \  
--replicationPort2 8989 --baseDN dc=example,dc=com \  
--baseDN ou=people,dc=example,dc=com \  
--restricted ou=people,dc=example,dc=com
```

Alternatively, you can enable replication using the interactive command line, making sure to specify that an entry balancing is being used and specifying the base DN of the entry-balancing point.

```
You must choose at least one base DN to be replicated.  
  
Replicate base DN dc=example,dc=com? (yes / no) [yes]: yes  
  
Replicate base DN ou=people,dc=example,dc=com? (yes / no)  
[yes]: yes  
  
Do you plan to configure entry balancing using the Proxy  
Server? (yes / no) [no]: yes  
  
Is dc=example,dc=com an entry-balancing point? (yes / no) [no]:  
no  
  
Is ou=people,dc=example,dc=com an entry-balancing point? (yes /  
no) [no]: yes
```

## Checking the Status of Replication in an Entry-Balanced Deployment

You can use the `dsreplication status` tool to check the status of an entry-balanced deployment. In this example, the `ou=people,dc=example,dc=com` subtree is entry-balanced. The data is split into two sets, `set1` and `set2`. The servers `host1` and `host2` are in replication set `set1` and servers `host3` and `host4` are in replication set `set2`.

The output shows information for the two sets as follows:

```
--- Replication Status for dc=example,dc=com: Enabled ---  
  
Server      : Entries: Replication Backlog: Oldest Backlog Change Age : Port :Security  
-----:-----:-----:-----:-----:-----:-----
```

```
host1.example.com:1389 : 1000      : 0          : N/A          : 8989 : Enabled
host2.example.com:2389 : 1000      : 0          : N/A          : 8989 : Enabled
host3.example.com:3389 : 1000      : 0          : N/A          : 8989 : Enabled
host4.example.com:4389 : 1000      : 0          : N/A          : 8989 : Enabled

-- Replication Status for ou=people,dc= example,dc=com (Set: set1): Enabled --

Server      : Entries: Replication Backlog: Oldest Backlog Change Age : Port :Security
-----:-----:-----:-----:-----:-----:-----

host1.example.com:1389 : 1000000 : 0          : N/A          : 8989 : Enabled
host2.example.com:2389 : 1000000 : 0          : N/A          : 8989 : Enabled

---Replication Status for ou=people,dc= example,dc=com (Set: set2): Enabled ---

Server      : Entries: Replication Backlog: Oldest Backlog Change Age : Port :Security
-----:-----:-----:-----:-----:-----:-----

host3.example.com:3389 : 1000000 : 0          : N/A          : 8989 : Enabled
host4.example.com:4389 : 1000000 : 0          : N/A          : 8989 : Enabled
```

You can use the `--setName` argument to see only the status for a specific replication set or sets. For example, use the following command line to see the status of only set `set1`:

```
$ bin/dsreplication status --hostname host1.example.com --port
1389 --adminUID admin --adminPassword secret --setName set1
```

## Example of Configuring Entry-Balancing Replication

This section describes how to set up a four-server multi-master replication topology that uses entry balancing to distribute entries across the servers. The procedure assumes that none of the servers have participated in any previous replication topology. Also, this procedure uses the binary copy method for initializing data into the replicas, which is recommended for production environments.

### Assumptions

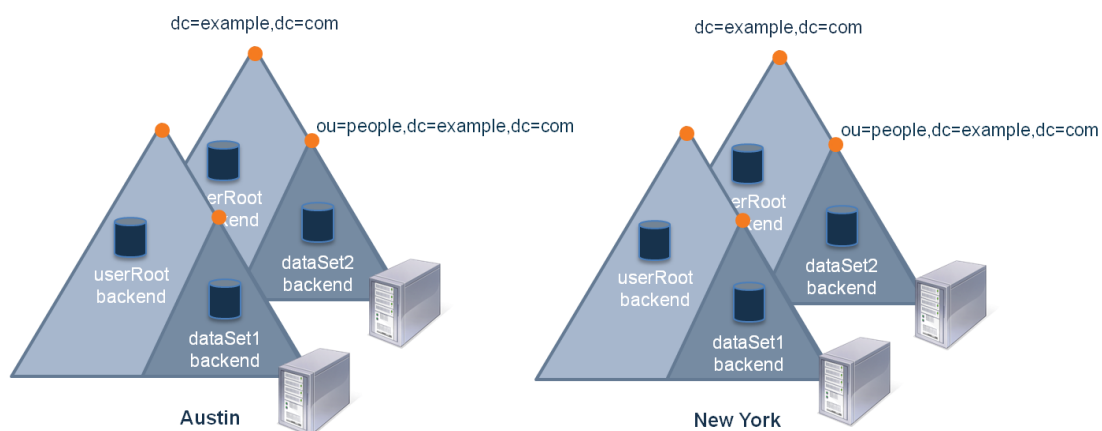
The following procedures assume that all directory server instances are built from scratch and are solely dedicated as directory servers. The example uses the LDAP (389) and replication (8989) ports respectively. It configures the following hosts:

- `replica1 -- austin-set1.example.com`
- `replica2 -- newyork-set1.example.com`
- `replica3 -- austin-set2.example.com`

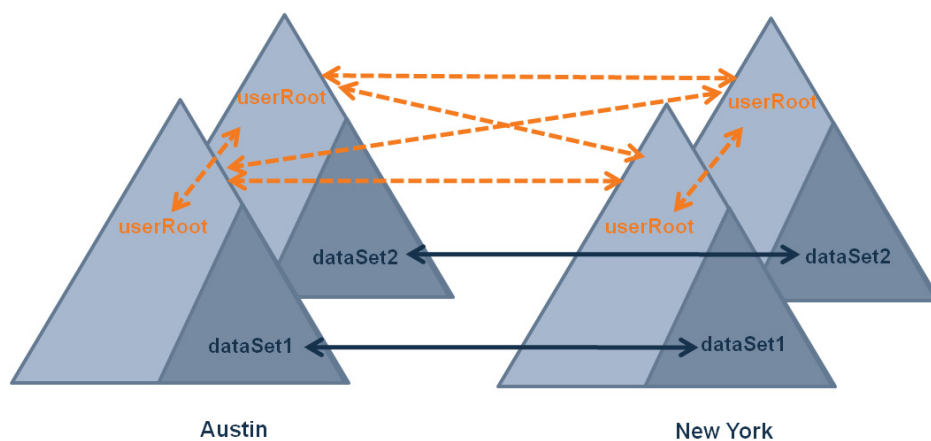
- `replica4 -- newyork-set2.example.com`

In this example, we have an entry-balanced domain of `dc=example,dc=com`. The data below the entry balancing point of `ou=people,dc=example,dc=com` is distributed across two data sets, `dataSet1` and `dataSet2`. Each data set is replicated between two directory servers. Each of these servers is associated with one of two locations, Austin and New York.

The following graphic illustrates this basic entry balanced deployment:



Replication takes place as described in the following graphic. Data in the `userRoot` backend is replicated between all of the servers in both locations. Data in each data set is replicated between a server in each location:



## Configuration Summary

To configure multi-mastered replication in an entry-balanced deployment, you must do the following:

- Install the directory server instances. In this deployment example, we will install two directory servers in the Austin location and two directory servers in the New York location.
- Create a new dataset backend and set the replication set name for each directory server instance.
- Create and set locations for the Austin and New York servers.
- Import or create data for the non-entry balanced global domain and for the entry balanced backends.
- Configure replication.
- Configure the proxy servers.
- Check the status of replication.

## Installing the Directory Servers

First, you install the directory server instances. In this example, we install the following four servers, two in the Austin location and two in the New York location:

- `austin-set1.example.com`
- `austin-set2.example.com`
- `newyork-set1.example.com`
- `newyork-set2.example.com`

We install the first server, `austin-set1`, as follows:

```
root@austin-set1# ./setup --cli --baseDN dc=example,dc=com \  
--ldapPort 389 --listenAddress austin-set1.example.com \  
--rootUserDN "cn=Directory Manager" --rootUserPassword pass \  
--aggressiveJVMTuning --maxHeapSize 1g --no-prompt \  
--noPropertiesFile --acceptLicense
```

We install the second Austin server, `austin-set2`, in the same way:

```
root@austin-set2# ./setup --cli --baseDN dc=example,dc=com \  
--ldapPort 389 --listenAddress austin-set2.example.com \  
--rootUserDN "cn=Directory Manager" --rootUserPassword pass \  
--aggressiveJVMTuning --maxHeapSize 1g --no-prompt \  
--noPropertiesFile --acceptLicense
```

Next, we the two New York servers, `newyork-set1` and `newyork-set2`, as follows:

```
root@newyork-set1# ./setup --cli --baseDN dc=example,dc=com \  
--ldapPort 389 --listenAddress newyork-set1.example.com \  
--rootUserDN "cn=Directory Manager" --rootUserPassword pass \  
--aggressiveJVMTuning --maxHeapSize 1g --no-prompt \  
--noPropertiesFile --acceptLicense  
  
root@newyork-set2# ./setup --cli --baseDN dc=example,dc=com \  
--ldapPort 389 --listenAddress newyork-set2.example.com \  

```

```
--rootUserDN "cn=Directory Manager" --rootUserPassword pass \  
--aggressiveJVMtuning --maxHeapSize 1g --no-prompt \  
--noPropertiesFile --acceptLicense
```

## Creating the Database Backends and Setting the Replication Set Name

We create a dataset database backends on each Austin directory server instance. First, we create `dataset` on `austin-set1.example.com`:

```
root@austin-set1# ./bin/dsconfig --no-prompt create-backend \  
--backend-name dataset --type local-db --set enabled:true \  
--set base-dn:ou=people,dc=example,dc=com \  
--set db-cache-percent:50 \  
--set "db-checkpointer-wakeup-interval:30 s" \  
--set db-num-cleaner-threads:1 --set prime-method:preload
```

Next, we create `dataset` on `austin-set2.example.com`:

```
root@austin-set2# ./bin/dsconfig --no-prompt create-backend \  
--backend-name dataset --type local-db --set enabled:true \  
--set base-dn:ou=people,dc=example,dc=com \  
--set db-cache-percent:50 \  
--set "db-checkpointer-wakeup-interval:30 s" \  
--set db-num-cleaner-threads:1 --set prime-method:preload
```

We set the replication set name for `austin-set1.example.com` to `dataset1`:

```
root@austin-set1# ./bin/dsconfig --no-prompt \  
set-global-configuration-prop \  
--set replication-set-name:dataset1
```

We set the replication set name for `austin-set2.example.com` to `dataset2`:

```
root@austin-set2# ./bin/dsconfig --no-prompt \  
set-global-configuration-prop \  
--set replication-set-name:dataset2
```

Now, we create a dataset database backends on each of the New York directory server instances in the same way. We create `dataset` on the `newyork-set1.example.com` and `newyork-set1.example.com` instances:

```
root@newyork-set1# ./bin/dsconfig --no-prompt create-backend \  
--backend-name dataset --type local-db --set enabled:true \  
--set base-dn:ou=people,dc=example,dc=com \  
--set db-cache-percent:50 \  
--set "db-checkpointer-wakeup-interval:30 s" \  
--set db-num-cleaner-threads:1 --set prime-method:preload  
  
root@newyork-set2# ./bin/dsconfig --no-prompt create-backend \  
--backend-name dataset --type local-db --set enabled:true \  
--set base-dn:ou=people,dc=example,dc=com \  
--set db-cache-percent:50 \  
--set "db-checkpointer-wakeup-interval:30 s" \  
--set db-num-cleaner-threads:1 --set prime-method:preload
```

We set the replication set name for `newyork-set1.example.com` to `dataset1`:

```
root@newyork-set1# ./bin/dsconfig --no-prompt \  
set-global-configuration-prop \  
--set replication-set-name:dataset1
```

We set the replication set name for `newyork-set2.example.com` to `dataset2`:

```
root@newyork-set2# ./bin/dsconfig --no-prompt \  
set-global-configuration-prop \  
--set replication-set-name:dataset2
```

## Creating and Setting Locations

On the `austin-set1` instance, we create the two locations, `newyork` and `austin`, and set the location of this instance to `austin`:

```
root@austin-set1# ./bin/dsconfig --no-prompt create-location \  
--location-name austin  
  
root@austin-set1# ./bin/dsconfig --no-prompt create-location \  
--location-name newyork \  
--set preferred-failover-location:austin  
  
root@austin-set1# ./bin/dsconfig --no-prompt \  
set-location-prop --location-name austin \  
--add preferred-failover-location:newyork  
  
root@austin-set1# ./bin/dsconfig --no-prompt \  
set-global-configuration-prop --set location:austin
```

We do the same on the `austin-set2` instance, creating the two locations, `newyork` and `austin`, and setting the location of the instance to `austin`:

```
root@austin-set2# ./bin/dsconfig --no-prompt create-location \  
--location-name austin  
  
root@austin-set2# ./bin/dsconfig --no-prompt create-location \  
--location-name newyork \  
--set preferred-failover-location:austin  
  
root@austin-set2# ./bin/dsconfig --no-prompt \  
set-location-prop --location-name austin \  
--add preferred-failover-location:newyork  
  
root@austin-set2# ./bin/dsconfig --no-prompt \  
set-global-configuration-prop --set location:austin
```

Next, we configure the two locations on the `newyork-set1` instance, and setting its location to `newyork`:

```
root@newyork-set1# ./bin/dsconfig --no-prompt create-location \  
--location-name austin  
  
root@newyork-set1# ./bin/dsconfig --no-prompt create-location \  
--location-name newyork \  
--set preferred-failover-location:austin
```

```
root@newyork-set1# ./bin/dsconfig --no-prompt \  
set-location-prop --location-name austin \  
--add preferred-failover-location:newyork  
  
root@newyork-set1# ./bin/dsconfig --no-prompt \  
set-global-configuration-prop --set location:newyork
```

Finally, we configure the two locations on the **newyork-set2** instance, also setting its location to **newyork**:

```
root@newyork-set2# ./bin/dsconfig --no-prompt create-location \  
--location-name austin  
  
root@newyork-set2# ./bin/dsconfig --no-prompt create-location \  
--location-name newyork \  
--set preferred-failover-location:austin  
  
root@newyork-set2# ./bin/dsconfig --no-prompt \  
set-location-prop --location-name austin \  
--add preferred-failover-location:newyork  
  
root@newyork-set2# ./bin/dsconfig --no-prompt \  
set-global-configuration-prop --set location:newyork
```

## Importing Entries

We import the **userRoot** data, based on data defined in the **userRoot.ldif** file, into each of the servers in our topology. This file does not contain entries at or within the entry-balancing point, **ou=people,dc=example,dc=com**. For example, we import the data into **austin-set1** as follows:

```
root@austin-set1# ./bin/import-ldif --backendID userRoot \  
--ldifFile `pwd`/userRoot.ldif \  
--includeBranch dc=example,dc=com \  
--rejectFile `pwd`/austin-set1-import-rejects \  
--port 389 --hostname austin-set1.example.com
```

We import the **dataset1** data into the servers in that replication set, **austin-set1** and **newyork-set1**, as follows:

```
root@austin-set1# ./bin/import-ldif --backendID dataset \  
--ldifFile `pwd`/dataset1.ldif \  
--includeBranch ou=people,dc=example,dc=com \  
--rejectFile `pwd`/austin-set1-dataset-import-rejects \  
--hostname austin-set1.example.com --port 389  
  
root@austin-set1# ./bin/import-ldif --backendID dataset \  
--ldifFile `pwd`/dataset1.ldif \  
--includeBranch ou=people,dc=example,dc=com \  
--rejectFile `pwd`/newyork-set1-dataset-import-rejects \  
--hostname newyork-set1.example.com --port 389
```

Finally, we import the **dataset2** data into the servers in the **set2** replication set, **austin-set2** and **newyork-set2**, as follows:

```
root@austin-set2# ./bin/import-ldif --backendID dataset \
--ldifFile `pwd`/dataset2.ldif \
--includeBranch ou=people,dc=example,dc=com \
--rejectFile `pwd`/austin-set1-dataset-import-rejects \
--hostname austin-set2.example.com --port 389

root@austin-set2# ./bin/import-ldif --backendID dataset \
--ldifFile `pwd`/dataset2.ldif \
--includeBranch ou=people,dc=example,dc=com \
--rejectFile `pwd`/newyork-set1-dataset-import-rejects \
--hostname newyork-set2.example.com --port 389
```

## Enabling Replication

Now we can enable replication between the servers in our topology. First, we configure replication between **austin-set1** and **austin-set2**. Notice that we specify the **--restricted** domain in the **dsreplication** command even though these two servers are not sharing the same dataset:

```
root@austin-set1# ./bin/dsreplication enable \
--host1 austin-set1.example.com \
--port1 389 --bindDN1 "cn=directory manager" \
--bindPassword1 pass --host2 austin-set2.example.com \
--port2 389 --bindDN2 "cn=directory manager" \
--bindPassword2 pass \
--replicationPort1 8989 \
--replicationPort2 8989 \
--baseDN dc=example,dc=com \
--baseDN ou=people,dc=example,dc=com \
--restricted ou=people,dc=example,dc=com \
--adminUID admin --adminPassword pass --trustAll \
--no-prompt --ignoreWarnings
```

Next, we enable replication between **austin-set1** and **newyork-set1**. This procedure automatically enables replication between **austin-set2** and **newyork-set1** as well.

```
root@austin-set1# ./bin/dsreplication enable \
--host1 austin-set1.example.com \
--port1 389 --bindDN1 "cn=directory manager" \
--bindPassword1 pass --host2 newyork-set1.example.com \
--port2 389 --bindDN2 "cn=directory manager" \
--bindPassword2 pass \
--replicationPort1 8989 \
--replicationPort2 8989 \
--baseDN dc=example,dc=com \
--baseDN ou=people,dc=example,dc=com \
--restricted ou=people,dc=example,dc=com \
--adminUID admin --adminPassword pass --trustAll \
--no-prompt --ignoreWarnings
```

Finally, we configure replication between **austin-set1** and **newyork-set2**. Because of the replication agreements created in the previous step, this procedure automatically enables replication between **austin-set2** and **newyork-set2** as well.



```
root@austin-set1# ./bin/dsreplication enable \  
--host1 austin-set1.example.com \  
--port1 389 --bindDN1 "cn=directory manager" \  
--bindPassword1 pass --host2 newyork-set2.example.com \  
--port2 389 --bindDN2 "cn=directory manager" \  
--bindPassword2 pass \  
--replicationPort1 8989 \  
--replicationPort2 8989 \  
--baseDN dc=example,dc=com \  
--baseDN ou=people,dc=example,dc=com \  
--restricted ou=people,dc=example,dc=com \  
--adminUID admin --adminPassword pass --trustAll \  
--no-prompt --ignoreWarnings
```

Note: initialization is not required because we have imported all of the same data into the replicas.

## Checking the Status of Replication

Once replication has been configured, we check the status of replication as follows:

```
root@austin-set1# ./bin/dsreplication status \  
--adminPassword pass --no-prompt --port 389
```

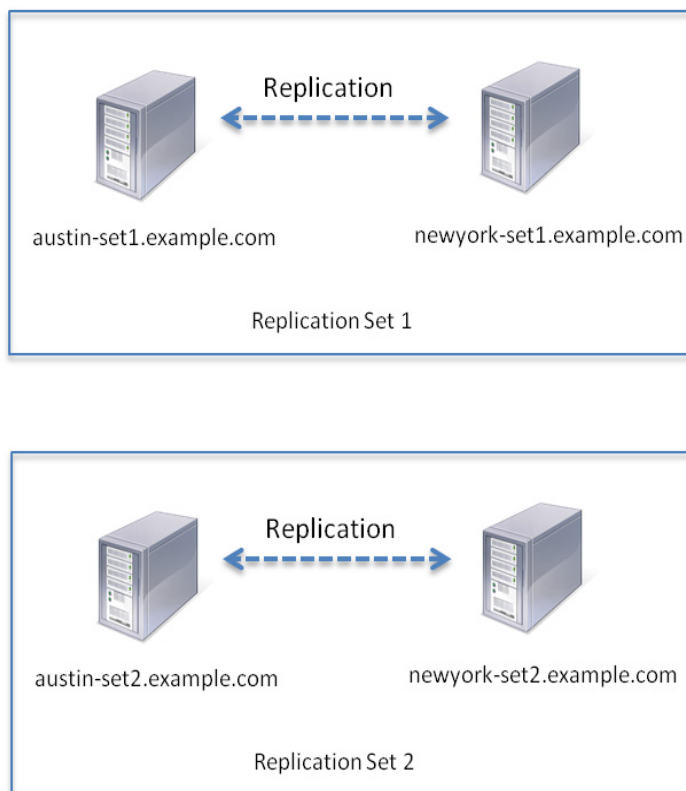
# Detaching a Replication Set from a Topology

Sometimes, you may need to detach a single server or a set of servers from the replication topology without disabling replication on the servers. To do so, you can use the **dsreplication detach** command.

For example, to detach the directory instances in the replication set **set1** from the replication topology, use the following command:

```
root@austin-set1# ./bin/dsreplication detach \  
--hostname server1.example.com --port 1389 --adminUID admin \  
--adminPassword password --setname set1
```

Replication no longer occurs between replication sets 1 and 2, though replication continues to take place within each of the two sets, as illustrated in the following figure.



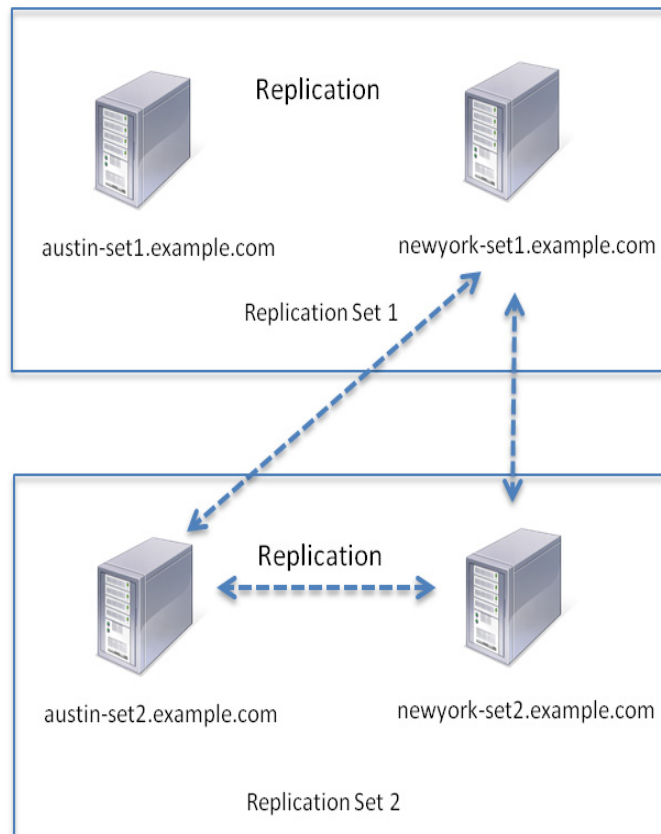
You can also detach a single server from the replication topology. For example, to detach only server 1 from the replication topology but leave replication set 1 connected, use the following command:

```
$ bin/dsreplication detach --hostname austin-set1.example.com \
--port 1389 --adminUID admin --adminPassword password
```

Note that **austin-set1** still has replication enabled even though it no longer participates in the replication topology of **newyork-set1**, **austin-set2**, and **newyork-set2**. Updates made on **austin-set1** will be recorded in the changelog database of the embedded replication server instance. When **austin-set1** is added back to the topology, these updates will be propagated to the other servers. Changes made on the other servers while **austin-set1** was detached will also be applied.

Use the **detach** subcommand only if **austin-set1** will be added back to the replication topology. Otherwise, consider using the **disable** subcommand instead.

The server continues to be part of replication set 1, though it no longer participates in replication, as illustrated in the following figure:



You might detach a single server for a short time to diagnose a problem without taking the server out of the topology. Once re-enabled, it receives updates from the other replication servers if the purge delay has not been exceeded. Otherwise, you need to reinitialize the re-enabled replica.



---

# 8

## Deploying the Directory Proxy Server

You can deploy UnboundID® Directory Proxy Server in a variety of ways, depending upon the needs of your enterprise. This chapter describes and illustrates several common deployment scenarios.

It includes the following sections:

- Creating a Standard Multi-Location Deployment
- Expanding the Deployment
- Merging Two Data Sets Using Proxy Transformations
- Deploying an Entry-Balancing Proxy Configuration

### Creating a Standard Multi-Location Deployment

In this example deployment, UnboundID Directory Proxy Server will be deployed in the data centers of two geographic locations: East and West. All LDAP external servers in this deployment are UnboundID® Directory Servers. The directory servers in the eastern city are assigned to the location named East, and the directory servers in the western city are assigned to the location named West.

This example will configure round-robin load balancing between four directory servers in the two locations as well as replication of the `dc=example,dc=com` base DN:

- `ds-east-01.example.com`
- `ds-east-02.example.com`
- `ds-west-01.example.com`
- `ds-west-02.example.com`

We will also configure four proxy servers:

- `proxy-east-01.example.com`
- `proxy-east-02.example.com`
- `proxy-west-01.example.com`
- `proxy-west-02.example.com`

## Overview of Deployment Steps

In this deployment scenario, we will take the following steps:

- Install the first Directory Proxy Server in East location using the **setup** or **setup.bat** file included in the zip installation file.
- Use the **create-initial-proxy-config** tool to provide a proxy user bind DN and password, define locations for each of our data centers, and configure the LDAP external servers in these data centers.
- Test external server communications after initial setup is complete and test a simulated external server failure.
- Install the second directory proxy server in the East location using the **setup** or **setup.bat** file included in the zip installation file and copy the configuration of the first proxy using the configuration cloning feature.
- Install two proxy servers in the West location, which includes using the **setup** file and manually setting the location to **west** using the **dsconfig** command, as well as copying the configuration of the proxy server using the configuration cloning feature.

After the directory proxy server has been configured and tested, we then provide a tour of the configuration of each of the directory proxy server components. These properties can be modified later as needed using the **dsconfig** tool.

## Installing the First Directory Proxy Server

To begin with, we have the UnboundID Directory Proxy Server installation zip file. In this example, we plan to use SSL security, so we also have a keystore certificate database and a pin file that contains the private key password for the keystore. The keystore files are only necessary when using SSL or StartTLS.

In this deployment scenario, the keystore database is assumed to be a Java Key Store (JKS), which can be created by the **keytool** programs. For more information about keytool, see <http://download.oracle.com/javase/6/docs/technotes/tools/windows/keytool.html>.

```
root@proxy-east-01: ls
ExampleKeystore           ExampleTruststore.jks
ExampleKeystore.pin       ExampleTruststore.pin
UnboundID-Proxy-3.1.0-with-je.zip
```

We unzip the compressed archive file into the UnboundID-Proxy directory and move to this directory.

```
root@proxy-east-01: unzip -q \
UnboundID-Proxy-3.1.0.0-with-je.zip
root@proxy-east-01: cd UnboundID-Proxy
root@proxy-east-01: ls
```

|            |                |                    |             |
|------------|----------------|--------------------|-------------|
| bak        | ldif           | revert-upgrade     | upgrade     |
| bat        | legal-notice   | setup              | upgrade.bat |
| bin        | lib            | setup.bat          |             |
| classes    | LICENSE.txt    | UnboundID_logo.png |             |
| config     | locks          | uninstall          |             |
| docs       | logs           | Uninstall.app      |             |
| import-tmp | QuickSetup.app | uninstall.bat      |             |

Because we are configuring SSL security, we copy the keystore and pin files into the config directory.

```
root@proxy-east01: cp ../Keystore* config/
root@proxy-east01: cp ../Truststore* config/
```

Next, we install the first directory proxy server by running the `setup` tool on `proxy-east-01.example.com` as follows:

```
root@proxy-east01: ./setup --no-prompt --acceptLicense \
--ldapPort 389 --rootUserPassword pass --entryBalancing \
--aggressiveJVMTuning --maxHeapSize 1g \
--enableStartTLS --ldapsPort 636 \
--useJavaKeystore config/ExampleKeystore.jks \
--keyStorePasswordFile config/ExampleKeystore.pin \
--certNickname server-cert \
--useJavaTrustStore config/ExampleTruststore.jks \
--trustStorePasswordFile config/ExampleTruststore.pin
```

New keystore password files are created stored in `config/keystore.pin` and `config/truststore.pin`. The original files, `config/ExampleKeystore.pin` and `config/ExampleTruststore.pin`, are no longer needed.

If you are not using SSL or StartTLS, then the following arguments are not necessary:

```
--enableStartTLS --ldapsPort --useJavaKeystore \
--keyStorePasswordFile --certNickname --useJavaTrustStore \
--trustStorePasswordFile
```

## Configuring the First Directory Proxy Server Using the `create-initial-proxy-config` Tool

Once the directory proxy server has been installed, it can be automatically configured using the `create-initial-proxy-config` tool. This tool can only be used once for this initial configuration, after which we will have to use `dsconfig` to make any changes to our directory proxy server configuration.

Configuring the Directory Proxy Server with the `create-initial-proxy-config` tool involves the following steps:

- Providing a directory proxy server base DN and password.
- Defining locations for each of our data centers, East and West.
- Configuring the LDAP external server in the East location.
- Configuring the LDAP external servers in the West location.

- Applying the changes to the directory proxy server.

Once we have completed setup, we run the `create-initial-proxy-config` tool as follows:

```
root@proxy-east01: bin/create-initial-proxy-config

>>>> >>>> Initial Proxy Configuration Tool

This tool can be used to generate a basic Directory Proxy Server
configuration by prompting for basic information about your
topology including directory server instances, their locations,
and credentials for communicating with them. This tool will
record the configuration in a 'dsconfig' batch file and option-
ally allow you to apply the configuration to the local Direc-
tory Proxy Server.

Some assumptions are made about the topology in order to keep
this tool simple:

1) all servers will be accessible via a single user account
2) all servers support the same communication security type
3) all servers are UnboundID, Alcatel-Lucent 8661, or Sun Java
System 5.x, 6.x, or 7.x, or Red Hat (including Fedora and 389)
directory servers

If your topology does not have these characteristics you can
use this tool to define a basic configuration and then use the
'dsconfig' tool or the web console to fine tune the configura-
tion.

Would you like to continue? (yes / no) [yes]:
```

First, we provide the bind DN and password that the directory proxy server will use to authenticate to the backend directory servers. The `create-initial-proxy-config` tool requires that the same bind DN and password be used to authenticate to all of the backend servers. All of our directory proxy servers have identical proxy user accounts and passwords.

```
>>>> >>>> External Server Access Credentials

Provide the DN of the root account that will be used when com-
municating with Directory Server instances. In a later step you
will be given the opportunity to have this tool create this
account or set its password for each Directory Server instance
you specify.

Enter the DN of the proxy user account [cn=Proxy
User,cn=Root,cn=config]:

Enter the password for 'cn=Proxy User,cn=Root,cn=config':

Confirm the password for 'cn=Proxy User,cn=Root,cn=config':
```

We specify 1 to use no security for communications with directory server instances.

```
>>>> >>>> External Server Communication Security

Specify the type of security that the Directory Proxy Server
will use when communicating with Directory Server instances:
```



- 1) None
- 2) SSL
- 3) StartTLS
- b) back
- q) quit

Enter choice [1]:

We use the `dc=example,dc=com` base DN for our directory proxy server. We are not using entry balancing for this configuration example, so we enter no in response to that question.

>>>> >>>> Proxy Base DNs

Enter the base DNs of the Directory Server instances that will be accessed through the Directory Proxy Server

- b) back
- q) quit

Enter a DN or choose a menu item [dc=example,dc=com]:

Are entries within 'dc=example,dc=com' split across multiple servers so that each server stores only a subset of the entries (i.e. is this base DN 'entry balanced')? (yes / no) [no]:

We press enter because we are only using one proxy base DN.

>>>> >>>> Proxy Base DNs

Enter the base DNs of the Directory Server instances that will be accessed through the Directory Proxy Server

- 1) Remove dc=example,dc=com
- b) back
- q) quit

Enter a DN or choose a menu item [Press ENTER when finished entering base DNs]:

## Defining Locations

Now we define our first location, East, to accommodate the servers in our deployment located on the East Coast of the United States.

>>>> >>>> Define Locations

Locations are used to define collections of servers which may have similar performance characteristics when accessed from this Directory Proxy Server. For example, a separate Location may be defined for each data center.

A good rule of thumb when naming locations is to use the name of your data centers or the cities containing them.

- b) back
- q) quit

```
Enter a location name or choose a menu item: East
```

- 1) Remove East
- b) back
- q) quit

We define a location named West for the servers in our deployment located on the West Coast. Then we press enter to indicate that we have finished specifying locations.

```
Enter another location name or choose a menu item [Press ENTER
when finished entering locations]: West
```

- 1) Remove East
- 2) Remove West
- b) back
- q) quit

```
Enter another location name or choose a menu item [Press ENTER
when finished entering locations]:
```

Now we select the location that contains the directory proxy server itself. Our directory proxy server is located in the East, so we select 1.

```
>>>> >>>> Proxy Location
```

```
Choose the location for this Directory Proxy Server
```

- 1) East
- 2) West
- b) back
- q) quit

```
Enter choice [1]: 1
```

## Configuring the External Servers in the East Location

Once the locations have been defined, we need to identify the directory servers. First, we define one of the servers in the East location.

```
>>>> External Servers
```

```
External Servers identify directory server instances including
host, port, and authentication information.
```

```
Enter the host and port (host:port) of the first directory
server in 'East'
```

- b) back
- q) quit

```
Enter a host:port or choose a menu item [localhost:389]:
ds-east-01.example.com:389
```

Next, we select 3 to indicate that we want the tool to create a proxy user account on all of our external servers.

```
Would you like to prepare ds-east-01.example.com:389 for access
by the Proxy Server?
```

- 1) Yes
- 2) No
- 3) Yes, and all subsequent servers
- 4) No, and all subsequent servers

```
Enter choice [1]: 3
```

```
Testing connection to ds-east-01.example.com:389 ..... Done
Testing 'cn=Proxy User,cn=Root DNs,cn=config' access to
ds-east-01.example.com:389 ..... Done
```

We need to connect as `cn=Directory Manager`.

```
Enter the DN of an account on ds-east-01.example.com:389 with
which to create or manage the 'cn=Proxy User,cn=Root
DNs,cn=config' account and configuration[cn=Directory Manager]:
```

```
Enter the password for 'cn=Directory Manager':
```

```
Testing 'cn=Proxy User,cn=Root DNs,cn=config' privileges...Done
Verifying backend 'dc=example,dc=com' .....Done
```

We repeat the process for the other server in the East location, `ds-east-02.example.com`.

```
Enter another server in east or choose a menu item
```

- 1) Remove ds-east-01.example.com:389
- b) back
- q) quit

```
Enter a host:port or choose a menu item [Press ENTER when fin-
ished entering servers for 'east']: ds-east-02.example.com:389
```

```
Testing connection to ds-east-02.example.com:389 ..... Done
Testing 'cn=Proxy User,cn=Root DNs,cn=config' access.....Done
```

```
Would you like to use the previously entered manager
credentials to access all prepared servers? (yes / no) [yes]:
```

```
Testing 'cn=Proxy User,cn=Root DNs,cn=config' privileges...Done
Verifying backend 'dc=example,dc=com' ..... Done
```

We press enter to complete configuring the East location.

```
Enter another server in east or choose a menu item
```

- 1) Remove ds-east-01.example.com:389
- 2) Remove ds-east-02.example.com:389
- b) back
- q) quit

```
Enter a host:port or choose a menu item [Press ENTER when fin-
ished entering servers for 'east']:
```

## Configuring the External Servers in the West Location

We repeat the process used for the East location to define the LDAP external servers used for the West location. We define the first external server, `ds-west-01.example.com`.

```
>>>> >>>> Location 'West' Details

>>>> External Servers

External Servers identify directory server instances including
host, port, and authentication information.

Enter the host and port (host:port) of the first directory
server in 'West'

    b) back
    q) quit

Enter a host:port or choose a menu item [localhost:389]: ds-
west-01.example.com:389

Testing connection to ds-west-01.example.com:389 ..... Done
Testing 'cn=Proxy User,cn=Root DNs,cn=config' access ...Done
Testing 'cn=Proxy User,cn=Root DNs,cn=config' privileges ..Done
Verifying backend 'dc=example,dc=com' .....Done
```

We define the second server in the west location, `ds-west-02.example.com`.

```
Enter another server in 'West'

    1) Remove ds-west-01.example.com:389
    b) back
    q) quit

Enter a host:port or choose a menu item [Press ENTER when fin-
ished entering servers]: ds-west-02.example.com:389

Testing connection to ds-west-02.example.com:389 ..... Done
Testing 'cn=Proxy User,cn=Root DNs,cn=config' access...Denied

Would you like to create or modify root user 'cn=Proxy User' so
that it is available for this Proxy Server? (yes / no) [yes]:

Created 'cn=Proxy User,cn=Root DNs,cn=config'
Testing 'cn=Proxy User,cn=Root DNs,cn=config' privileges...Done
Verifying backend 'dc=example,dc=com' ..... Done
```

We press enter to indicate that we have finished defining external servers for the west location.

```
Enter another server in 'West'

    1) Remove ds-west-01.example.com:389
    2) Remove ds-west-02.example.com:389
    b) back
    q) quit

Enter a host:port or choose a menu item [Press ENTER when fin-
ished entering servers]:
```

## Applying the Changes to the Directory Proxy Server

We review the configuration summary. Once we have confirmed that the changes are correct, we press enter to write the configuration.

```
>>>> >>>> Configuration Summary

External Server Security:  None
Proxy User DN:            cn=Proxy User,cn=Root DNs,cn=config

Location East
  Failover Order: West
  Servers: ds-east-01.example.com:389,
          ds-east-02.example.com:389

Location West
  Failover Order: East
  Servers: ds-west-01.example.com:389,
          ds-west-02.example.com:389

Base DN: dc=example,dc=com
  Servers: ds-east-01.example.com:389,
          ds-east-02.example.com:389,ds-west-01.example.com:389,
          ds-west-02.example.com:389

b)  back
q)  quit
w)  write configuration

Enter choice [w]:
```

We press enter to apply the changes to the directory proxy server. Once the changes have been applied, we cannot use the `create-initial-proxy-config` tool to configure this directory proxy server again.

```
>>>>>>> Write Configuration

The configuration will be written to a 'dsconfig' batch file
that can be used to configure other Directory Proxy Servers

Writing Directory Proxy Server configuration to /UnboundID/
proxy/UnboundID-Proxy/dps-cfg.txt .....Done

Apply these configuration changes to the local Directory Proxy
Server? (yes /no) [yes]:

How do you want to connect to the Directory Proxy Server on
localhost?

1)  LDAP
2)  LDAP with SSL
3)  LDAP with StartTLS

Enter choice [1]:

Administrator user bind DN [cn=Directory Manager]:

Password for user 'cn=Directory Manager':
```

```
Creating Locations ..... Done
Updating Failover Locations ..... Done
Updating Global Configuration ..... Done
Creating Health Checks ..... Done
Creating External Servers ..... Done
Creating Load-Balancing Algorithm for dc=example,dc=com....Done
Creating Request Processors for dc=example,dc=com ..... Done
Creating Subtree Views for dc=example,dc=com ..... Done
Updating Client Connection Policy for dc=example,dc=com....Done

See /UnboundID/proxy/UnboundID-Proxy/logs/create-initial-proxy-
config.log for a detailed log of this operation

To see basic server configuration status and configuration you
can launch /UnboundID/proxy/UnboundID-Proxy/bin/status
```

## Configuring the Additional Proxy Servers

We install and configure the second directory proxy server by running the **setup** tool on **proxy-east-02.example.com**. First, we copy the keystore and pin files into the config directory:

```
root@proxy-east-02: cp ../Keystore* config/
root@proxy-east-02: cp ../Truststore* config/
```

Next, we install the second directory proxy server by running the **setup** tool on **proxy-east-02.example.com** as follows:

```
root@proxy-east-02: ./setup --no-prompt --listenAddress \
proxy-east-02.example.com --ldapPort 389 \
--entryBalancing --enableStartTLS --ldapsPort 636 \
--useJavaKeystore config/ExampleKeystore.jks \
--keyStorePasswordFile config/ExampleKeystore.pin \
--certNickName server-cert \
--useJavaTrustStore config/ExampleTruststore.jks \
--trustStorePasswordFile config/ExampleTruststore.pin \
--rootUserPassword pass --acceptLicense \
--aggressiveJVMTuning --maxHeapSize 1g \
--localHostName proxy-east-02.example.com \
--peerHostName proxy-east-01.example.com \
--peerPort 389 --location East
```

```
UnboundID Directory Proxy Server 3.1.0.0
Please wait while the setup program initializes...
Reading Peer Configuration ..... Done
Applying Peer Configuration ..... Done
Updating Administration Data ..... Done
Configuring Directory Proxy Server ..... Done
Configuring Certificates ..... Done
Updating Topology Admin Data ..... Done
Starting Directory Proxy Server ..... Done
```

To see basic server configuration status and configuration you can launch **/bin/status**

See `/logs/tools/setup.log` for a detailed log of this operation.

We configure the third directory proxy server, `proxy-west-01.example.com`, as before. First, we copy the keystore and pin files into the config directory:

```
root@proxy-west-01: cp ../Keystore* config/
root@proxy-west-01: cp ../Truststore* config/
```

Then, we run the `setup` tool on `proxy-west-01.example.com` as follows:

```
root@proxy-west-01: ./setup --no-prompt \
--listenAddress proxy-west-01.example.com \
--ldapPort 389 --entryBalancing --enableStartTLS \
--ldapsPort 636 \
--useJavaKeystore config/ExampleKeystore.jks \
--keyStorePasswordFile config/ExampleKeystore.pin \
--certNickName server-cert \
--useJavaTrustStore config/ExampleTruststore.jks \
--trustStorePasswordFile config/ExampleTruststore.pin \
--rootUserPassword pass --acceptLicense \
--aggressiveJVMTuning --maxHeapSize 1g \
--localHostName proxy-west-01.example.com \
--peerHostName proxy-east-01.example.com \
--peerPort 389 --location West
```

```
UnboundID Directory Proxy Server 3.1.0.0
Please wait while the setup program initializes...
Reading Peer Configuration ..... Done
Applying Peer Configuration ..... Done
Updating Administration Data ..... Done
Configuring Directory Proxy Server ..... Done
Configuring Certificates ..... Done
Updating Topology Admin Data ..... Done
Starting Directory Proxy Server ..... Done
```

To see basic server configuration status and configuration you can launch `/bin/status`

Finally, we install the last directory proxy server, by first copying the keystore and pin files and then running the `setup` tool:

```
root@proxy-west-02: cp ../Keystore* config/
root@proxy-west-02: cp ../Truststore* config/

root@proxy-west-02: ./setup --no-prompt \
--listenAddress proxy-west-02.example.com --ldapPort 389 \
--entryBalancing --enableStartTLS --ldapsPort 636 \
--useJavaKeystore config/ExampleKeystore.jks \
--keyStorePasswordFile config/ExampleKeystore.pin \
--certNickName server-cert \
--useJavaTrustStore config/ExampleTruststore.jks \
--trustStorePasswordFile config/ExampleTruststore.pin \
--rootUserPassword pass --acceptLicense \
--aggressiveJVMTuning --maxHeapSize 1g \
--localHostName proxy-west-02.example.com \
```

```
--peerHostName proxy-east-01.example.com \  
--peerPort 389 --location West  
  
UnboundID Directory Proxy Server 3.1.0.0  
Please wait while the setup program initializes...  
Reading Peer Configuration ..... Done  
Applying Peer Configuration ..... Done  
Updating Administration Data ..... Done  
Configuring Directory Proxy Server ..... Done  
Configuring Certificates ..... Done  
Updating Topology Admin Data ..... Done  
Starting Directory Proxy Server ..... Done  
  
To see basic server configuration status and configuration you  
can launch /bin/status  
  
See /logs/tools/setup.log for a detailed log of this operation.
```

At this point, all proxy servers have the same Admin Data backend and have the **all-servers** group defined as their **configuration-server-group** in the Proxy Server Global Configuration object. When making a change to a proxy server using the **dsconfig** command-line tool or the Web console, you will have the choice to apply the changes locally only or to all proxy servers in the **all-servers** group.

## Testing External Server Communications After Initial Setup

After setting up the basic deployment scenario, the communication between the directory proxy server and the LDAP external servers can be tested using a feature in the directory proxy server in combination with an LDAP search.

This feature exposes a special search base DN for testing external server connectivity, called the backend server pass-through subtree view. While disabled by default, you can enable this feature using **dsconfig** in the "Client Connection Policy" menu. Set the value of the **include-backend-server-passthrough-subtree-views** property to true. You can also call **dsconfig** in non-interactive mode with the following options:

```
root@proxy-east-01: dsconfig set-client-connection-policy-prop \  
--policy-name default \  
--set include-backend-server-passthrough-subtree-views:true
```

Once set to true, an LDAP search against the proxy with the base DN **"dc=example,dc=com,ds-backend-server=ds-east-02.example.com:389"** instructs the proxy to perform the search against the **ds-east-02.example.com:389** external server with the base DN set to **"dc=example,dc=com."**

The following example fetches the **dc=example,dc=com** entry from **ds-east-01.example.com** server:

```
root@proxy-east-01: bin/ldapsearch \  
--bindDN "cn=Directory Manager" \  
--bindPassword password \  
--baseDN "dc=example,dc=com,ds-backend-server=ds-east-01.example.com:389" \  
--searchScope base --useStartTLS "(objectclass=*)" "
```



Perform this search on each external server to determine if external server communication has been configured correctly on the directory proxy server.

You can also use this special subtree view to track the operations performed on each external server to help determine load balancing requirements. The following example searches an external server's monitor entry to display operation statistics:

```
root@proxy-east-01: bin/ldapsearch \  
--bindDN "cn=directory manager" \  
--bindPassword password \  
--baseDN "cn=monitor,ds-backend-server=ds-east-02.example.com:389" \  
--searchScope sub --useStartTLS "(cn=ldap*statistics)" \  
  
dn: cn=LDAP Connection Handler 192.168.1.203 port 389 \  
Statistics,cn=monitor,ds-backend-server=ds-east-02.example.com:389 \  
  
objectClass: top \  
objectClass: ds-monitor-entry \  
objectClass: ds-ldap-statistics-monitor-entry \  
objectClass: extensibleObject \  
cn: LDAP Connection Handler 192.168.1.203 port 389 \  
Statistics \  
connectionsEstablished: 3004 \  
connectionsClosed: 2990 \  
bytesRead: 658483 \  
bytesWritten: 2061549 \  
ldapMessagesRead: 17278 \  
ldapMessagesWritten: 22611 \  
operationsAbandoned: 0 \  
operationsInitiated: 17278 \  
operationsCompleted: 14241 \  
abandonRequests: 22 \  
addRequests: 1 \  
addResponses: 1 \  
bindRequests: 3006 \  
bindResponses: 3006 \  
compareRequests: 0 \  
compareResponses: 0 \  
deleteRequests: 0 \  
deleteResponses: 0 \  
extendedRequests: 2987 \  
extendedResponses: 2987 \  
modifyRequests: 1 \  
modifyResponses: 1 \  
modifyDNRequests: 0 \  
modifyDNResponses: 0 \  
searchRequests: 8271 \  
searchResultEntries: 8370 \  
searchResultReferences: 0 \  
searchResultsDone: 8246 \  
unbindRequests: 2990
```

This LDAP search can be run with the base DN values for the **ds-east-01** and **ds-east-02** servers to track the distribution of search and bind requests over time. These statistics are reset to zero when the server restarts.

## Testing a Simulated External Server Failure

Once you have tested connectivity, run a simulated failure of a load-balanced external server to verify that the directory proxy server redirects LDAP requests appropriately.

We stop the `ds-east-01.example.com:389` server instance and test searches through `proxy-east-01.example.com`.

First, perform several searches against the proxy. Verify activity in each of the servers in the east location, `ds-east-01` and `ds-east-02`, by looking at the access logs. The following simple search can be repeated as needed:

```
root@proxy-east-01: bin/ldapsearch \  
--bindDN "cn=Directory Manager" \  
--bindPassword password --baseDN "dc=example,dc=com" \  
--searchScope base --useStartTLS "(objectclass=*)" "
```

Next, stop the directory server instance on `ds-east-01.example.com` using the `stop-ds` command and immediately retry the above searches. There should be no errors or noticeable delay in processing the search.

```
root@proxy-east-01: bin/stop-ds  
  
root@proxy-east-01: bin/ldapsearch \  
--bindDN "cn=Directory Manager" \  
--bindPassword password --baseDN "dc=example,dc=com" \  
--searchScope base --useStartTLS "(objectclass=*)" "
```

Restart the directory server instance on `ds-east-01.example.com`. Check the access log to confirm that the directory proxy server started to include the `ds-east-01` server in round robin load-balancing within 30 seconds. The default time is 30 seconds, though you can change this default if desired.

## Expanding the Deployment

In this example deployment, UnboundID Directory Server is deployed in a third, centrally-located data center. The directory servers in the central city is assigned to a new location named Central.

The directory proxy server itself, installed on `proxy-east-01.example.com`, remains in the East location. This example will reconfigure round-robin load balancing between the six directory servers in three locations:

```
ds-east-01.example.com  
ds-east-02.example.com  
ds-west-01.example.com  
ds-west-02.example.com  
ds-central-01.example.com  
ds-central-02.example.com
```

## Overview of Deployment Steps

In this deployment scenario, we will take the following steps:

- Prepare the new external servers using the `prepare-external-servers` tool.
- Use the `dsconfig` tool to configure the new LDAP external servers in the central data center and reconfigure the load-balancing algorithm to take these servers into account.
- Test external server communications after the servers have been configured and test a simulated external server failure.

### Preparing Two New External Servers Using the `prepare-external-server` Tool

First, we prepare the directory proxy server to add two new servers to the deployment, `ds-central-01` and `ds-central-02`. In this example, we will connect to the `ds-central-01` directory server using StartTLS. Because we are using StartTLS, we need to capture the `ds-central-01` server's certificate and put it in the trust store on our directory proxy server instance.

The `prepare-external-server` tool is located in the `bin` or `bat` directory of the server root directory, `UnboundID-Proxy`. In this example, we run the tool on the `ds-east-01` instance of the directory proxy server.

```
root@proxy-east-01: ./prepare-external-server \
--hostname ds-central-01.example.com --port 389 \
--baseDN dc=example,dc=com \
--proxyBindPassword password --useStartTLS \
--proxyTrustStorePath ../config/ExampleTruststore.jks
--proxyTrustStorePasswordFile ../config/truststore.pin
```

```
Testing connection to ds-central-01:389....Done
Testing 'cn=Proxy User' access to ds-central-01:389....
```

```
Do you wish to trust the following certificate?
Certificate Subject: CN=example.home, O=UnboundID Self-Signed Certificate
Issuer Subject:      CN=example.home, O=UnboundID Self-Signed Certificate
Validity:            Tue May 12 21:59:42 EDT 2010 to Mon May 07 21:59:42 EDT 2029
```

```
Enter 'y' to trust the certificate or 'n' to reject it.
```

```
y
```

```
Failed to bind as 'cn=Proxy User'
```

```
Would you like to create or jmodify root user 'cn=Proxy User' so that it is
available for this Proxy Server? (yes / no) [yes]:
```

```
Enter the DN of an account on ds-central-01:389 with which to create or man-
age the 'cn=Proxy User' account [cn=Directory Manager]:
```

```
Enter the password for 'cn=Directory Manager':
```

```
Created 'cn=Proxy User,cn=Root DNs,cn=config'
Testing 'cn=Proxy User' privileges ....Done
```

We repeat the process on the other new server in the central location, **ds-central-02**.

```
root@proxy-east-01: ./prepare-external-server \
--hostname ds-central-02.example.com --port 389 \
--baseDN dc=example,dc=com \
--proxyBindPassword password --useStartTLS \
--proxyTrustStorePath ../config/ExampleTruststore.jks \
--proxyTrustStorePasswordFile ../config/truststore.pin \
```

```
Testing connection to ds-central-02:389....Done
Testing 'cn=Proxy User' access to ds-central-02:389....
```

```
Do you wish to trust the following certificate?
Certificate Subject: cn=example.home, o=UnboundID Self-Signed Certificate
Issuer Subject:      cn=example.home, o=UnboundID Self-Signed Certificate
Validity:            Tue May 12 21:59:42 EDT 2010 to Mon May 07 21:59:42 EDT 2029
```

Enter 'y' to trust the certificate or 'n' to reject it.

y

Failed to bind as 'cn=Proxy User'

Would you like to create or jmodify root user 'cn=Proxy User' so that it is available for this Proxy Server? (yes / no) [yes]:

Enter the DN of an account on ds-central-02:389 with which to create or manage the 'cn=Proxy User' account [cn=Directory Manager]:

Enter the password for 'cn=Directory Manager':

```
Created 'cn=Proxy User,cn=Root DNs,cn=config'
Testing 'cn=Proxy User' privileges ....Done
```

## Adding the New Directory Servers to the Proxy

After preparing the external directory servers to communicate with the directory proxy server, we can now add the two servers in the central location to the directory proxy server instance. Because we have run the **prepare-external-servers** tool, the two servers have the **cn=Proxy User** entry configured.

We run the **dsconfig** tool, which is located in the bin or bat directory of the server root directory, UnboundID-Proxy.

```
root@proxy-east-01: ./dsconfig

>>>> Specify LDAP connection parameters

Directory Proxy Server hostname or IP address [localhost]:

How do you want to connect to the Directory Proxy Server at
localhost?
```

```
1)  LDAP
2)  LDAP with SSL
3)  LDAP with StartTLS

Enter choice [1]: 1

Directory Proxy Server at localhost port number [389]:
Administrator user bind DN [cn=Directory Manager]:
Password for user 'cn=Directory Manager':
```

## Adding a New Location

First, we add a new **central** location, to which our new directory servers will be added. We select “Location” to configure locations.

```
>>>> UnboundID Directory Proxy Server configuration console
main menu

What do you want to configure?

1)  Client Connection Policy   9)  Log Publisher
2)  Connection Handler        10)  Log Retention Policy
3)  External Server           11)  Log Rotation Policy
4)  Global Attribute Index     12)  Placement Algorithm
5)  Global Configuration      13)  Proxy Transformation
6)  LDAP Health Check         14)  Request Processor
7)  Load-Balancing Algorithm  15)  Subtree View
8)  Location                  16)  Work Queue

o)  'Basic' objects are shown - change this
q)  quit

Enter choice: 8
```

We enter 2 to create a new location.

```
>>>> Location management menu

What would you like to do?

1)  List existing Locations
2)  Create a new Location
3)  View and edit an existing Location
4)  Delete an existing Location

b)  back
q)  quit

Enter choice [b]: 2
```

We create a new location from scratch that is named **central**.

```
>>>> Select an existing Location to use as a template for the
new Location configuration or 'n' to create one from scratch:

1)  east
2)  west
```

- n) new Location created from scratch
- c) cancel
- q) quit

Enter choice [n]:

>>>> Enter a name for the Location that you want to create: central

Next, we configure the **preferred-failover-location** property of the new location so that this location fails over first to the east location and then to the west location, should all of the servers in the central location become unavailable.

>>>> Configure the properties of the Location

|    | Property                                                        | Value(s) |
|----|-----------------------------------------------------------------|----------|
|    | -----                                                           | -----    |
| 1) | description                                                     | -        |
| 2) | preferred-failover-location                                     | -        |
| ?) | help                                                            |          |
| f) | finish - create the new Location                                |          |
| d) | display the equivalent dsconfig arguments to create this object |          |
| b) | back                                                            |          |
| q) | quit                                                            |          |

Enter choice [b]: 2

>>>> Configuring the 'preferred-failover-location' property

Specifies a set of alternate Locations in which servers may be accessed if no servers in this Location are available. If multiple values are provided, then the order in which the Locations are listed is the order in which they should be tried.

Do you want to modify the 'preferred-failover-location' property?

- 1) Leave undefined
- 2) Add one or more values
- ?) help
- q) quit

Enter choice [1]: 2

We need to add the east and west locations as values of the property, specifying them in the order that they will be used for failover.

Select the Locations you wish to add:

- 1) east
- 2) west
- 3) Create a new Location
- 4) Add all Locations
- ?) help
- b) back
- q) quit

Enter one or more choices separated by commas [b]: 1,2

We confirm that these are the correct values and finish configuring the location.

```
>>>> Configuring the 'preferred-failover-location' property
(Continued)
```

The 'preferred-failover-location' property references the following Locations:

- 1) east
- 2) west

Do you want to modify the 'preferred-failover-location' property?

- 1) Use these values
- 2) Add one or more values
- 3) Remove one or more values
- 4) Leave undefined
- 5) Revert changes
  
- ?) help
- q) quit

Enter choice [1]:

```
>>>> Configure the properties of the Location
```

|       | Property                                                        | Value(s)   |
|-------|-----------------------------------------------------------------|------------|
| ----- |                                                                 |            |
| 1)    | description                                                     | -          |
| 2)    | preferred-failover-location                                     | east, west |
| ?)    | help                                                            |            |
| f)    | finish - create the new Location                                |            |
| d)    | display the equivalent dsconfig arguments to create this object |            |
| b)    | back                                                            |            |
| q)    | quit                                                            |            |

Enter choice [b]: f

The Location was created successfully

## Editing The Existing Locations

Next, we edit the existing east and west locations to include the new central location in their failover logic. The new failover logic will be based on geographic distance, so that the east location will first fail over to central and then the west location.

We edit the east location.

```
>>>> Location management menu
```

What would you like to do?

- 1) List existing Locations
- 2) Create a new Location

- 3) View and edit an existing Location
- 4) Delete an existing Location
- b) back
- q) quit

Enter choice [b]: 3

>>>> Select the Location from the following list:

- 1) Central
- 2) East
- 3) West
- b) back
- q) quit

Enter choice [b]: 2

We select 2 to edit the `preferred-failover-location` property.

>>>> Configure the properties of the Location

|       | Property                                                                                                      | Value(s) |
|-------|---------------------------------------------------------------------------------------------------------------|----------|
| ----- |                                                                                                               |          |
| 1)    | description                                                                                                   | -        |
| 2)    | preferred-failover-location                                                                                   | West     |
| ?)    | help                                                                                                          |          |
| f)    | finish - apply any changes to the Location                                                                    |          |
| d)    | display the equivalent dsconfig command lines to either recreate this object or only to apply pending changes |          |
| b)    | back                                                                                                          |          |
| q)    | quit                                                                                                          |          |

Enter choice [b]: 2

We remove the `west` location from the `preferred-failover-location` property. We will add it back later.

>>>> Configuring the 'preferred-failover-location' property

Specifies a set of alternate Locations in which servers may be accessed if no servers in this Location are available. If multiple values are provided, then the order in which the Locations are listed is the order in which they should be tried.

Do you want to modify the 'preferred-failover-location' property?

- 1) Keep the value: West
- 2) Add one or more values
- 3) Remove one or more values
- 4) Leave undefined
- ?) help
- q) quit

Enter choice [1]: 3

Select the Locations you wish to remove:



- 1) west
- ?) help
- b) back
- q) quit

Enter one or more choices separated by commas [b]: 1

Now, we add a new value to the `preferred-failover-location` property.

```
>>>> Configuring the 'preferred-failover-location' property
(Continued)
```

Do you want to modify the 'preferred-failover-location' property?

- 1) Leave undefined
- 2) Add one or more values
- 3) Revert changes
- ?) help
- q) quit

Enter choice [1]: 2

We select the values of the new failover locations for the East. We enter 1 and then 3, so that east will failover first to the central location and then to the West location.

Select the Locations you wish to add:

- 1) Central
- 2) East
- 3) West
- 4) Create a new Location
- 5) Add all Locations
- ?) help
- b) back
- q) quit

Enter one or more choices separated by commas [b]: 1,3

```
>>>> Configuring the 'preferred-failover-location' property
(Continued)
```

The 'preferred-failover-location' property references the following Locations:

- 1) Central
- 2) West

Do you want to modify the 'preferred-failover-location' property?

- 1) Use these values
- 2) Add one or more values
- 3) Remove one or more values
- 4) Leave undefined
- 5) Revert changes

```
? ) help
q ) quit
```

```
Enter choice [1]:
```

We confirm the new configuration information and then enter `f` to save the changes.

```
>>>> Configure the properties of the Location
```

|    | Property                    | Value(s)      |
|----|-----------------------------|---------------|
|    | -----                       |               |
| 1) | description                 | -             |
| 2) | preferred-failover-location | Central, West |

```
? ) help
f ) finish - apply any changes to the Location
d ) display the equivalent dsconfig command lines to either
    recreate this object or only to apply pending changes
b ) back
q ) quit
```

```
Enter choice [b]: f
```

We repeat the process to reconfigure the failover logic for the **West** location to include the new **Central** location. We select “Location” from the main configuration menu and enter `3` to select the **west** location.

```
The Location was modified successfully
```

```
>>>> Location management menu
```

```
What would you like to do?
```

```
1) List existing Locations
2) Create a new Location
3) View and edit an existing Location
4) Delete an existing Location

b) back
q) quit
```

```
Enter choice [b]: 3
```

```
>>>> Select the Location from the following list:
```

```
1) Central
2) East
3) West

b) back
q) quit
```

```
Enter choice [b]: 3
```

We enter `2` to configure the **preferred-failover-location** property.

```
>>>> Configure the properties of the Location
      Property                                Value(s)
      -----
1)  description                               -
2)  preferred-failover-location  East

?)  help
f)  finish - apply any changes to the Location
d)  display the equivalent dsconfig command lines to either
    recreate this object or only to apply pending changes
b)  back
q)  quit

Enter choice [b]: 2
```

We remove the **East** location from the **preferred-failover-location** property. We will add it back later.

```
>>>> Configuring the 'preferred-failover-location' property

Specifies a set of alternate Locations in which servers may be
accessed if no servers in this Location are available. If
multiple values are provided, then the order in which the
Locations are listed is the order in which they should be tried.

Do you want to modify the 'preferred-failover-location'
property?
```

```
1)  Keep the value: East
2)  Add one or more values
3)  Remove one or more values
4)  Leave undefined

?)  help
q)  quit

Enter choice [1]: 3

Select the Locations you wish to remove:
```

```
1)  East
?)  help
b)  back
q)  quit

Enter one or more choices separated by commas [b]: 1
```

We add a new value to the **preferred-failover-location** property.

```
>>>> Configuring the 'preferred-failover-location' property
(Continued)

Do you want to modify the 'preferred-failover-location'
property?

1)  Leave undefined
2)  Add one or more values
3)  Revert changes
```

```
? ) help
q ) quit
```

```
Enter choice [1]: 2
```

We select the values of the new failover locations for the West. We enter 1 and then 3, so that west location will fail over first to the Central location and then to the East location.

```
Select the Locations you wish to add:
```

```
1) Central
2) East
3) West
4) Create a new Location
5) Add all Locations

? ) help
b ) back
q ) quit
```

```
Enter one or more choices separated by commas [b]: 1,2
```

```
>>>> Configuring the 'preferred-failover-location' property
(Continued)
```

```
The 'preferred-failover-location' property references the
following Locations:
```

```
1) Central
2) East
```

```
Do you want to modify the 'preferred-failover-location'
property?
```

```
1) Use these values
2) Add one or more values
3) Remove one or more values
4) Leave undefined
5) Revert changes

? ) help
q ) quit
```

```
Enter choice [1]:
```

We confirm the new configuration information and then enter f to save our changes.

```
>>>> Configure the properties of the Location
```

|     | Property                                                                                                         | Value(s)      |
|-----|------------------------------------------------------------------------------------------------------------------|---------------|
|     | -----                                                                                                            |               |
| 1)  | description                                                                                                      | -             |
| 2)  | preferred-failover-location                                                                                      | Central, East |
|     |                                                                                                                  |               |
| ? ) | help                                                                                                             |               |
| f)  | finish - apply any changes to the Location                                                                       |               |
| d)  | display the equivalent dsconfig command lines to either<br>recreate this object or only to apply pending changes |               |
| b)  | back                                                                                                             |               |
| q)  | quit                                                                                                             |               |

```
Enter choice [b]: f

The Location was modified successfully
```

We list the locations to confirm that the new location was added correctly.

```
>>>> Location management menu

What would you like to do?

    1) List existing Locations
    2) Create a new Location
    3) View and edit an existing Location
    4) Delete an existing Location

    b) back
    q) quit
```

```
Enter choice [b]: 1
```

```
Location : Type
-----:-----
central  : generic
east     : generic
west     : generic
```

```
Press RETURN to continue
```

```
>>>> Location management menu

What would you like to do?

    1) List existing Locations
    2) Create a new Location
    3) View and edit an existing Location
    4) Delete an existing Location

    b) back
    q) quit
```

```
Enter choice [b]:
```

## Adding New Health Checks for the Central Servers

Next, we add new health checks for the two new servers. We select “LDAP Health Check” from the main configuration menu.

```
>>>> UnboundID Directory Proxy Server configuration console
main menu
```

```
What do you want to configure?
```

- |                             |                          |
|-----------------------------|--------------------------|
| 1) Client Connection Policy | 9) Log Publisher         |
| 2) Connection Handler       | 10) Log Retention Policy |
| 3) External Server          | 11) Log Rotation Policy  |
| 4) Global Attribute Index   | 12) Placement Algorithm  |
| 5) Global Configuration     | 13) Proxy Transformation |
| 6) LDAP Health Check        | 14) Request Processor    |

- 7) Load-Balancing Algorithm    15) Subtree View
- 8) Location                      16) Work Queue
- o) 'Basic' objects are shown - change this
- q) quit

Enter choice: 6

We select 2 to create a new health check.

>>>> LDAP Health Check management menu

What would you like to do?

- 1) List existing LDAP Health Checks
- 2) Create a new LDAP Health Check
- 3) View and edit an existing LDAP Health Check
- 4) Delete an existing LDAP Health Check
- b) back
- q) quit

Enter choice [b]: 2

We select 3 to use the `ds-east-01` health check as a template for the new health check.

>>>> Select an existing LDAP Health Check to use as a template for the new LDAP Health Check configuration or 'n' to create one from scratch:

- 1) Consume Admin Alerts
- 2) Get Root DSE
- 3) `ds-east-01.example.com:389_dc_example_dc_com-search-health-check`
- 4) `ds-east-02.example.com:389_dc_example_dc_com-search-health-check`
- 5) `ds-west-01.example.com:389_dc_example_dc_com-search-health-check`
- 6) `ds-west-02.example.com:389_dc_example_dc_com-search-health-check`
- n) new LDAP Health Check created from scratch
- c) cancel
- q) quit

Enter choice [n]: 3

We name the new health check using the same naming strategy we established for the other servers in the deployment. As this health check is for the `ds-central-01` server, the name takes the following format:

>>>> Enter a name for the Search LDAP Health Check that you want to create: `ds-central-01.example.com:389_dc_example_dc_com-search-health-check`

We review the configuration properties and then enter `f` to finish configuring the new health check and save our changes.

>>>> Configure the properties of the Search LDAP Health Check

|     | Property                                                        | Value (s)           |
|-----|-----------------------------------------------------------------|---------------------|
| 1)  | description                                                     | -                   |
| 2)  | enabled                                                         | true                |
| 3)  | use-for-all-servers                                             | false               |
| 4)  | base-dn                                                         | "dc=example,dc=com" |
| 5)  | scope                                                           | base-object         |
| 6)  | filter                                                          | (objectClass=*)     |
| 7)  | maximum-local-available-response-time                           | 1 s                 |
| 8)  | maximum-nonlocal-available-response-time                        | 1 s                 |
| 9)  | minimum-local-degraded-response-time                            | 500 ms              |
| 10) | minimum-nonlocal-degraded-response-time                         | 500 ms              |
| 11) | maximum-local-degraded-response-time                            | 10 s                |
| 12) | maximum-nonlocal-degraded-response-time                         | 10 s                |
| 13) | minimum-local-unavailable-response-time                         | 5 s                 |
| 14) | minimum-nonlocal-unavailable-response-time                      | 5 s                 |
| 15) | allow-no-entries-returned                                       | true                |
| 16) | allow-multiple-entries-returned                                 | true                |
| 17) | available-filter                                                | -                   |
| 18) | degraded-filter                                                 | -                   |
| 19) | unavailable-filter                                              | -                   |
| ?)  | help                                                            |                     |
| f)  | finish - create the new Search LDAP Health Check                |                     |
| d)  | display the equivalent dsconfig arguments to create this object |                     |
| b)  | back                                                            |                     |
| q)  | quit                                                            |                     |

Enter choice [b]: f

The Search LDAP Health Check was created successfully

We follow the same steps to create another new health check for the **ds-central-02** server.

>>>> LDAP Health Check management menu

What would you like to do?

- 1) List existing LDAP Health Checks
- 2) Create a new LDAP Health Check
- 3) View and edit an existing LDAP Health Check
- 4) Delete an existing LDAP Health Check
- b) back
- q) quit

Enter choice [b]: 2

As for the previous health check, we select 3 to use the **ds-central-03** health check as a template for the new health check.

>>>> Select an existing LDAP Health Check to use as a template for the new LDAP Health Check configuration or 'n' to create one from scratch:

- 1) Consume Admin Alerts
- 2) Get Root DSE
- 3) ds-central-01.example.com:389\_dc\_example\_dc\_com-search-health-check

- 4) ds-east-01.example.com:389\_dc\_example\_dc\_com-search health-check
- 5) ds-east-02.example.com:389\_dc\_example\_dc\_com-search health-check
- 6) ds-west-01.example.com:389\_dc\_example\_dc\_com-search health-check
- 7) ds-west-02.example.com:389\_dc\_example\_dc\_com-search health-check
- n) new LDAP Health Check created from scratch
- c) cancel
- q) quit

Enter choice [n]: 3

We name the new health check as before, though this time the server is **ds-central-02**.

```
>>>> Enter a name for the Search LDAP Health Check that you want to
create:
ds-central-02.example.com:389_dc_example_dc_com_search-health-check
```

We review the configuration properties and then enter **f** to finish configuring the new health check and save our changes.

```
>>>> Configure the properties of the Search LDAP Health Check
```

|     | Property                                                        | Value(s)            |
|-----|-----------------------------------------------------------------|---------------------|
| 1)  | description                                                     | -                   |
| 2)  | enabled                                                         | true                |
| 3)  | use-for-all-servers                                             | false               |
| 4)  | base-dn                                                         | "dc=example,dc=com" |
| 5)  | scope                                                           | base-object         |
| 6)  | filter                                                          | (objectClass=*)     |
| 7)  | maximum-local-available-response-time                           | 1 s                 |
| 8)  | maximum-nonlocal-available-response-time                        | 1 s                 |
| 9)  | minimum-local-degraded-response-time                            | 500 ms              |
| 10) | minimum-nonlocal-degraded-response-time                         | 500 ms              |
| 11) | maximum-local-degraded-response-time                            | 10 s                |
| 12) | maximum-nonlocal-degraded-response-time                         | 10 s                |
| 13) | minimum-local-unavailable-response-time                         | 5 s                 |
| 14) | minimum-nonlocal-unavailable-response-time                      | 5 s                 |
| 15) | allow-no-entries-returned                                       | true                |
| 16) | allow-multiple-entries-returned                                 | true                |
| 17) | available-filter                                                | -                   |
| 18) | degraded-filter                                                 | -                   |
| 19) | unavailable-filter                                              | -                   |
| ?)  | help                                                            |                     |
| f)  | finish - create the new Search LDAP Health Check                |                     |
| d)  | display the equivalent dsconfig arguments to create this object |                     |
| b)  | back                                                            |                     |
| q)  | quit                                                            |                     |

Enter choice [b]: f

The Search LDAP Health Check was created successfully



```
>>>> LDAP Health Check management menu

What would you like to do?

    1) List existing LDAP Health Checks
    2) Create a new LDAP Health Check
    3) View and edit an existing LDAP Health Check
    4) Delete an existing LDAP Health Check

    b) back
    q) quit

Enter choice [b]:
```

## Adding New External Servers

Next, we add new external servers by selecting “External Server” from the configuration console main menu.

```
>>>> UnboundID Directory Proxy Server configuration console
main menu

What do you want to configure?

    1) Client Connection Policy    9) Log Publisher
    2) Connection Handler          10) Log Retention Policy
    3) External Server             11) Log Rotation Policy
    4) Global Attribute Index      12) Placement Algorithm
    5) Global Configuration        13) Proxy Transformation
    6) LDAP Health Check          14) Request Processor
    7) Load-Balancing Algorithm   15) Subtree View
    8) Location                   16) Work Queue

    o) 'Basic' objects are shown - change this
    q) quit

Enter choice: 3
```

We enter 2 to create a new external server.

```
>>>> External Server management menu

What would you like to do?

    1) List existing External Servers
    2) Create a new External Server
    3) View and edit an existing External Server
    4) Delete an existing External Server

    b) back
    q) quit

Enter choice [b]: 2
```

We will base the configuration of the new external server on the existing configuration of the `ds-east-01` server, so we select 1.

```
>>>> Select an existing External Server to use as a template for
the new External Server configuration or 'n' to create one from
scratch:
```

- 1) ds-east-01.example.com:389
- 2) ds-east-02.example.com:389
- 3) ds-west-01.example.com:389
- 4) ds-west-02.example.com:389
  
- n) new External Server created from scratch
- c) cancel
- q) quit

```
Enter choice [n]: 1
```

We enter a name for the new **ds-central-01** server that complies with our naming strategy.

```
>>>> Enter a name for the UnboundID DS External Server that you
want to create: ds-central-01.example.com:389
```

We review and modify the configuration properties of the external server.

```
>>>> Configure the properties of the UnboundID DS External Server
```

| Property                                                        | Value(s)                             |
|-----------------------------------------------------------------|--------------------------------------|
| -----                                                           |                                      |
| 1) description                                                  | -                                    |
| 2) server-host-name                                             | ds-east-01.example.com               |
| 3) server-port                                                  | 389                                  |
| 4) location                                                     | east                                 |
| 5) bind-dn                                                      | cn=Proxy User                        |
| 6) password                                                     | *****                                |
| 7) connection-security                                          | starttls                             |
| 8) authentication-method                                        | simple                               |
| 9) health-check                                                 | ds-east-01.example.com:389_dc_       |
|                                                                 | example_dc_com-search-health-check   |
| 10) health-check-frequency                                      | 30 s                                 |
| 11) allowed-operation                                           | abandon, add, bind, compare, delete, |
|                                                                 | extended, modify, modify-dn, search  |
| 12) key-manager-provider                                        | Null                                 |
| 13) trust-manager-provider                                      | JKS                                  |
|                                                                 |                                      |
| ? ) help                                                        |                                      |
| f) finish - create the new UnboundID DS External Server         |                                      |
| a) show advanced properties of the UnboundID DS External Server |                                      |
| d) display the equivalent dsconfig arguments to create this     |                                      |
| object                                                          |                                      |
| b) back                                                         |                                      |
| q) quit                                                         |                                      |

```
Enter choice [b]: 2
```

First, we change the **server-host-name** property to reflect the name of the **ds-central-01** server.

```
>>>> Configuring the 'server-host-name' property
```

```
The host name or IP address of the target LDAP server.
```

```
Syntax:  STRING
```

Do you want to modify the 'server-host-name' property?

- 1) Keep the value: ds-east-01.example.com
- 2) Change the value
- ?) help
- q) quit

Enter choice [1]: 2

Enter a value for the 'server-host-name' property: ds-central-01.example.com

Next, we change the `location` property to reflect the central location.

>>>> Configure the properties of the UnboundID DS External Server

| Property                                                              | Value(s)                                                                    |
|-----------------------------------------------------------------------|-----------------------------------------------------------------------------|
| -----                                                                 |                                                                             |
| 1) description                                                        | -                                                                           |
| 2) server-host-name                                                   | ds-central-01.example.com                                                   |
| 3) server-port                                                        | 389                                                                         |
| 4) location                                                           | east                                                                        |
| 5) bind-dn                                                            | cn=Proxy User                                                               |
| 6) password                                                           | *****                                                                       |
| 7) connection-security                                                | starttls                                                                    |
| 8) authentication-method                                              | simple                                                                      |
| 9) health-check                                                       | ds-east-01.example.com:389_dc_<br>example_dc_com-search-health-check        |
| 10) health-check-frequency                                            | 30 s                                                                        |
| 11) allowed-operation                                                 | abandon, add, bind, compare, delete,<br>extended, modify, modify-dn, search |
| 12) key-manager-provider                                              | Null                                                                        |
| 13) trust-manager-provider                                            | JKS                                                                         |
| ?) help                                                               |                                                                             |
| f) finish - create the new UnboundID DS External Server               |                                                                             |
| a) show advanced properties of the UnboundID DS External Server       |                                                                             |
| d) display the equivalent dsconfig arguments to create this<br>object |                                                                             |
| b) back                                                               |                                                                             |
| q) quit                                                               |                                                                             |

Enter choice [b]: 4

>>>> Configuring the 'location' property

Specifies the location for the LDAP External Server.

Do you want to modify the 'location' property?

- 1) Keep the value: east
- 2) Change it to the Location: central
- 3) Change it to the Location: west
- 4) Create a new Location
- 5) Leave undefined
- ?) help
- q) quit

Enter choice [1]: 2

Next, we change the `health-check` property to reflect the new health check we created for the `ds-central-01` server in the previous section.

```
>>>> Configure the properties of the UnboundID DS External Server
      Property                                Value(s)
-----
1)  description                             -
2)  server-host-name                        ds-central-01.example.com
3)  server-port                             389
4)  location                               central
5)  bind-dn                                cn=Proxy User
6)  password                               *****
7)  connection-security                    starttls
8)  authentication-method                   simple
9)  health-check                           ds-east-01.example.com:389_dc_
example_dc_com-search-health-check
10) health-check-frequency                  30 s
11) allowed-operation                       abandon, add, bind, compare, delete,
extended, modify, modify-dn, search
12) key-manager-provider                    Null
13) trust-manager-provider                  JKS

?)  help
f)  finish - create the new UnboundID DS External Server
a)  show advanced properties of the UnboundID DS External
    Server
d)  display the equivalent dsconfig arguments to create this
    object
b)  back
q)  quit

Enter choice [b]: 9
```

We select 3 to remove the old value of the property.

```
>>>> Configuring the 'health-check' property

Specifies the health check to use for the LDAP External Server.

Do you want to modify the 'health-check' property?

1)  Keep the value:

    ds-east-01.example.com:389_dc_example_dc_com-search-
    health-check
2)  Add one or more values
3)  Remove one or more values
4)  Leave undefined

?)  help
q)  quit

Enter choice [1]: 3

Select the LDAP Health Checks you wish to remove:

1)  ds-east-01.example.com:389_dc_example_dc_com-search-
    health-check

?)  help
b)  back
q)  quit

Enter one or more choices separated by commas [b]: 1
```

Now we add the health-check entered in the previous section.

```
>>>> Configuring the 'health-check' property (Continued)
```

```
Do you want to modify the 'health-check' property?
```

- 1) Leave undefined
- 2) Add one or more values
- 3) Revert changes

```
? ) help
```

```
q ) quit
```

```
Enter choice [1]: 2
```

We select 3 to choose the health check associated with the **ds-central-01** server.

```
Select the LDAP Health Checks you wish to add:
```

- |                                                                        |                                                                     |
|------------------------------------------------------------------------|---------------------------------------------------------------------|
| 1) Consume Admin Alerts                                                | 6) ds-east-02.example.com:389_dc_example_dc_com-search-health-check |
| 2) Get Root DSE                                                        | 7) ds-west-01.example.com:389_dc_example_dc_com-search-health-check |
| 3) ds-central-01.example.com:389_dc_example_dc_com-search-health-check | 8) ds-west-02.example.com:389_dc_example_dc_com-search-health-check |
| 4) ds-central-02.example.com:389_dc_example_dc_com-search-health-check | 9) Create a new LDAP Health Check                                   |
| 5) ds-east-01.example.com:389_dc_example_dc_com-search-health-check    | 10) Add all LDAP Health Checks                                      |
| ? ) help                                                               |                                                                     |
| b) back                                                                |                                                                     |
| q) quit                                                                |                                                                     |

```
Enter one or more choices separated by commas [b]: 3
```

We select 1 to use the value associated with the **ds-central-01** health check.

```
>>>> Configuring the 'health-check' property (Continued)
```

```
Do you want to modify the 'health-check' property?
```

- 1) Use the value:  
ds-central-01.example.com:389\_dc\_example\_dc\_com-search-health-check
- 2) Add one or more values
- 3) Remove one or more values
- 4) Leave undefined
- 5) Revert changes

```
? ) help
```

```
q ) quit
```

```
Enter choice [1]:
```

We review the configuration of the new external server and enter f to create the server.

```
>>>> Configure the properties of the UnboundID DS External Server
```

|     | Property                                                        | Value(s)                                                                    |
|-----|-----------------------------------------------------------------|-----------------------------------------------------------------------------|
| 1)  | description                                                     | -                                                                           |
| 2)  | server-host-name                                                | ds-central-01.example.com                                                   |
| 3)  | server-port                                                     | 389                                                                         |
| 4)  | location                                                        | central                                                                     |
| 5)  | bind-dn                                                         | cn=Proxy User                                                               |
| 6)  | password                                                        | *****                                                                       |
| 7)  | connection-security                                             | starttls                                                                    |
| 8)  | authentication-method                                           | simple                                                                      |
| 9)  | health-check                                                    | ds-central-01.example.com:389_<br>dc_example_dc_com-search-health-check     |
| 10) | health-check-frequency                                          | 30 s                                                                        |
| 11) | allowed-operation                                               | abandon, add, bind, compare, delete,<br>extended, modify, modify-dn, search |
| 12) | key-manager-provider                                            | Null                                                                        |
| 13) | trust-manager-provider                                          | JKS                                                                         |
| ?)  | help                                                            |                                                                             |
| f)  | finish - create the new UnboundID DS External Server            |                                                                             |
| a)  | show advanced properties of the UnboundID DS External Server    |                                                                             |
| d)  | display the equivalent dsconfig arguments to create this object |                                                                             |
| b)  | back                                                            |                                                                             |
| q)  | quit                                                            |                                                                             |

Enter choice [b]: f

The UnboundID DS External Server was created successfully

We follow the same procedure to add the new **ds-central-02** external server.

```
>>>> External Server management menu
```

What would you like to do?

- 1) List existing External Servers
- 2) Create a new External Server
- 3) View and edit an existing External Server
- 4) Delete an existing External Server
- b) back
- q) quit

Enter choice [b]: 2

This time, we select the new **ds-central-01** server configuration as a template by entering 1 and then enter the name of the new server.

```
>>>> Select an existing External Server to use as a template for  
the new External Server configuration or 'n' to create one from  
scratch:
```

- 1) ds-central-01.example.com:389
- 2) ds-east-01.example.com:389
- 3) ds-east-02.example.com:389
- 4) ds-west-01.example.com:389
- 5) ds-west-02.example.com:389

- n) new External Server created from scratch
- c) cancel
- q) quit

Enter choice [n]: 1

>>>> Enter a name for the UnboundID DS External Server that you want to create: ds-central-02.example.com:389

We modify the `server-host-name` property to reflect the name of the `ds-central-02` server.

>>>> Configure the properties of the UnboundID DS External Server

|       | Property                                                        | Value(s)                                                                    |
|-------|-----------------------------------------------------------------|-----------------------------------------------------------------------------|
| ----- |                                                                 |                                                                             |
| 1)    | description                                                     | -                                                                           |
| 2)    | server-host-name                                                | ds-central-01.example.com                                                   |
| 3)    | server-port                                                     | 389                                                                         |
| 4)    | location                                                        | central                                                                     |
| 5)    | bind-dn                                                         | cn=Proxy User\                                                              |
| 6)    | password                                                        | *****                                                                       |
| 7)    | connection-security                                             | starttls                                                                    |
| 8)    | authentication-method                                           | simple                                                                      |
| 9)    | health-check                                                    | ds-central-01.example.com:389<br>_dc_example_dc_com-search-health-check     |
| 10)   | health-check-frequency                                          | 30 s                                                                        |
| 11)   | allowed-operation                                               | abandon, add, bind, compare, delete,<br>extended, modify, modify-dn, search |
| 12)   | key-manager-provider                                            | Null                                                                        |
| 13)   | trust-manager-provider                                          | JKS                                                                         |
| ?)    | help                                                            |                                                                             |
| f)    | finish - create the new UnboundID DS External Server            |                                                                             |
| a)    | show advanced properties of the UnboundID DS External Server    |                                                                             |
| d)    | display the equivalent dsconfig arguments to create this object |                                                                             |
| b)    | back                                                            |                                                                             |
| q)    | quit                                                            |                                                                             |

Enter choice [b]: 2

>>>> Configuring the 'server-host-name' property

The host name or IP address of the target LDAP server.

Syntax: STRING

Do you want to modify the 'server-host-name' property?

- 1) Keep the value: ds-central-01.example.com
- 2) Change the value
- ?) help
- q) quit

Enter choice [1]: 2

Enter a value for the 'server-host-name' property: ds-central-02.example.com

We change the **health-check** property to reflect the new health check we created for the **ds-central-02** server in the previous section.

```
>>>> Configure the properties of the UnboundID DS External Server
```

|     | Property                                                        | Value(s)                                                                    |
|-----|-----------------------------------------------------------------|-----------------------------------------------------------------------------|
| 1)  | description                                                     | -                                                                           |
| 2)  | server-host-name                                                | ds-central-02.example.com                                                   |
| 3)  | server-port                                                     | 389                                                                         |
| 4)  | location                                                        | central                                                                     |
| 5)  | bind-dn                                                         | cn=Proxy User\                                                              |
| 6)  | password                                                        | *****                                                                       |
| 7)  | connection-security                                             | starttls                                                                    |
| 8)  | authentication-method                                           | simple                                                                      |
| 9)  | health-check                                                    | ds-central-01.example.com:389<br>_dc_example_dc_com-search-health-check     |
| 10) | health-check-frequency                                          | 30 s                                                                        |
| 11) | allowed-operation                                               | abandon, add, bind, compare, delete,<br>extended, modify, modify-dn, search |
| 12) | key-manager-provider                                            | Null                                                                        |
| 13) | trust-manager-provider                                          | JKS                                                                         |
| ?)  | help                                                            |                                                                             |
| f)  | finish - create the new UnboundID DS External Server            |                                                                             |
| a)  | show advanced properties of the UnboundID DS External Server    |                                                                             |
| d)  | display the equivalent dsconfig arguments to create this object |                                                                             |
| b)  | back                                                            |                                                                             |
| q)  | quit                                                            |                                                                             |

```
Enter choice [b]: 9
```

First, we delete the old health-check property value.

```
>>>> Configuring the 'health-check' property
```

```
Specifies the health check to use for the LDAP External Server.
```

```
Do you want to modify the 'health-check' property?
```

- 1) Keep the value:  
ds-central-01.example.com:389\_dc\_example\_dc\_com-search-health-check
- 2) Add one or more values
- 3) Remove one or more values
- 4) Leave undefined
- ?) help
- q) quit

```
Enter choice [1]: 3
```

```
Select the LDAP Health Checks you wish to remove:
```

- 1) ds-central-01.example.com:389\_dc\_example\_dc\_com-search-health-check



```
? ) help
b ) back
q ) quit
```

Enter one or more choices separated by commas [b]: 1

Then, we add the new health-check property value for `ds-central-02`.

```
>>>> Configuring the 'health-check' property (Continued)
```

Do you want to modify the 'health-check' property?

```
1) Leave undefined
2) Add one or more values
3) Revert changes

? ) help
q ) quit
```

Enter choice [1]: 2

Select the LDAP Health Checks you wish to add:

```
1) Consume Admin Alerts           6) ds-east-02.example.com:389_
                                   dc_example_dc_com-search-
                                   health-check
2) Get Root DSE                   7) ds-west-01.example.com:389_
                                   dc_example_dc_com-search-
                                   health-check
3) ds-central-01.example.com:389_  8) ds-west-02.example.com:389_
   dc_example_dc_com-search-health-  dc_example_dc_com-search-
   check                             health-check
4) ds-central-02.example.com:389_  9) Create a new LDAP Health
   dc_example_dc_com-search-health- Check
   check
5) ds-east-01.example.com:389_dc_ 10) Add all LDAP Health Checks
   example_dc_com-search-health-check

? ) help
b ) back
q ) quit
```

Enter one or more choices separated by commas [b]: 4

We select 1 to use the value associated with the `ds-central-02` health check.

```
>>>> Configuring the 'health-check' property (Continued)
```

Do you want to modify the 'health-check' property?

```
1) Use the value:
   ds-central-02.example.com:389_dc_example_dc_com-search-
   health-check
2) Add one or more values
3) Remove one or more values
4) Leave undefined
5) Revert changes

? ) help
q ) quit
```

Enter choice [1]:

We review the configuration of the new external server and enter f to create the server.

```
>>>> Configure the properties of the UnboundID DS External Server

      Property                                Value(s)
      -----
1)  description                               -
2)  server-host-name                         ds-central-02.example.com
3)  server-port                              389
4)  location                                 central
5)  bind-dn                                  cn=Proxy User
6)  password                                 *****
7)  connection-security                     starttls
8)  authentication-method                    simple
9)  health-check                             ds-central-02.example.com:389_
   dc_example_dc_com-search-health-check
10) health-check-frequency                    30 s
11) allowed-operation                        abandon, add, bind, compare, delete,
   extended, modify, modify-dn, search

12) key-manager-provider                     Null
13) trust-manager-provider                   JKS

?)  help
f)  finish - create the new UnboundID DS External Server
a)  show advanced properties of the UnboundID DS External
    Server
d)  display the equivalent dsconfig arguments to create this
    object
b)  back
q)  quit

Enter choice [b]: f

The UnboundID DS External Server was created successfully
```

## Modifying the Load-Balancing Algorithm

We now need to modify the existing load-balancing algorithm to include the newly created servers, so we select “Load-Balancing Algorithm” from the configuration console main menu.

```
>>>> UnboundID Directory Proxy Server configuration console
main menu

What do you want to configure?
  1) Client Connection Policy   9) Log Publisher
  2) Connection Handler        10) Log Retention Policy
  3) External Server           11) Log Rotation Policy
  4) Global Attribute Index     12) Placement Algorithm
  5) Global Configuration       13) Proxy Transformation
  6) LDAP Health Check          14) Request Processor
  7) Load-Balancing Algorithm  15) Subtree View
  8) Location                   16) Work Queue

  o) 'Basic' objects are shown - change this
  q) quit

Enter choice: 7
```

We select 3 to edit the existing load-balancing algorithm.

```
>>>> Load-Balancing Algorithm management menu

What would you like to do?

1) List existing Load-Balancing Algorithms
2) Create a new Load-Balancing Algorithm
3) View and edit an existing Load-Balancing Algorithm
4) Delete an existing Load-Balancing Algorithm

b) back
q) quit
```

Enter choice [b]: 3

```
>>>> There is only one Load-Balancing Algorithm:

'dc_example_dc_com-round-robin'. Are you sure that this is the
correct one? (yes / no) [yes]:
```

We now add the `ds-central-01` and `ds-central-02` servers to the `backend-server` configuration property of the round robin load-balancing algorithm.

```
>>>> Configure the properties of the Round Robin Load-Balancing Algo-
rithm
```

|     | Property                                                                                                       | Value(s)                                                                                                                |
|-----|----------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------|
| 1)  | description                                                                                                    | -                                                                                                                       |
| 2)  | enabled                                                                                                        | true                                                                                                                    |
| 3)  | backend-server                                                                                                 | ds-east-01.example.com:389,<br>ds-east-02.example.com:389,<br>ds-west-01.example.com:389,<br>ds-west-02.example.com:389 |
| 4)  | use-location                                                                                                   | true                                                                                                                    |
| 5)  | prefer-degraded-servers-over-failover                                                                          | false                                                                                                                   |
| 6)  | maximum-allowed-local-response-time                                                                            | 30 s                                                                                                                    |
| 7)  | maximum-allowed-nonlocal-response-time                                                                         | 30 s                                                                                                                    |
| 8)  | maximum-retry-count                                                                                            | 1                                                                                                                       |
| 9)  | initial-connections                                                                                            | 10                                                                                                                      |
| 10) | max-connections                                                                                                | 64                                                                                                                      |
| ?)  | help                                                                                                           |                                                                                                                         |
| f)  | finish - apply any changes to the Round Robin Load Balancing Algorithm                                         |                                                                                                                         |
| d)  | display the equivalent dsconfig command lines to either re-create this object or only to apply pending changes |                                                                                                                         |
| b)  | back                                                                                                           |                                                                                                                         |
| q)  | quit                                                                                                           |                                                                                                                         |

Enter choice [b]: 3

We enter 2 to add one or more values.

```
>>>> Configuring the 'backend-server' property

Specifies the set of backend servers that will be available
to process forwarded requests.
```

The 'backend-server' property references the following LDAP External Servers:

- \*) ds-east-01.example.com:389
- \*) ds-east-02.example.com:389
- \*) ds-west-01.example.com:389
- \*) ds-west-02.example.com:389

Do you want to modify the 'backend-server' property?

- 1) Keep these values
- 2) Add one or more values
- 3) Remove one or more values
- 4) Remove all values
- ?) help
- q) quit

Enter choice [1]: 2

Next, we enter 1 and 2 to select the **ds-central-01** and **ds-central-02** external servers.

Select the LDAP External Servers you wish to add:

- 1) ds-central-01.example.com:389
- 2) ds-central-02.example.com:389
- 3) Create a new LDAP External Server
- 4) Add all LDAP External Servers
- ?) help
- b) back
- q) quit

Enter one or more choices separated by commas [b]: 1,2

>>>> Configuring the 'backend-server' property (Continued)

The 'backend-server' property references the following LDAP External Servers:

- \*) ds-central-01.example.com:389
- \*) ds-central-02.example.com:389
- \*) ds-east-01.example.com:389
- \*) ds-east-02.example.com:389
- \*) ds-west-01.example.com:389
- \*) ds-west-02.example.com:389

Do you want to modify the 'backend-server' property?

- 1) Use these values
- 2) Add one or more values
- 3) Remove one or more values
- 4) Remove all values
- 5) Revert changes
- ?) help
- q) quit

Enter choice [1]:

We review the changes made to the load-balancing algorithm's configuration properties. When we are satisfied, we enter f to save our changes.

```
>>>> Configure the properties of the Round Robin Load-Balancing Algorithm
```

|     | Property                                                                                                       | Value(s)                                                                                                                                                                                    |
|-----|----------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|     | -----                                                                                                          | -----                                                                                                                                                                                       |
| 1)  | description                                                                                                    | -                                                                                                                                                                                           |
| 2)  | enabled                                                                                                        | true                                                                                                                                                                                        |
| 3)  | backend-server                                                                                                 | ds-central-01.example.com:389,<br>ds-central-02.example.com:389,<br>ds-east-01.example.com:389,<br>ds-east-02.example.com:389,<br>ds-west-01.example.com:389,<br>ds-west-02.example.com:389 |
| 4)  | use-location                                                                                                   | true                                                                                                                                                                                        |
| 5)  | prefer-degraded-servers-over-failover                                                                          | false                                                                                                                                                                                       |
| 6)  | maximum-allowed-local-response-time                                                                            | 30 s                                                                                                                                                                                        |
| 7)  | maximum-allowed-nonlocal-response-time                                                                         | 30 s                                                                                                                                                                                        |
| 8)  | maximum-retry-count                                                                                            | 1                                                                                                                                                                                           |
| 9)  | initial-connections                                                                                            | 10                                                                                                                                                                                          |
| 10) | max-connections                                                                                                | 64                                                                                                                                                                                          |
| ?)  | help                                                                                                           |                                                                                                                                                                                             |
| f)  | finish - apply any changes to the Round Robin Load Balancing Algorithm                                         |                                                                                                                                                                                             |
| d)  | display the equivalent dsconfig command lines to either re-create this object or only to apply pending changes |                                                                                                                                                                                             |
| b)  | back                                                                                                           |                                                                                                                                                                                             |
| q)  | quit                                                                                                           |                                                                                                                                                                                             |

```
Enter choice [b]: f
```

The Round Robin Load-Balancing Algorithm was modified successfully

```
>>>> Load-Balancing Algorithm management menu
```

What would you like to do?

- 1) List existing Load-Balancing Algorithms
- 2) Create a new Load-Balancing Algorithm
- 3) View and edit an existing Load-Balancing Algorithm
- 4) Delete an existing Load-Balancing Algorithm
- b) back
- q) quit

```
Enter choice [b]:
```

The change has been saved and applied to the directory proxy server. The load-balancing algorithm is referenced in the `load-balancing-algorithm` property of the request processor used by this directory proxy server. To view this property, go to the main configuration console menu and select "Request Processor".

```
>>>> UnboundID Directory Proxy Server configuration console  
main menu
```

What do you want to configure?

- 1) Client Connection Policy
- 2) Connection Handler
- 10) Log Retention Policy
- 11) Log Rotation Policy

- 3) External Server
  - 4) Global Attribute Index
  - 5) Global Configuration
  - 6) LDAP Health Check
  - 7) Load Balancing Algorithm
  - 8) Location
  - 9) Log Publisher
  - 12) Placement Algorithm
  - 13) Proxy Transformation
  - 14) Request Processor
  - 15) Server Affinity Provider
  - 16) Subtree View
  - 17) Work Queue
- o) 'Basic' objects are shown - change this  
q) quit

Enter choice: 14

Select 3 to view and edit an existing request processor.

>>>> Request Processor management menu

What would you like to do?

- 1) List existing Request Processors
  - 2) Create a new Request Processor
  - 3) View and edit an existing Request Processor
  - 4) Delete an existing Request Processor
- b) back  
q) quit

Enter choice [b]: 3

Select the request processor used by your proxy server. The configuration properties are displayed as follows:

>>>> Configure the properties of the Proxying Request Processor

| Property                    | Value(s)                                                                                                                                                                                                                                                                                                                                                                                         |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -----                       |                                                                                                                                                                                                                                                                                                                                                                                                  |
| 1) description              | -                                                                                                                                                                                                                                                                                                                                                                                                |
| 2) enabled                  | true                                                                                                                                                                                                                                                                                                                                                                                             |
| 3) allowed-operation        | abandon, add, bind, compare, delete, extended, modify, modify-dn, search                                                                                                                                                                                                                                                                                                                         |
| 4) load-balancing-algorithm | dc_example_dc_com-round-robin                                                                                                                                                                                                                                                                                                                                                                    |
| 5) transformation           | -                                                                                                                                                                                                                                                                                                                                                                                                |
| 6) referral-behavior        | pass-through                                                                                                                                                                                                                                                                                                                                                                                     |
| 7) supported-control        | account-usable, assertion, authorization-identity, get-authorization-entry, get-effective-rights, ignore-no-user-modification, intermediate-client, manage-dsa-it, matched-values, no-op, password-policy, post-read, pre-read, proxied-authorization-v1, proxied-authorization-v2, real-attributes-only, retain-identity, server-side-sort, subentries, subtree delete, virtual-attributes-only |
| 8) supported-control-oid    | -                                                                                                                                                                                                                                                                                                                                                                                                |

```
? ) help
f ) finish - apply any changes to the Proxying Request Processor
d ) display the equivalent dsconfig command lines to either
    re-create this object or only to apply pending changes
b ) back
q ) quit
```

Enter choice [b]:

This request processor is used by the subtree view serviced by the directory proxy server, which is in turn referenced by the client connection policy.

Note that the changes made here are already in effect. The directory proxy server does not have to be restarted.

## Testing External Server Communication After Adding New External Servers

After adding and configuring the new external servers, test the communication between the directory proxy server and the LDAP external servers using the **include-backend-server-passthrough-subtree-views** property of the directory proxy server in combination with an LDAP search. For more information about this option, see “Testing External Server Communications After Initial Setup” on page 208.

We run the following search to test communications on the **ds-central-01** server:

```
root@proxy-east-01: bin/ldapsearch --port 389 --bindDN "cn=directory manager" \
--bindPassword password \
--baseDN "dc=example,dc=com,ds-backend-server=ds-central-01.example.com:389" \
--searchScope base "(objectclass=*)"
```

You can repeat this search on the **ds-central-02** server, to confirm that the server returns the entry as expected.

## Testing a Simulated External Server Failure

Once you have tested connectivity, run a simulated failure of a load-balanced external server to verify that the directory proxy server redirects LDAP requests appropriately.

We stop the **ds-east-01.example.com:389** and **ds-east-02.example.com:389** server instances and test searches through **proxy-east-01.example.com**.

First, perform several searches against the proxy. Verify activity in each of the servers in the east location, **ds-east-01** and **ds-east-02**, by looking at the access logs. The following simple search can be repeated as needed:

```
root@proxy-east-01: bin/ldapsearch --bindDN "cn=Directory Manager" \
--bindPassword password --baseDN "dc=example,dc=com" \
--searchScope base --useStartTLS "(objectclass=*)"
```

Next, stop the directory server instance on **ds-east-01.example.com** and **ds-east-02.example.com** using the **stop-ds** command and immediately retry the above searches. There should be no errors or noticeable delay in processing the search.

```
root@proxy-east-01: bin/stop-ds

root@proxy-east-01: bin/ldapsearch \
--bindDN "cn=Directory Manager" --bindPassword password \
--baseDN "dc=example,dc=com" --searchScope base --useStartTLS \
"(objectclass=*)"
```

Check the access log to confirm that requests made to these servers are routed to the central servers, as these servers are the first failover location in the failover list for the **ds-east-01** and **ds-east-02** servers.

Restart the directory server instance on **ds-east-01.example.com** and **ds-east-02.example.com**. Check their access logs to ensure that traffic is redirected back from the failover servers.

## Merging Two Data Sets Using Proxy Transformations

In this example, the Example.com company acquires Sample Corporation. During the merger, Example.com migrates data from Sample's **o=sample** rooted directory, converting Sample's **sampleAccount** auxiliary object class usage to Example.com's **exampleAccount** object class for entries rooted under **dc=example,dc=com**. Knowing that it can take considerable time for Sample's directory clients to become aware of the new DIT and schema, proxy data transformations are created to give the Sample clients as consistent a view of the data as possible during the migratory period. These transformations allow the clients to search and modify entries under **o=sample** using the Sample Corp. schema.

The following section describes how the proxy data transformations will be configured.

### Overview of the Attribute and DN Mapping

To achieve the merger of the two data sets, we create proxy transformations that map the Sample source attributes to Example.com target attributes as described in Table 10-1, "Attribute Mapping". The Example.com schema already defines an attribute to contain the

**TABLE 10-1. Attribute Mapping**

| Sample Attribute             | Example.com Attribute          | Description                          |
|------------------------------|--------------------------------|--------------------------------------|
| <b>sampleID</b>              | <b>uid</b>                     | RDN of user entries                  |
| <b>sampleRegion</b>          | <b>exSampleRegion</b>          | String value representing the region |
| <b>sampleLinkedAc-counts</b> | <b>exSampleLinkedAc-counts</b> | DN value                             |



RDN of user entries, called `uid`. However, Example.com chooses to create two new attributes within its `exampleAccount` object class to accommodate two attributes in the Sample schema for representing the region and the DN of linked accounts.

During the merger, Example.com decides to re-parent Sample’s customer entries, which are defined under two different subtrees, `ou=east,o=sample` and `ou=west,o=sample`, placing them under Example.com’s `ou=people,dc=example,dc=com` subtree. Associated proxy transformations are described in Table 10-2, “Mapping DNs”. In this process, Example.com collapses the Sample tree, moving entries from the east and west region under a single DN, `dc=example,dc=com`. The DN proxy transformations assume that all Sample users have been co-located under this single Example.com subtree.

Legacy Sample LDAP applications searching for entries in either the Sample base DN `ou=east,o=sample` or `ou=west,o=sample` will be successfully serviced, though there will be one or more differences in the user entries seen by the Sample legacy applications. Since the Example.com directory has no knowledge of the Sample user’s former `ou=east` or `ou=west` association, search results for client searching under `o=sample` will return a DN that may differ from the original search base. For instance, as search for `sampleID=abc123`

TABLE 10-2. Mapping DNs

| Sample DN                     | Example.com DN                 |
|-------------------------------|--------------------------------|
| <code>ou=east,o=sample</code> | <code>dc=example,dc=com</code> |
| <code>ou=west,o=sample</code> | <code>dc=example,dc=com</code> |
| <code>o=sample</code>         | <code>dc=example,dc=com</code> |

under

`ou=west,o=sample` may return the user entry for `abc123` with the DN of `sampleID=abc123,ou=east,o=sample`. Special considerations for mapping two different source DNs to the same target DN are discussed in the following section.

About Mapping Multiple Source DNs to the Same Target DN

Some complications exist when defining multiple DN mappings that are used for the same request processor and the same source or target DN (or that have source or target DNs that are hierarchically related). The client request may not include enough information to disambiguate and determine the proper rule to follow.

Several solutions exist to avoid problems of disambiguation. If the client does not need to be able to see all mappings at the same time, then a new client connection policy can be created to use connection criteria that select the set of mappings applied to the client based on information such as the IP address or bind DN. Each client connection policy would have separated subtree views with separate proxying request processors that reference the appropriate transformation for that client.

Alternatively, if it is unnecessary to search under the `o=sample` base DN, then separate subtree views can be created in the same client connection policy. For example, one subtree view

would be created for **ou=east,o=sample** and one for **ou=west,o=sample**. Each subtree view is then associated with its own proxying request processor, one for **ou=east** requests and one for **ou=west** requests.

## An Example of a Migrated Sample Customer Entry

Following is an example of a Sample customer entry that has been migrated to the Example.com database. The user entry is defined in the Example.com directory server's database as follows. The attributes that have undergone a proxy transformation are marked in bold. Note that this view is how the entry appears to search requests under the **dc=example,dc=com** base DN.

```
dn: uid=scase,ou=People,dc=example,dc=com
objectClass: person
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: exampleAccount
objectClass: top
description: A customer account migrated from Sample merger
uid: scase
exAccountNumber: 234098
exSampleRegion: East
exSampleLinkedAccounts: uid=jcase,ou=people,dc=example,dc=com
userPassword: password
givenName: Sterling
cn: Sterling Case
sn: Case
telephoneNumber: +1 804 094 3356
street: 00468 Second Street
l: Arlington
mail: sterlingcase@maildomain.com
st: VA
```

Following is what the directory proxy server returns to LDAP clients who have requested the entry when searching under the **o=sample** base DN. Note that the DN returned includes **ou=east**, even though this branch does not exist in the Example.com DIT. It also returns the attribute names as they are defined in the Sample schema.

```
dn: sampleID=scase,ou=east,o=sample
objectClass: person
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: exampleAccount
objectClass: top
description: A customer account migrated from Sample merger
sampleID: scase
exAccountNumber: 234098
sampleRegion: East
sampleLinkedAccounts: sampleID=jcase,ou=people,ou=east,o=sample
userPassword: password
givenName: Sterling
cn: Sterling Case
sn: Case
```

```
telephoneNumber: +1 804 094 3356
street: 00468 Second Street
l: Arlington
mail: sterlingcase@maildomain.com
st: VA
```

## Overview of Deployment Steps

In this deployment scenario, we will take the following steps:

- Install the updated schema file on the directory proxy server.
- Create three attribute mapping proxy transformations and three DN mapping proxy transformations
- Create a new proxying request processor, using the existing `dc_example_dc_com` request processor as a template.
- Assign the six proxy transformations to the new proxying request processor.
- Create a new subtree view for `o=sample` that references the new proxying request processor.
- Add the new subtree view to the existing client connection policy.
- Test our configuration by performing some searches on the Sample DIT.

## Installing the Schema File

First, we stop the proxy and install a schema file so that the directory proxy server schema matches that of the LDAP external servers. The schema file, `named 97-proxy-deploy-guide.ldif`, contains the following information:

```
dn: cn=schema
objectClass: top
objectClass: ldapSubentry
objectClass: subschema
cn: schema
attributeTypes: ( 1.3.6.1.4.1.32473.2.1.1
  NAME 'exAccountNumber'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
  SINGLE-VALUE )
attributeTypes: ( 1.3.6.1.4.1.32473.1.1.3
  NAME 'sampleLinkedAccounts'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 )
attributeTypes: ( 1.3.6.1.4.1.32473.1.1.2
  NAME 'sampleRegion'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
  SINGLE-VALUE )
attributeTypes: ( 1.3.6.1.4.1.32473.1.1.1
  NAME 'sampleID'
```

```
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE )
attributeTypes: ( 1.3.6.1.4.1.32473.2.1.3
  NAME 'exSampleLinkedAccounts'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 )
attributeTypes: ( 1.3.6.1.4.1.32473.2.1.2
  NAME 'exSampleRegion'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
  SINGLE-VALUE )
objectClasses: ( 1.3.6.1.4.1.32473.2.2.1
  NAME 'exampleAccount'
  SUP top
  AUXILIARY
  MAY ( exAccountNumber $
        exSampleRegion $
        exSampleLinkedAccounts $
        sampleID $
        sampleRegion $
        sampleLinkedAccounts ) )
```

The schema file defines some Example.com schema, such as **exAccountNumber** and **exSampleRegion**, and some Sample schema, such as **sampleRegion** and **sampleID**.

In the main installation directory, **UnboundID-Proxy**, we run the **stop-proxy** command to stop the directory proxy server instance.

```
root@proxy-east-01:bin/stop-proxy
```

Next, we copy the schema into the **config/schema** directory and restart the directory proxy server.

```
root@proxy-east-01:cp /path-to/97-proxy-deploy-guide.ldif \
config/schema/
root@proxy-east-01: bin/start-proxy
```

We could also apply the new schema file dynamically without stopping and restarting the server by copying the schema into the **config/schema** directory and then adding a schema file task entry as follows:

```
dn: ds-task-id=Add 97-proxy-deploy-guide.ldif,cn=Scheduled Tasks,cn=tasks
changetype: add
objectClass: top
objectClass: ds-task
objectClass: ds-task-add-schema-file
ds-task-id: Add 97-proxy-deploy-guide.ldif
ds-task-class-name: com.unboundid.directory.server.tasks.AddSchemaFileTask
ds-task-schema-file-name: 97-proxy-deploy-guide.ldif
```

## Creating and Configuring the Proxy Transformations

Next, we create three attribute mapping proxy transformations and three DN mapping proxy transformations. We run the `dsconfig` tool, which is located in the `bin` or `bat` directory of the server root directory, `UnboundID-Proxy`.

```
root@proxy-east-01:~/dsconfig
>>>> Specify LDAP connection parameters

Directory Proxy Server hostname or IP address [localhost]:

How do you want to connect to the Directory Proxy Server at localhost?

    1)  LDAP
    2)  LDAP with SSL
    3)  LDAP with StartTLS

Enter choice [1]: 1

Directory Proxy Server at localhost port number [389]:
Administrator user bind DN [cn=Directory Manager]:
Password for user 'cn=Directory Manager':
```

We select “Proxy Transformations” in the main configuration console menu to configure proxy transformations.

```
>>>> UnboundID Directory Proxy Server configuration console
main menu

What do you want to configure?

    1)  Client Connection Policy   10)  Log Retention Policy
    2)  Connection Handler         11)  Log Rotation Policy
    3)  External Server            12)  Placement Algorithm
    4)  Global Attribute Index     13)  Proxy Transformation
    5)  Global Configuration       14)  Request Processor
    6)  LDAP Health Check          15)  Server Affinity Provider
    7)  Load Balancing Algorithm  16)  Subtree View
    8)  Location                   17)  Work Queue
    9)  Log Publisher

    o)  'Basic' objects are shown - change this
    q)  quit

Enter choice: 13
```

## Creating the Attribute Mapping Proxy Transformations

We enter 2 to create a new proxy transformation.

```
>>>> Proxy Transformation management menu2

What would you like to do?

    1)  List existing Proxy Transformations
    2)  Create a new Proxy Transformation
```

- 3) View and edit an existing Proxy Transformation
- 4) Delete an existing Proxy Transformation
- b) back
- q) quit

Enter choice [b]: 2

We are going to start by creating a mapping from the `sampleRegion` attribute to the `exSampleRegion` attribute, so we select 1 to create a new attribute mapping proxy transformation.

>>>> Select the type of Proxy Transformation that you want to create:

- 1) Attribute Mapping Proxy Transformation
- 2) DN Mapping Proxy Transformation
- ? ) help
- c) cancel
- q) quit

Enter choice [c]: 1

We give the new proxy transformation a name that illustrates the attribute mapping that it performs.

>>>> Enter a name for the Attribute Mapping Proxy Transformation that you want to create:  
`sampleRegion-to-exSampleRegion`

We enter 1 to enable the proxy transformation by default.

>>>> Configuring the 'enabled' property

Indicates whether this proxy transformation is enabled for use in the server.

Select a value for the 'enabled' property:

- 1) true
- 2) false
- ? ) help
- c) cancel
- q) quit

Enter choice [c]: 1

We provide the name of the source attribute in the Sample schema that we are mapping to the Example.com schema, which is `sampleRegion`.

>>>> Configuring the 'source-attribute' property

Specifies the name of the attribute that may appear in client requests which should be remapped to the target attribute. Note that the source attribute must not be equal to the target attribute.

Syntax: `STRING`

```
Enter a value for the 'source-attribute' property:
sampleRegion
```

Next, we provide the name of the target attribute in the Example.com schema, which is `exSampleRegion`.

```
>>>> Configuring the 'target-attribute' property
```

```
Specifies the name of the attribute to which the source
attribute should be mapped. Note that the target
attribute must not be equal to the source attribute.
```

```
Syntax:  STRING
```

```
Enter a value for the 'target-attribute' property:
exSampleRegion
```

We review the configuration properties, and then enter `f` to create the new attribute mapping proxy transformation.

```
>>>> Configure the properties of the Attribute Mapping Proxy
Transformation
```

|    | Property                                                        | Value(s)                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|----|-----------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | -----                                                           | -----                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 1) | description                                                     | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 2) | enabled                                                         | true                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 3) | source-attribute                                                | sampleRegion                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 4) | target-attribute                                                | exSampleRegion                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 5) | enable-dn-mapping                                               | true                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 6) | enable-control-mapping                                          | true                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 7) | map-control                                                     | assertion-request,<br>authorization-identity-response,<br>entry-change-notification,<br>get-authorization-entry-request,<br>get-authorization-entry-response,<br>interactive-transaction-<br>specification-response,<br>intermediate-client-request,<br>matched-values-request,<br>post-read-request,<br>post-read-response,<br>pre-read-request,<br>pre-read-response,<br>proxied-authorization-v1-request,<br>proxied-authorization-v2-request,<br>sort-request,sort-response |
|    |                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| ?) | help                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| f) | finish - create the new Attribute Mapping Proxy Transformation  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| d) | display the equivalent dsconfig arguments to create this object |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| b) | back                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| q) | quit                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

```
Enter choice [b]: f
```

```
The Attribute Mapping Proxy Transformation was created successfully
```

We repeat the previous steps to create another attribute mapping proxy transformation, this time to map between the Sample Corporation's `sampleID` attribute and the Example.com `uid` attribute.

```
>>>> Proxy Transformation management menu

What would you like to do?

1) List existing Proxy Transformations
2) Create a new Proxy Transformation
3) View and edit an existing Proxy Transformation
4) Delete an existing Proxy Transformation

b) back
q) quit

Enter choice [b]: 2
```

We use the proxy transformation that we created in the previous steps as a template for the new proxy transformation.

```
>>>> Select an existing Proxy Transformation to use as a template for the new Proxy Transformation configuration or 'n' to create one from scratch:

1) sampleRegion-to-exSampleRegion
n) new Proxy Transformation created from scratch
c) cancel
q) quit

Enter choice [n]: 1
```

We name the new attribute mapping transformation `sampleID-to-uid`.

```
>>>> Enter a name for the Attribute Mapping Proxy Transformation that you want

to create: sampleID-to-uid
```

We update the properties for the new proxy transformation. First, we change the source-attribute property by selecting 3.

```
>>>> Configure the properties of the Attribute Mapping Proxy Transformation
```

|    | Property               | Value(s)           |
|----|------------------------|--------------------|
| 1) | description            | -                  |
| 2) | enabled                | true               |
| 3) | source-attribute       | sampleRegion       |
| 4) | target-attribute       | exSampleRegion     |
| 5) | enable-dn-mapping      | true               |
| 6) | enable-control-mapping | true               |
| 7) | map-control            | assertion-request, |



```
authorization-identity-response,
entry-change-notification,
get-authorization-entry-request,
get-authorization-entry-response,
interactive-transaction-
specification-response,
intermediate-client-request,
matched-values-request,
post-read-request,
post-read-response,
pre-read-request,
pre-read-response,
proxied-authorization-v1-request,
proxied-authorization-v2-request,
sort-request,sort-response

?) help
f) finish - create the new Attribute Mapping Proxy
Transformation
d) display the equivalent dsconfig arguments to create this
object
b) back
q) quit

Enter choice [b]: 3
```

We select 2 to enter a new value for this property and then enter the new source-attribute name, `sampleID`.

```
>>>> Configuring the 'source-attribute' property

Specifies the name of the attribute that may appear in
client requests which should be remapped to the target
attribute. Note that the source attribute must not be
equal to the target attribute.

Syntax:  STRING

Do you want to modify the 'source-attribute' property?

1) Keep the value: sampleRegion
2) Change the value

?) help
q) quit

Enter choice [1]: 2

Enter a value for the 'source-attribute' property: sampleID
```

Next, we update the `target-attribute` property.

```
>>>> Configure the properties of the Attribute Mapping Proxy
Transformation
```

|    | Property          | Value(s)       |
|----|-------------------|----------------|
| 1) | description       | -              |
| 2) | enabled           | true           |
| 3) | source-attribute  | sampleID       |
| 4) | target-attribute  | exSampleRegion |
| 5) | enable-dn-mapping | true           |

```
6) enable-control-mapping true
7) map-control               assertion-request,
                             authorization-identity-response,
                             entry-change-notification,
                             get-authorization-entry-request,
                             get-authorization-entry-response,
                             interactive-transaction-
                             specification-response,
                             intermediate-client-request,
                             matched-values-request,
                             post-read-request,
                             post-read-response,
                             pre-read-request,
                             pre-read-response,
                             proxied-authorization-v1-request,
                             proxied-authorization-v2-request,
                             sort-request,sort-response

?) help
f) finish - create the new Attribute Mapping Proxy
  Transformation
d) display the equivalent dsconfig arguments to create this
  object
b) back
q) quit
```

Enter choice [b]: 4

We change the value of the `target-attribute` property to `uid`.

>>>> Configuring the 'target-attribute' property

Specifies the name of the attribute to which the source attribute should be mapped. Note that the target attribute must not be equal to the source attribute.

Syntax: STRING

Do you want to modify the 'target-attribute' property?

```
1) Keep the value: exSampleRegion
2) Change the value

?) help
q) quit
```

Enter choice [1]: 2

Enter a value for the 'target-attribute' property: uid

We review the properties of the new attribute mapping proxy transformation and then enter `f` to save our changes.

>>>> Configure the properties of the Attribute Mapping Proxy Transformation

| Property | Value(s) |
|----------|----------|
| -----    |          |

```
1) description          -
2) enabled              true
3) source-attribute     sampleID
4) target-attribute     uid
5) enable-dn-mapping    true
6) enable-control-mapping true
7) map-control          assertion-request,
                        authorization-identity-response,
                        entry-change-notification,
                        get-authorization-entry-request,
                        get-authorization-entry-response,
                        interactive-transaction-
                        specification-response,
                        intermediate-client-request,
                        matched-values-request,
                        post-read-request,
                        post-read-response,
                        pre-read-request,
                        pre-read-response,
                        proxied-authorization-v1-request,
                        proxied-authorization-v2-request,
                        sort-request,sort-response

?) help
f) finish - create the new Attribute Mapping Proxy
  Transformation
d) display the equivalent dsconfig arguments to create this
  object
b) back
q) quit
```

Enter choice [b]: f

The Attribute Mapping Proxy Transformation was created successfully

We repeat the previous steps again to create a last attribute mapping proxy transformation, mapping between the Sample `sampleLinkedAccounts` attribute and the Example.com `exSampleLinkedAccounts` attribute.

```
>>>> Proxy Transformation management menu
```

What would you like to do?

```
1) List existing Proxy Transformations
2) Create a new Proxy Transformation
3) View and edit an existing Proxy Transformation
4) Delete an existing Proxy Transformation

b) back
q) quit
```

Enter choice [b]: 2

We use the proxy transformation created in the previous steps as a template for this new proxy transformation.

```
>>>> Select an existing Proxy Transformation to use as a
template for the new Proxy Transformation configuration or 'n'
```

```
to
create one from scratch:

    1) sampleID-to-uid
    2) sampleRegion-to-exSampleRegion

    n) new Proxy Transformation created from scratch
    c) cancel
    q) quit

Enter choice [n]: 1
```

We name the new attribute mapping proxy transformation `sampleLinkedAccounts-to-exSampleLinkedAccounts`.

```
>>>> Enter a name for the Attribute Mapping Proxy
Transformation that you want to create:
sampleLinkedAccounts-to-exSampleLinkedAccounts
```

As in the previous procedure, we update the properties for the new proxy transformation. First, we change the `source-attribute` property by selecting 3.

```
>>>> Configure the properties of the Attribute Mapping Proxy
Transformation

    Property                                Value(s)
    -----
    1) description                          -
    2) enabled                              true
    3) source-attribute                      sampleID
    4) target-attribute                     uid
    5) enable-dn-mapping                    true
    6) enable-control-mapping               true
    7) map-control                          assertion-request,
   authorization-identity-response,
   entry-change-notification,
   get-authorization-entry-request,
   get-authorization-entry-response,
   interactive-transaction-
   specification-response,
   intermediate-client-request,
   matched-values-request,
   post-read-request,
   post-read-response,
   pre-read-request,
   pre-read-response,
   proxied-authorization-v1-request,
   proxied-authorization-v2-request,
   sort-request,sort-response

    ?) help
    f) finish - create the new Attribute Mapping Proxy
       Transformation
    d) display the equivalent dsconfig arguments to create this
       object
    b) back
    q) quit
```

```
Enter choice [b]: 3
```

We select 2 to enter a new value for this property and then enter the new source-attribute name, `sampleID`.

```
>>>> Configuring the 'source-attribute' property
```

```
Specifies the name of the attribute that may appear in
client requests which should be remapped to the target
attribute. Note that the source attribute must not be
equal to the target attribute.
```

```
Syntax:  STRING
```

```
Do you want to modify the 'source-attribute' property?
```

- 1) Keep the value: `sampleID`
- 2) Change the value
- ?) help
- q) quit

```
Enter choice [1]: 2
```

```
Enter a value for the 'source-attribute' property:
sampleLinkedAccounts
```

Next, we update the `target-attribute` property.

```
>>>> Configure the properties of the Attribute Mapping Proxy
Transformation
```

|    | Property                                                       | Value(s)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|----|----------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | -----                                                          | -----                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 1) | <code>description</code>                                       | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 2) | <code>enabled</code>                                           | <code>true</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 3) | <code>source-attribute</code>                                  | <code>sampleLinked Accounts</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 4) | <code>target-attribute</code>                                  | <code>uid</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 5) | <code>enable-dn-mapping</code>                                 | <code>true</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 6) | <code>enable-control-mapping</code>                            | <code>true</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 7) | <code>map-control</code>                                       | <code>assertion-request,</code><br><code>authorization-identity-response,</code><br><code>entry-change-notification,</code><br><code>get-authorization-entry-request,</code><br><code>get-authorization-entry-response,</code><br><code>interactive-transaction-</code><br><code>specification-response,</code><br><code>intermediate-client-request,</code><br><code>matched-values-request,</code><br><code>post-read-request,</code><br><code>post-read-response,</code><br><code>pre-read-request,</code><br><code>pre-read-response,</code><br><code>proxied-authorization-v1-request,</code><br><code>proxied-authorization-v2-request,</code><br><code>sort-request,sort-response</code> |
| ?) | help                                                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| f) | finish - create the new Attribute Mapping Proxy Transformation |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| d) | display the equivalent dsconfig arguments to create this       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

```
        object
    b) back
    q) quit
Enter choice [b]: 4
```

We change the value of the `target-attribute` property to `exSampleLinkedAccounts`.

```
>>>> Configuring the 'target-attribute' property
```

Specifies the name of the attribute to which the source attribute should be mapped. Note that the target attribute must not be equal to the source attribute.

Syntax: STRING

Do you want to modify the 'target-attribute' property?

- 1) Keep the value: `exSampleRegion`
- 2) Change the value
- ?) help
- q) quit

```
Enter choice [1]: 2
```

Enter a value for the 'target-attribute' property:  
`exSampleLinkedAccounts`

We review the properties of the new attribute mapping proxy transformation and then enter `f` to save our changes.

```
>>>> Configure the properties of the Attribute Mapping Proxy
Transformation
```

|    | Property                            | Value(s)                                                                                                                                                                                                                                                                                                                                                                                                     |
|----|-------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | -----                               | -----                                                                                                                                                                                                                                                                                                                                                                                                        |
| 1) | <code>description</code>            | -                                                                                                                                                                                                                                                                                                                                                                                                            |
| 2) | <code>enabled</code>                | true                                                                                                                                                                                                                                                                                                                                                                                                         |
| 3) | <code>source-attribute</code>       | <code>sampleLinkedAccounts</code>                                                                                                                                                                                                                                                                                                                                                                            |
| 4) | <code>target-attribute</code>       | <code>exSampleLinkedAccounts</code>                                                                                                                                                                                                                                                                                                                                                                          |
| 5) | <code>enable-dn-mapping</code>      | true                                                                                                                                                                                                                                                                                                                                                                                                         |
| 6) | <code>enable-control-mapping</code> | true                                                                                                                                                                                                                                                                                                                                                                                                         |
| 7) | <code>map-control</code>            | assertion-request,<br>authorization-identity-response,<br>entry-change-notification,<br>get-authorization-entry-request,<br>get-authorization-entry-response,<br>interactive-transaction-<br>specification-response,<br>intermediate-client-request,<br>matched-values-request,<br>post-read-request,<br>post-read-response,<br>pre-read-request,<br>pre-read-response,<br>proxied-authorization-v1-request, |

```
proxied-authorization-v2-request,
sort-request,sort-response

?) help
f) finish - create the new Attribute Mapping Proxy
Transformation
d) display the equivalent dsconfig arguments to create this
object
b) back
q) quit
```

Enter choice [b]: f

The Attribute Mapping Proxy Transformation was created successfully

## Creating the DN Mapping Proxy Transformations

Now we create the DN mapping proxy transformations. We select 2 to create a new proxy transformation.

```
>>>> Proxy Transformation management menu
```

What would you like to do?

- 1) List existing Proxy Transformations
- 2) Create a new Proxy Transformation
- 3) View and edit an existing Proxy Transformation
- 4) Delete an existing Proxy Transformation
- b) back
- q) quit

Enter choice [b]: 2

We enter n to create a new proxy transformation from scratch.

```
>>>> Select an existing Proxy Transformation to use as a
template for the new Proxy Transformation configuration or 'n'
to
create one from scratch:
```

- 1) sampleID-to-uid
- 2) sampleLinkedAccounts-to-exSampleLinkedAccounts
- 3) sampleRegion-to-exSampleRegion
- n) new Proxy Transformation created from scratch
- c) cancel
- q) quit

Enter choice [n]: n

We select 2 to create a new DN mapping proxy transformation.

```
>>>> Select the type of Proxy Transformation that you want to
create:
```

- 1) Attribute Mapping Proxy Transformation
- 2) DN Mapping Proxy Transformation
- ? ) help
- c) cancel
- q) quit

Enter choice [c]: 2

This transformation maps `ou=east,o=sample` in the Sample schema `dc=example,dc=com` in the Example.com schema. We name it `sample_east-to-example`.

```
>>>> Enter a name for the DN Mapping Proxy Transformation that
you want to create: sample_east-to-example
```

```
>>>> Configuring the 'enabled' property
```

Indicates whether this proxy transformation is enabled  
for use in the server.

We select 1 to enable the transformation by default.

Select a value for the 'enabled' property:

- 1) true
- 2) false
- ? ) help
- c) cancel
- q) quit

Enter choice [c]: 1

Next, we specify the source DN as it appears in client requests.

```
>>>> Configuring the 'source-dn' property
```

Specifies the source DN that may appear in client  
requests which should be remapped to the target DN. Note  
that the source DN must not be equal to the target DN.

Syntax: DN

Enter a value for the 'source-dn' property:

`ou=east,o=sample`

We specify the target DN, where requests for the source DN should be routed.

```
>>>> Configuring the 'target-dn' property
```

Specifies the DN to which the source DN should be mapped.  
Note that the target DN must not be equal to the source  
DN.

Syntax: DN

Enter a value for the 'target-dn' property:

`dc=example,dc=com`



We review the configuration properties, and then enter **f** to create the new DN mapping proxy transformation.

```
>>>> Configure the properties of the DN Mapping Proxy Transformation
```

|    | Property                                                        | Value(s)                                                                                                                                                                                                                                                                                                                                                  |
|----|-----------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | -----                                                           | -----                                                                                                                                                                                                                                                                                                                                                     |
| 1) | description                                                     | -                                                                                                                                                                                                                                                                                                                                                         |
| 2) | enabled                                                         | true                                                                                                                                                                                                                                                                                                                                                      |
| 3) | source-dn                                                       | "ou=east,o=sample"                                                                                                                                                                                                                                                                                                                                        |
| 4) | target-dn                                                       | "dc=example,dc=com"                                                                                                                                                                                                                                                                                                                                       |
| 5) | enable-attribute-mapping                                        | true                                                                                                                                                                                                                                                                                                                                                      |
| 6) | map-attribute                                                   | If no specific map attributes are defined but attribute mapping is enabled, then all attributes with a distinguished name or name and optional UID syntax will be examined to determine if any mapping is required.                                                                                                                                       |
| 7) | enable-control-mapping                                          | true                                                                                                                                                                                                                                                                                                                                                      |
| 8) | map-control                                                     | assertion-request,<br>authorization-identity-response,<br>entry-change-notification,<br>get-effective-rights-request,<br>interactive-transaction-specification-response,<br>intermediate-client-request,<br>matched-values-request,<br>post-read-response,<br>pre-read-response,<br>proxied-authorization-v1-request,<br>proxied-authorization-v2-request |
|    |                                                                 |                                                                                                                                                                                                                                                                                                                                                           |
| ?) | help                                                            |                                                                                                                                                                                                                                                                                                                                                           |
| f) | finish - create the new DN Mapping Proxy Transformation         |                                                                                                                                                                                                                                                                                                                                                           |
| d) | display the equivalent dsconfig arguments to create this object |                                                                                                                                                                                                                                                                                                                                                           |
| b) | back                                                            |                                                                                                                                                                                                                                                                                                                                                           |
| q) | quit                                                            |                                                                                                                                                                                                                                                                                                                                                           |

```
Enter choice [b]: f
```

```
The DN Mapping Proxy Transformation was created successfully
```

Next, we create a new DN mapping proxy transformation that maps **ou=west,o=sample** in the Sample schema to **dc=example,dc=com** in the Example.com schema. We name it **Sample\_west-to-example**.

```
>>>> Proxy Transformation management menu
```

```
What would you like to do?
```

- 1) List existing Proxy Transformations
- 2) Create a new Proxy Transformation
- 3) View and edit an existing Proxy Transformation
- 4) Delete an existing Proxy Transformation

b) back  
q) quit

Enter choice [b]: 2

We select 4 to use the DN mapping proxy transformation that we created in the previous step as a template.

```
>>>> Select an existing Proxy Transformation to use as a
template for the new
```

Proxy Transformation configuration or 'n' to create one from scratch:

- 1) sampleID-to-uid
- 2) sampleLinkedAccounts-to-exSampleLinkedAccounts
- 3) sampleRegion-to-exSampleRegion
- 4) sample\_east-to-example
- n) new Proxy Transformation created from scratch
- c) cancel
- q) quit

Enter choice [n]: 4

We name the new proxy transformation `sample_west-to-example`.

```
>>>> Enter a name for the DN Mapping Proxy Transformation that
you want to create: sample_west-to-example
```

Next, we configure the properties of the new DN mapping proxy transformation. First, we enter 3 to configure the `source-dn` property.

```
>>>> Configure the properties of the DN Mapping Proxy Transformation
```

|       | Property                 | Value(s)                                                                                                                                                                                                                                                                                                                                                  |
|-------|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ----- |                          |                                                                                                                                                                                                                                                                                                                                                           |
| 1)    | description              | -                                                                                                                                                                                                                                                                                                                                                         |
| 2)    | enabled                  | true                                                                                                                                                                                                                                                                                                                                                      |
| 3)    | source-dn                | "ou=east,o=sample"                                                                                                                                                                                                                                                                                                                                        |
| 4)    | target-dn                | "dc=example,dc=com"                                                                                                                                                                                                                                                                                                                                       |
| 5)    | enable-attribute-mapping | true                                                                                                                                                                                                                                                                                                                                                      |
| 6)    | map-attribute            | If no specific map attributes are defined but attribute mapping is enabled, then all attributes with a distinguished name or name and optional UID syntax will be examined to determine if any mapping is required.                                                                                                                                       |
| 7)    | enable-control-mapping   | true                                                                                                                                                                                                                                                                                                                                                      |
| 8)    | map-control              | assertion-request,<br>authorization-identity-response,<br>entry-change-notification,<br>get-effective-rights-request,<br>interactive-transaction-specification-response,<br>intermediate-client-request,<br>matched-values-request,<br>post-read-response,<br>pre-read-response,<br>proxied-authorization-v1-request,<br>proxied-authorization-v2-request |

- ? ) help
- f ) finish - create the new DN Mapping Proxy Transformation
- d ) display the equivalent dsconfig arguments to create this object
- b ) back
- q ) quit

Enter choice [b]: 3

We change the source DN to ou=west,o=sample.

>>>> Configuring the 'source-dn' property

Specifies the source DN that may appear in client requests which should be remapped to the target DN. Note that the source DN must not be equal to the target DN.

Syntax: DN

Do you want to modify the 'source-dn' property?

- 1) Keep the value: "ou=east,o=sample"
- 2) Change the value
- ? ) help
- q ) quit

Enter choice [1]: 2

Enter a value for the 'source-dn' property:

ou=west,o=sample

We keep the **target-dn** value the same as before. After reviewing the configuration properties, we enter f to save our changes.

>>>> Configure the properties of the DN Mapping Proxy Transformation

| Property                    | Value(s)                                                                                                                                                                                                                                                                         |
|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -----                       |                                                                                                                                                                                                                                                                                  |
| 1) description              | -                                                                                                                                                                                                                                                                                |
| 2) enabled                  | true                                                                                                                                                                                                                                                                             |
| 3) source-dn                | "ou=west,o=sample"                                                                                                                                                                                                                                                               |
| 4) target-dn                | "dc=example,dc=com"                                                                                                                                                                                                                                                              |
| 5) enable-attribute-mapping | true                                                                                                                                                                                                                                                                             |
| 6) map-attribute            | If no specific map attributes are defined but attribute mapping is enabled, then all attributes with a distinguished name or name and optional UID syntax will be examined to determine if any mapping is required.                                                              |
| 7) enable-control-mapping   | true                                                                                                                                                                                                                                                                             |
| 8) map-control              | assertion-request,<br>authorization-identity-response,<br>entry-change-notification,<br>get-effective-rights-request,<br>interactive-transaction-specification-response,<br>intermediate-client-request,<br>matched-values-request,<br>post-read-response,<br>pre-read-response, |

```
proxied-authorization-v1-request,
proxied-authorization-v2-request

?) help
f) finish - create the new DN Mapping Proxy Transformation
d) display the equivalent dsconfig arguments to create this
   object
b) back
q) quit

Enter choice [b]: f

The DN Mapping Proxy Transformation was created successfully
```

Finally, we create a DN mapping proxy transformation for the base DN of the Sample database.

```
>>>> Proxy Transformation management menu

What would you like to do?

1) List existing Proxy Transformations
2) Create a new Proxy Transformation
3) View and edit an existing Proxy Transformation
4) Delete an existing Proxy Transformation

b) back
q) quit

Enter choice [b]: 2
```

We use one of the previous DN mapping proxy transformations as a template for the new one.

```
>>>> Select an existing Proxy Transformation to use as a template
for the new Proxy Transformation configuration or 'n' to
create one from scratch:

1) sampleID-to-uid
2) sampleLinkedAccounts-to-exSampleLinkedAccounts
3) sampleRegion-to-exSampleRegion
4) sample_east-to-example
5) sample_west-to-example

n) new Proxy Transformation created from scratch
c) cancel
q) quit

Enter choice [n]: 4
```

As this proxy transformation maps `o=sample` to `dc=example,dc=com`, we name it `sample-to-example`.

```
>>>> Enter a name for the DN Mapping Proxy Transformation that
you want to create: sample-to-example
```

Next, we configure the `source-dn` property of the new DN mapping proxy transformation.

```
>>>> Configure the properties of the DN Mapping Proxy Transformation
```

|    | Property                 | Value(s)                                                                                                                                                                                                                                                                                                                                                  |
|----|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1) | description              | -                                                                                                                                                                                                                                                                                                                                                         |
| 2) | enabled                  | true                                                                                                                                                                                                                                                                                                                                                      |
| 3) | source-dn                | "ou=east,o=sample"                                                                                                                                                                                                                                                                                                                                        |
| 4) | target-dn                | "dc=example,dc=com"                                                                                                                                                                                                                                                                                                                                       |
| 5) | enable-attribute-mapping | true                                                                                                                                                                                                                                                                                                                                                      |
| 6) | map-attribute            | If no specific map attributes are defined but attribute mapping is enabled, then all attributes with a distinguished name or name and optional UID syntax will be examined to determine if any mapping is required.                                                                                                                                       |
| 7) | enable-control-mapping   | true                                                                                                                                                                                                                                                                                                                                                      |
| 8) | map-control              | assertion-request,<br>authorization-identity-response,<br>entry-change-notification,<br>get-effective-rights-request,<br>interactive-transaction-specification-response,<br>intermediate-client-request,<br>matched-values-request,<br>post-read-response,<br>pre-read-response,<br>proxied-authorization-v1-request,<br>proxied-authorization-v2-request |
| ?) | help                     |                                                                                                                                                                                                                                                                                                                                                           |
| f) | finish                   | - create the new DN Mapping Proxy Transformation                                                                                                                                                                                                                                                                                                          |
| d) | display                  | the equivalent dsconfig arguments to create this object                                                                                                                                                                                                                                                                                                   |
| b) | back                     |                                                                                                                                                                                                                                                                                                                                                           |
| q) | quit                     |                                                                                                                                                                                                                                                                                                                                                           |

Enter choice [b]: 3

We set the source DN to the Sample base DN, o=sample.

>>>> Configuring the 'source-dn' property

Specifies the source DN that may appear in client requests which should be remapped to the target DN. Note that the source DN must not be equal to the target DN.

Syntax: DN

Do you want to modify the 'source-dn' property?

- 1) Keep the value: "ou=east,o=sample"
- 2) Change the value
- ?) help
- q) quit

Enter choice [1]: 2

Enter a value for the 'source-dn' property: o=sample

We review the configuration changes and then enter f to save our changes.

>>>> Configure the properties of the DN Mapping Proxy Transformation

|    | Property                 | Value(s)                                                                                                                                                                                                                                                                                                                                                  |
|----|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1) | description              | -                                                                                                                                                                                                                                                                                                                                                         |
| 2) | enabled                  | true                                                                                                                                                                                                                                                                                                                                                      |
| 3) | source-dn                | "o=sample"                                                                                                                                                                                                                                                                                                                                                |
| 4) | target-dn                | "dc=example,dc=com"                                                                                                                                                                                                                                                                                                                                       |
| 5) | enable-attribute-mapping | true                                                                                                                                                                                                                                                                                                                                                      |
| 6) | map-attribute            | If no specific map attributes are defined but attribute mapping is enabled, then all attributes with a distinguished name or name and optional UID syntax will be examined to determine if any mapping is required.                                                                                                                                       |
| 7) | enable-control-mapping   | true                                                                                                                                                                                                                                                                                                                                                      |
| 8) | map-control              | assertion-request,<br>authorization-identity-response,<br>entry-change-notification,<br>get-effective-rights-request,<br>interactive-transaction-specification-response,<br>intermediate-client-request,<br>matched-values-request,<br>post-read-response,<br>pre-read-response,<br>proxied-authorization-v1-request,<br>proxied-authorization-v2-request |
| ?  | help                     |                                                                                                                                                                                                                                                                                                                                                           |
| f) | finish                   | - create the new DN Mapping Proxy Transformation                                                                                                                                                                                                                                                                                                          |
| d) | display                  | the equivalent dsconfig arguments to create this object                                                                                                                                                                                                                                                                                                   |
| b) | back                     |                                                                                                                                                                                                                                                                                                                                                           |
| q) | quit                     |                                                                                                                                                                                                                                                                                                                                                           |

Enter choice [b]: f

The DN Mapping Proxy Transformation was created successfully

>>>> Proxy Transformation management menu

What would you like to do?

- 1) List existing Proxy Transformations
- 2) Create a new Proxy Transformation
- 3) View and edit an existing Proxy Transformation
- 4) Delete an existing Proxy Transformation

- b) back
- q) quit

Enter choice [b]:

## Creating a Request Processor to Manage the Proxy Transformations

Next, we need to create a new proxying request processor that includes our new attribute and DN mapping proxy transformations. We will use the existing `dc_example_dc_com` request processor as a template.

From the main configuration console menu, we select “Request Processor” to create a new request processor.

```
>>>> UnboundID Directory Proxy Server configuration console
main menu

What do you want to configure?

    1) Client Connection Policy    10) Log Retention Policy
    2) Connection Handler         11) Log Rotation Policy
    3) External Server            12) Placement Algorithm
    4) Global Attribute Index     13) Proxy Transformation
    5) Global Configuration       14) Request Processor
    6) LDAP Health Check         15) Server Affinity Provider
    7) Load Balancing Algorithm  16) Subtree View
    8) Location                  17) Work Queue
    9) Log Publisher

    o) 'Basic' objects are shown - change this
    q) quit

Enter choice: 14
```

We enter 2 to create a new request processors.

```
>>>> Request Processor management menu

What would you like to do?

    1) List existing Request Processors
    2) Create a new Request Processor
    3) View and edit an existing Request Processor
    4) Delete an existing Request Processor

    b) back
    q) quit

Enter choice [b]: 2
```

We enter 1 to use our current request processor as a template.

```
>>>> Select an existing Request Processor to use as a template
for the new Request Processor configuration or 'n' to create
one from scratch:

    1) dc_example_dc_com-req-processor

    n) new Request Processor created from scratch
    c) cancel
    q) quit

Enter choice [n]: 1
```

We provide a name for the new proxying request processor, `o_sample-req-processor`.

```
>>>> Enter a name for the Proxying Request Processor that you
want to create: o_sample-req-processor
```

We review the properties. The load-balancing algorithm is the same as for the previous request processor, though the **transformation** property must be changed. We enter 5 to modify this property.

```
>>>> Configure the properties of the Proxying Request Processor

      Property                                Value(s)
      -----
1)  description                               -
2)  enabled                                  true
3)  allowed-operation                        abandon, add, bind, compare,
      delete, extended, modify,
      modify-dn, search
4)  load-balancing-algorithm                 dc_example_dc_com-round-robin
5)  transformation                           -
6)  referral-behavior                       pass-through
7)  supported-control                       account-usable, assertion,
      authorization-identity,
      get-authorization-entry,
      get-effective-rights,
      ignore-no-user-modification,
      intermediate-client,
      manage-dsa-it, matched-values,
      no-op, password-policy,
      post-read, pre-read,
      proxied-authorization-v1,
      proxied-authorization-v2,
      real-attributes-only,
      retain-identity,
      server-side-sort, subentries,
      subtree-delete,
      virtual-attributes-only
8)  supported-control-oid                     -
?)  help
f)  finish - create the new Proxying Request Processor
d)  display the equivalent dsconfig arguments to create this
    object
b)  back
q)  quit

Enter choice [b]: 5
```

We enter 2 to specify the proxy transformations that we created in the previous sections.

```
>>>> Configuring the 'transformation' property

Specifies the types of transformations that should be
applied to requests and responses processed by this
Proxying Request Processor.

If multiple transformations are provided, then they will
be invoked in the specified order for request
transformations, and in the reverse order for response
transformations.

Do you want to modify the 'transformation' property?

1)  Leave undefined
2)  Add one or more values
```



```
? ) help
q ) quit

Enter choice [1]: 2
```

We select the attribute mapping proxy transformations first, in our case 4, 5, and 6. Next, we select the DN mapping proxy transformations. The order in which we select the DN transformations is important because we have related DNs. We begin with the DNs that are lower in the tree first, and finish with the base DN transformation.

```
Select the Proxy Transformations you wish to add:

1) sample-to-example          5) sampleLinkedAccounts-to-
                               exSampleLinkedAccounts
2) sample_east-to-example     6) sampleRegion-to-
                               exSampleRegion
3) sample_west-to-example     7) Create a new Proxy
                               Transformation
4) sampleID-to-uid            8) Add all Proxy Transformations
? ) help
b ) back
q ) quit

Enter one or more choices separated by commas [b]: 4,5,6,2,3,1
```

We confirm that the proxy transformations are listed in the correct order and enter 1 to accept and use the values.

```
>>>> Configuring the 'transformation' property (Continued)

The 'transformation' property references the following Proxy
Transformations:

1) sampleID-to-uid
2) sampleLinkedAccounts-to-exSampleLinkedAccounts
3) sampleRegion-to-exSampleRegion
4) sample_east-to-example
5) sample_west-to-example
6) sample-to-example

Do you want to modify the 'transformation' property?

1) Use these values
2) Add one or more values
3) Remove one or more values
4) Leave undefined
5) Revert changes

? ) help
q ) quit

Enter choice [1]: 1
```

After reviewing the request processor properties, we enter f to save our changes.

```
>>>> Configure the properties of the Proxying Request Processor
```

|       | Property                 | Value(s)                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-------|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ----- |                          |                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 1)    | description              | -                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 2)    | enabled                  | true                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 3)    | allowed-operation        | abandon, add, bind, compare,<br>delete,extended, modify,<br>modify-dn, search                                                                                                                                                                                                                                                                                                                                                                 |
| 4)    | load-balancing-algorithm | dc_example_dc_com-round-robin                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 5)    | transformation           | sampleID-to-uid,<br>sampleLinkedAccounts-to-<br>exSampleLinkedAccounts,<br>sampleRegion-to-exSampleRegion,<br>sample_east-to-example,<br>sample_west-to-example,<br>sample-to-example                                                                                                                                                                                                                                                         |
| 6)    | referral-behavior        | pass-through                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 7)    | supported-control        | account-usable, assertion,<br>authorization-identity,<br>get-authorization-entry,<br>get-effective-rights,<br>ignore-no-user-modification,<br>intermediate-client,<br>manage-dsa-it, matched-values,<br>no-op, password-policy,<br>post-read, pre-read,<br>proxied-authorization-v1,<br>proxied-authorization-v2,<br>real-attributes-only,<br>retain-identity,<br>server-side-sort, subentries,<br>subtree-delete,<br>virtual-attributes-only |
| 8)    | supported-control-oid    | -                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| ?)    | help                     |                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| f)    | finish                   | - create the new Proxying Request Processor                                                                                                                                                                                                                                                                                                                                                                                                   |
| d)    | display                  | the equivalent dsconfig arguments to create this<br>object                                                                                                                                                                                                                                                                                                                                                                                    |
| b)    | back                     |                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| q)    | quit                     |                                                                                                                                                                                                                                                                                                                                                                                                                                               |

Enter choice [b]: f

The Proxying Request Processor was created successfully

We enter **b** to go back to the main configuration menu.

>>>> Request Processor management menu

What would you like to do?

- 1) List existing Request Processors
- 2) Create a new Request Processor
- 3) View and edit an existing Request Processor
- 4) Delete an existing Request Processor
- b) back
- q) quit

Enter choice [b]:

## Creating a Subtree View

From the main configuration console menu, we select “Subtree View” to create a new subtree view.

```
>>>> UnboundID Directory Proxy Server configuration console
main menu

What do you want to configure?

    1) Client Connection Policy    10) Log Retention Policy
    2) Connection Handler          11) Log Rotation Policy
    3) External Server             12) Placement Algorithm
    4) Global Attribute Index      13) Proxy Transformation
    5) Global Configuration        14) Request Processor
    6) LDAP Health Check          15) Server Affinity Provider
    7) Load Balancing Algorithm    16) Subtree View
    8) Location                   17) Work Queue
    9) Log Publisher

    o) 'Basic' objects are shown - change this
    q) quit

Enter choice: 16
```

We enter 2 to create a new subtree view.

```
>>>> Subtree View management menu

What would you like to do?

    1) List existing Subtree Views
    2) Create a new Subtree View
    3) View and edit an existing Subtree View
    4) Delete an existing Subtree View

    b) back
    q) quit

Enter choice [b]: 2
```

We select 1 to use the existing subtree view as a template.

```
>>>> Select an existing Subtree View to use as a template for
the new Subtree View configuration or 'n' to create one from
scratch:

    1) dc_example_dc_com-view
    n) new Subtree View created from scratch
    c) cancel
    q) quit

Enter choice [n]: 1
```

We name the new subtree view `o_sample-view`.

```
>>>> Enter a name for the Subtree View that you want to create:
o_sample-view
```

We configure the **base-dn** property to be the base DN of the Sample dataset.

```
>>>> Configure the properties of the Subtree View

      Property          Value(s)
      -----
1)  description         -
2)  base-dn             "dc=example,dc=com"
3)  request-processor   dc_example_dc_com-req-processor

?)  help
f)  finish - create the new Subtree View
d)  display the equivalent dsconfig arguments to create this
    object
b)  back
q)  quit

Enter choice [b]: 2

>>>> Configuring the 'base-dn' property

      The DN of the entry that is at the base of the subtree
      for this Subtree View.

      Syntax:  DN

Do you want to modify the 'base-dn' property?

1)  Keep the value: "dc=example,dc=com"
2)  Change the value

?)  help
q)  quit

Enter choice [1]: 2

Enter a value for the 'base-dn' property: o=sample
```

Next, we change the **request-processor** property to reflect the new proxying request processor we created in the previous section, **o\_sample-req-processor**.

```
>>>> Configure the properties of the Subtree View

      Property          Value(s)
      -----
1)  description         -
2)  base-dn             "o=sample"
3)  request-processor   dc_example_dc_com-req-processor

?)  help
f)  finish - create the new Subtree View
d)  display the equivalent dsconfig arguments to create this
    object
b)  back
q)  quit

Enter choice [b]: 3

>>>> Configuring the 'request-processor' property

      The request processor that will be used to process
      requests in portions of the DIT managed by this Subtree
      View.

Do you want to modify the 'request-processor' property?
```

- 1) Keep the value: dc\_example\_dc\_com-req-processor
- 2) Change it to the Request Processor:  
o\_sample-req-processor
- 3) Create a new Request Processor
- ?) help
- q) quit

Enter choice [1]: 2

We review the subtree view configuration and enter **f** to save our changes.

```
>>>> Configure the properties of the Subtree View

Property          Value(s)
-----
1) description     -
2) base-dn         "o=sample"
3) request-processor o_sample-req-processor

?) help
f) finish - create the new Subtree View
d) display the equivalent dsconfig arguments to create this
  object
b) back
q) quit
```

Enter choice [b]: f

The Subtree View was created successfully

We enter **b** to go back to the main configuration menu.

```
>>>> Subtree View management menu

What would you like to do?

1) List existing Subtree Views
2) Create a new Subtree View
3) View and edit an existing Subtree View
4) Delete an existing Subtree View

b) back
q) quit
```

Enter choice [b]:

## Editing the Client Connection Policy

Finally, we edit the client connection policy to add our new **o=sample** subtree view.

In the configuration console main menu, we select “Client Connection Policy” to edit the client connection policy.

```
>>>> UnboundID Directory Proxy Server configuration console
main menu
```

What do you want to configure?

- |                             |                              |
|-----------------------------|------------------------------|
| 1) Client Connection Policy | 10) Log Retention Policy     |
| 2) Connection Handler       | 11) Log Rotation Policy      |
| 3) External Server          | 12) Placement Algorithm      |
| 4) Global Attribute Index   | 13) Proxy Transformation     |
| 5) Global Configuration     | 14) Request Processor        |
| 6) LDAP Health Check        | 15) Server Affinity Provider |
| 7) Load Balancing Algorithm | 16) Subtree View             |
| 8) Location                 | 17) Work Queue               |
| 9) Log Publisher            |                              |
- o) 'Basic' objects are shown - change this  
q) quit

Enter choice: 1

We enter 3 to edit the existing default client connection policy.

>>>> Client Connection Policy management menu

What would you like to do?

- 1) List existing Client Connection Policies  
2) Create a new Client Connection Policy  
3) View and edit an existing Client Connection Policy  
4) Delete an existing Client Connection Policy
- b) back  
q) quit

Enter choice [b]: 3

>>>> There is only one Client Connection Policy: 'default'. Are you sure that this is the correct one? (yes / no) [yes]:

In the configuration properties, we enter 16 to add a value to the `subtree-view` property.

>>>> Configure the properties of the Client Connection Policy

|     | Property                      | Value(s)                                                                                      |
|-----|-------------------------------|-----------------------------------------------------------------------------------------------|
|     | -----                         | -----                                                                                         |
| 1)  | policy-id                     | default                                                                                       |
| 2)  | description                   | -                                                                                             |
| 3)  | enabled                       | true                                                                                          |
| 4)  | evaluation-order-index        | 9999                                                                                          |
| 5)  | connection-criteria           | -                                                                                             |
| 6)  | allowed-operation             | abandon,<br>add, bind,<br>compare,<br>delete,<br>extended,<br>modify,<br>modify-dn,<br>search |
| 7)  | allowed-extended-operation    | -                                                                                             |
| 8)  | denied-extended-operation     | -                                                                                             |
| 9)  | allowed-auth-type             | sasl, simple                                                                                  |
| 10) | allowed-sasl-mechanism        | -                                                                                             |
| 11) | denied-sasl-mechanism         | -                                                                                             |
| 12) | include-backend-subtree-views | true                                                                                          |
| 13) | included-backend-base-dn      | -                                                                                             |

```
14) excluded-backend-base-dn -
15) include-backend-server-passthrough-subtree-views true
16) subtree-view dc_example_dc_com-view
?) help
f) finish - apply any changes to the Client Connection Policy
d) display the equivalent dsconfig command lines to either
    re-create this object or only to apply pending changes
b) back
q) quit
```

Enter choice [b]: 16

We enter 2 to add the new subtree view that we created for Sample earlier.

>>>> Configuring the 'subtree-view' property

Specifies the set of manually-configured subtree views that will be accessible to clients associated with this Client Connection Policy.

If the base DN for a manually-configured subtree view conflicts with the base DN for an automatically-included backend subtree view (if the include-backend-subtree-views property has a value of true), then the manually-configured subtree view will be used rather than the automatically-included backend subtree view.

Do you want to modify the 'subtree-view' property?

```
1) Keep the value: dc_example_dc_com-view
2) Add one or more values
3) Remove one or more values
4) Leave undefined

?) help
q) quit
```

Enter choice [1]: 2

We enter 1 to select the subtree view we created in the previous section.

Select the Subtree Views you wish to add:

```
1) o_sample-view
2) Create a new Subtree View

?) help
b) back
q) quit
```

Enter one or more choices separated by commas [b]: 1

We review the subtree views now referenced by the property and enter 1 to use these values.

>>>> Configuring the 'subtree-view' property (Continued)

The 'subtree-view' property references the following Subtree Views:

```
*) dc_example_dc_com-view
*) o_sample-view
```

Do you want to modify the 'subtree-view' property?

- 1) Use these values
- 2) Add one or more values
- 3) Remove one or more values
- 4) Leave undefined
- 5) Revert changes
- ?) help
- q) quit

Enter choice [1]:

We review the configuration properties of the client connection policy and enter f to save our changes.

>>>> Configure the properties of the Client Connection Policy

|       | Property                                         | Value(s)                                                                                   |
|-------|--------------------------------------------------|--------------------------------------------------------------------------------------------|
| ----- |                                                  |                                                                                            |
| 1)    | policy-id                                        | default                                                                                    |
| 2)    | description                                      | -                                                                                          |
| 3)    | enabled                                          | true                                                                                       |
| 4)    | evaluation-order-index                           | 9999                                                                                       |
| 5)    | connection-criteria                              | -                                                                                          |
| 6)    | allowed-operation                                | abandon,<br>add, bind,<br>compare, delete,<br>extended,<br>modify,<br>modify-dn,<br>search |
| 7)    | allowed-extended-operation                       | -                                                                                          |
| 8)    | denied-extended-operation                        | -                                                                                          |
| 9)    | allowed-auth-type                                | sasl, simple                                                                               |
| 10)   | allowed-sasl-mechanism                           | -                                                                                          |
| 11)   | denied-sasl-mechanism                            | -                                                                                          |
| 12)   | include-backend-subtree-views                    | true                                                                                       |
| 13)   | included-backend-base-dn                         | -                                                                                          |
| 14)   | excluded-backend-base-dn                         | -                                                                                          |
| 15)   | include-backend-server-passthrough-subtree-views | true                                                                                       |
| 16)   | subtree-view                                     | dc_example_dc_<br>com-view,<br>o_sample-view                                               |

- ?) help
- f) finish - apply any changes to the Client Connection Policy
- d) display the equivalent dsconfig command lines to either re-create this object or only to apply pending changes
- b) back
- q) quit

Enter choice [b]: f

The Client Connection Policy was modified successfully

We enter b to go back to the main configuration menu

>>>> Client Connection Policy management menu



What would you like to do?

- 1) List existing Client Connection Policies
  - 2) Create a new Client Connection Policy
  - 3) View and edit an existing Client Connection Policy
  - 4) Delete an existing Client Connection Policy
- b) back  
q) quit

Enter choice [b]:

## Testing Proxy Transformations

After setting up the deployment scenario, the proxy server will now respond to requests to the `dc=example,dc=com` and `o=sample` base DNs. We now test the service by imitating Sample client requests to search and modify users.

The following example fetches the user with `sampleID=scase` under the `ou=east,o=sample` base DN:

```
root@proxy-east-01: bin/ldapsearch --bindDN "cn=directory manager"
--bindPassword password --baseDN "ou=east,o=sample" "(sampleID=scase)"

dn: sampleID=scase,ou=People,ou=east,o=sample
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: exampleAccount
objectClass: top
description: A customer account migrated from Sample merger
sampleID: scase
userPassword: {SSHA}A5O4RrQHwXc2Ii3btD4exGdP0TVW9VL3CR3ZXA==
exAccountNumber: 234098
givenName: Sterling
cn: Sterling Case
sn: Case
telephoneNumber: +1 804 094 3356
street: 00468 Second Street
mail: sterlingcase@maildomain.com
l: Arlington
st: VA
sampleRegion: east
sampleLinkedAccounts: sampleID=jcase,ou=People,ou=east,o=sample
```

Next, we modify the `sampleRegion` value, changing it to `west`. To do this, we first create a `ldapmodify` input file, called `scase-mod.ldif`, with the following contents:

```
dn: sampleID=scase,ou=People,ou=east,o=sample
changetype: modify
replace: sampleRegion
sampleRegion: west
```

Now we use the file as an argument in the `ldapmodify` command as follows.

```
root@proxy-east-01: bin/ldapmodify --bindDN "cn=Directory Manager"
--bindPassword password --filename scase-mod.ldif
```

We should see the following output:

```
Processing MODIFY request for sampleID=scase,ou=People,
ou=east,o=sample

MODIFY operation successful for DN sampleID=scase,ou=People,
ou=east,o=sample
```

If we search for scase's `sampleRegion` value under `o=sample`, we should see `west`:

```
root@proxy-east-01: bin/ldapsearch --bindDN "cn=directory manager"
--bindPassword password --baseDN "o=sample" "(sampleID=scase)" \
sampleRegion

dn: sampleID=scase,ou=People,ou=east,o=sample
sampleRegion: west
```

We can also search for `scase` by `uid` rather than `sampleID`, under the `dc=example,dc=com` base DN. We see the Example.com schema version of the entry:

```
root@proxy-east-01: bin/ldapsearch --bindDN "cn=directory manager"
--bindPassword password --baseDN "dc=example,dc=com" "(uid=scase)"

dn: uid=scase,ou=People,dc=example,dc=com
objectClass: person
objectClass: exampleAccount
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: top
description: A customer account migrated from Sample merger
uid: scase
userPassword: {SSHA}A5O4RrQHwXc2Ii3btD4exGdP0TVW9VL3CR3ZXA==
exAccountNumber: 234098
givenName: Sterling
cn: Sterling Case
sn: Case
telephoneNumber: +1 804 094 3356
street: 00468 Second Street
mail: sterlingcase@maildomain.com
l: Arlington
st: VA
exSampleRegion: west
exSampleLinkedAccounts: uid=jcase,ou=People,dc=example,dc=com
```

## Deploying an Entry-Balancing Proxy Configuration

In this deployment scenario, the `austin-proxy1` instance of the directory proxy server communicates with four external servers. The directory proxy server is configured to use entry balancing, so that two sets of user entries are split among the four external servers. The first set of user entries is defined with the replicated pair of external servers `austin-set1.example.com` and `newyork-set1.example.com`. The second set of entries is defined in `austin-set2.example.com` and `newyork-set2.example.com`. The entry

balancing base DN is `ou=people,dc=example,dc=com`. The data below this base DN will be split among the four external servers, while the data above the `ou=people,dc=example,dc=com` entry-balancing point is defined identically on each server.

Any entry modification outside of the entry-balancing base DN is performed against all backends. The entry balancing behavior is defined by a entry-balancing request processor that is itself configured using subordinate proxying request processors.

This scenario references the following servers, two directory servers and a proxy server located in Austin and two directory servers located in New York:

- `austin-proxy1.example.com`
- `austin-set1.example.com`
- `austin-set2.example.com`
- `newyork-set1.example.com`
- `newyork-set2.example.com`

## Overview of Deployment Steps

In this deployment scenario, we will take the following steps:

- Install the proxy server on `austin-proxy1`.
- Use the `create-initial-proxy-config` tool to provide our initial setup for entry-balancing. The initial setup includes defining multiple subtree views and global indexes in support of entry balancing.
- Change the placement algorithm of the `proxy-01` server to use an entry-count placement algorithm. This algorithm is used to select the backend set to which to forward an add request. It looks at the number of entries in the backend sets and forwards the add request to the backend with either the fewest or the most entries, depending on the configuration. You can also configure the placement algorithm to make the decision based on the on-disk database size rather than the number of entries.

## Installing Directory Proxy Server

We start by configuring the directory proxy server. The four external servers, `austin-set1.example.com`, `newyork-set1.example.com`, `austin-set2.example.com`, and `newyork-set2.example.com`, are running.

We run the `setup` program as follows.

```
root@austin-proxy1: ./setup --acceptLicense \  
--listenAddress austin-proxy1.example.com \  
--ldapPort 389 --rootUserDN "cn=Directory Manager" \  
--rootUserPassword pass --entryBalancing \  
--aggressiveJVMTuning --maxHeapSize 2g --no-prompt
```

## Configuring the Entry-Balancing Directory Proxy Server

Once the directory proxy server has been installed, it can be automatically configured using the `create-initial-proxy-config` tool. This tool can only be used once for this initial configuration, after which we will have to use `dsconfig` to make any changes to our directory proxy server configuration.

We run the `create-initial-proxy-config` tool as follows:

```
root@austin-proxy1: ./bin/create-initial-proxy-config
```

Our topology meets the requirements, so we press return to continue:

```
Some assumptions are made about the topology in order to keep
this tool simple:
```

- 1) all servers will be accessible via a single user account
- 2) all servers support the same communication security type
- 3) all servers are UnboundID Directory Servers

```
If your topology does not have these characteristics you can
use this tool to define a basic configuration and then use the
'dsconfig' tool or the web console to fine tune the configura-
tion.
```

```
Would you like to continue? (yes / no) [yes]:
```

First, we provide the external server access credentials. All of our directory proxy servers have identical proxy user accounts and passwords.

```
Enter the DN of the proxy user account [cn=Proxy User]:
```

```
Enter the password for 'cn=Proxy User,cn=Root DNs,cn=config':
```

```
Confirm the password for 'cn=Proxy User,cn=Root DNs,cn=config':
```

Next, we specify the type of security we are using. In this example, we do not use security.

```
Specify the type of security that the Directory Proxy Server
will use when communicating with Directory Server instances:
```

- 1) None
- 2) SSL
- 3) StartTLS
  
- b) back
- q) quit

```
Enter choice [1]:
```

We will first enter the `dc=example,dc=com` base DN as a non-entry-balancing base DN.

```
Enter a base DN of the Directory Server instances that will be
accessed through the Directory Proxy Server:
```

- b) back
- q) quit

```
Enter a DN or choose a menu item [dc=example,dc=com]:
```

Next, we define the balancing point as a separate base DN, which is entry balanced:

Enter another base DN of the directory server instances that will be accessed through the Directory Proxy Server:

```
1) Remove dc=example,dc=com
b) back
q) quit
```

Enter a DN or choose a menu item [Press ENTER when finished entering base DNs]: ou=people,dc=example,dc=com

Are entries within 'ou=people,dc=example,dc=com' split across multiple servers so that each server stores only a subset of the entries (i.e. is this base DN 'entry balanced')? (yes / no)  
[no]: yes

In this example, the data in ou=people,dc=example,d=com will be split across two back-end sets. So we enter 2 to specify that the data will be balanced across two sets of servers.

Across how many sets of servers is the data balanced?

```
c) cancel creating ou=people,dc=example,dc=com
q) quit
```

Enter a number greater than one or choose a menu item: 2

The balancing point is the same as our base DN, ou=people,dc=example,dc=com.

>>>> Entry Balancing Base

The entry balancing base DN specifies the entry below which the data is balanced. Entries not below this entry must be duplicated in all the server sets. If all the entries in the base DN are distributed the entry balancing base DN is the same as the base DN.

```
c) cancel creating ou=people,dc=example,dc=com
b) back
q) quit
```

Enter the entry balancing base DN or choose a menu item  
[ou=people,dc=example,dc=com]: ou=people,dc=example,dc=com

To improve the performance for equality search filters referencing the uid attribute, we create a uid global index. We enter yes to add a new attribute.

Would you like to add attributes to the global index? (yes / no)  
[no]: yes

Next, we specify the uid attribute.

Enter attributes that you would like to add to the global index:

```
c) cancel creating ou=people,dc=example,dc=com
b) back
q) quit
```

```
Enter an attribute name or choose a menu item [Press ENTER when
finished entering index attributes]: uid
```

To optimize directory proxy server performance from the moment it starts accepting connections, we enter 3 to prime the global index:

```
Should the index for attribute 'uid' be primed such that it is
loaded into memory before the Directory Proxy Server begins
accepting connections?
```

- 1) Yes
- 2) No
- 3) Yes, and all subsequent attributes
- 4) No, and all subsequent attributes

```
Enter choice [1]: 3
```

```
Are the values of 'uid' guaranteed to be unique? (yes / no)
[no]: yes
```

We press enter to finish specifying index attributes.

```
Enter attributes that you would like to add to the global index:
```

- 1) Remove uid (primed,unique)
- c) cancel creating ou=people,dc=example,dc=com
- b) back
- q) quit

```
Enter an attribute name or choose a menu item [Press ENTER when
finished entering index attributes]:
```

We press enter to enable RDN index priming.

```
Would you like to enable RDN index priming for
'ou=people,dc=example,dc=com'? (yes / no) [yes]:
```

We press enter to finish specifying base DNs.

```
Enter another base DN of the directory server instances that
will be accessed through the Directory Proxy Server:
```

- 1) Remove dc=example,dc=com
- 2) Remove ou=people,dc=example,dc=com (distributed)
- b) back
- q) quit

```
Enter a DN or choose a menu item [Press ENTER when finished
entering base DNs]:
```

The external

servers are spread among two locations, New York and Austin. This proxy server instance is located in the `austin` location.

A good rule of thumb when naming locations is to use the name of your data centers or the cities containing them.

```
    b)  back
    q)  quit

Enter a location name or choose a menu item: austin

    1)  Remove austin
    b)  back
    q)  quit
```

Next, we define the `newyork` location:

```
Enter another location name or choose a menu item [Press ENTER
when finished entering locations]: newyork

    1) Remove austin
    2) Remove newyork
    b)  back
    q)  quit

Enter another location name or choose a menu item [Press ENTER
when finished entering locations]:
```

We select the `austin` location for this directory proxy server instance:

```
Choose the location for this Directory Proxy Server

    1) austin
    2) newyork
    b) back
    q) quit

Enter choice [1]:
```

Next, we specify the LDAP external server instances associated with this location.

```
Enter the host and port (host:port) of the first directory
server in 'austin'

    b)  back
    q)  quit

Enter a host:port or choose a menu item [localhost:389]:
austin-set1.example.com:389
```

Next, we specify that the `austin-set1` server can handle requests from the global domain and from set 1 restricted domain.

```
Assign server austin-set1.example.com:389 to handle requests
for one or more of the defined sets of data:

    1) dc=example,dc=com
    2) ou=people,dc=example,dc=com; Server Set 1
    3) ou=people,dc=example,dc=com; Server Set 2

Enter one or more choices separated by commas: 1,2
```

We enter 3 to prepare the server for access by the directory proxy server.

Would you like to prepare austin-set1.example.com:389 for access by the Directory Proxy Server?

- 1) Yes
- 2) No
- 3) Yes, and all subsequent servers
- 4) No, and all subsequent servers

Enter choice [3]:

Next, we select the entry-balanced data set that the austin-set1 server replicates with other servers.

You may choose a single entry-balanced data set with which austin-set1.example.com:389 will replicate data with other servers

- 1) ou=people,dc=example,dc=com; Server Set 1
- 2) None, data will not be replicated

Enter choice: 1

Testing connection to austin-set1.example.com:389 ..... Done  
Testing 'cn=Proxy User,cn=Root DNs,cn=config' access ....Denied

We modify the root user for use by the directory proxy server, specifying the directory manager password for the initial creation of the proxy user.

Would you like to create or modify root user 'cn=Proxy User,cn=Root DNs,cn=config' so that it is available for this Directory Proxy Server? (yes / no) [yes]:

Enter the DN of an account on austin-set1.example.com:389 with which to create or manage the 'cn=Proxy User,cn=Root DNs,cn=config' account and configuration [cn=Directory Manager]:

Enter the password for 'cn=Directory Manager':

Created 'cn=Proxy User,cn=Root DNs,cn=config'  
Testing 'cn=Proxy User,cn=Root DNs,cn=config' privileges...Done  
Setting replication set name .....

Since the replication set name has already been configured, we do not need to use the name created automatically by the directory proxy server.

This server is currently configured for replication set 'dataset1'. Would you like to reconfigure this server for replication set 'set-1'? (yes / no) [no]:

Setting replication set name ..... Done  
Verifying backend 'dc=example,dc=com' ..... Done  
Verifying backend 'ou=people,dc=example,dc=com' ..... Done  
Testing 'cn=Proxy User' privileges ..... Done  
Verifying backend 'dc=example,dc=com' ..... Done



Next, we define the other Austin and New York servers using the same procedure as in the previous example:

```
Enter another server in 'austin'
```

```
1) Remove austin-set1.example.com:389
```

```
b) back
```

```
q) quit
```

```
Enter a host:port or choose a menu item [Press ENTER when finished entering servers]: austin-set2.example.com:389
```

```
Assign server austin-set2.example.com:389 to handle requests for one or more of the defined sets of data
```

```
1) dc=example,dc=com
```

```
2) ou=people,dc=example,dc=com; Server Set 1
```

```
3) ou=people,dc=example,dc=com; Server Set 2
```

```
Enter one or more choices separated by commas: 1,3
```

```
You may choose a single entry-balanced data set with which austin-set2.example.com:389 will replicate data with other
```

```
servers
```

```
1) ou=people,dc=example,dc=com; Server Set 2
```

```
2) None, data will not be replicated
```

```
Enter choice: 1
```

```
Testing connection to austin-set2.example.com:389 ....Done
```

```
Testing 'cn=Proxy User,cn=Root DNs,cn=config' access ... Denied
```

```
Would you like to create or modify root user 'cn=Proxy User,cn=Root DNs,cn=config' so that it is available for this Directory Proxy Server? (yes / no) [yes]:
```

```
Would you like to use the previously entered manager credentials to access all prepared servers? (yes / no) [yes]:
```

```
Created 'cn=Proxy User,cn=Root DNs,cn=config'
```

```
Testing 'cn=Proxy User,cn=Root DNs,cn=config' privileges...Done
```

```
Setting replication set name .....
```

```
This server is currently configured for replication set 'dataset2'. Would you like to reconfigure this server for replication set 'set-2'? (yes / no) [no]:
```

```
Setting replication set name ..... Done
```

```
Verifying backend 'dc=example,dc=com' ..... Done
```

```
Verifying backend 'ou=people,dc=example,dc=com' ..... Done
```

```
Enter another server in 'austin'
```

```
1) Remove austin-set1.example.com:389
```

```
2) Remove austin-set2.example.com:389
```

```
b) back
```

```
q) quit
```

Enter a host:port or choose a menu item [Press ENTER when finished entering servers]:

```
>>>> >>>> Location 'newyork' Details
>>>> External Servers
```

External Servers identify directory server instances including host, port, and authentication information.

Enter the host and port (host:port) of the first directory server in 'newyork':

```
b)back
q)quit
```

Enter a host:port or choose a menu item [localhost:389]:  
newyork-set1.example.com:389

Assign server newyork-set1.example.com:389 to handle requests for one or more of the defined sets of data

```
1)dc=example,dc=com
2)ou=people,dc=example,dc=com; Server Set 1
3)ou=people,dc=example,dc=com; Server Set 2
```

Enter one or more choices separated by commas: 1,2

You may choose a single entry-balanced data set with which newyork-set1.example.com:389 will replicate data with other servers

```
1)ou=people,dc=example,dc=com; Server Set 1
2)None, data will not be replicated
```

Enter choice: 1

Testing connection to newyork-set1.example.com:389 ....Done  
Testing 'cn=Proxy User,cn=Root DNs,cn=config' access ... Denied

Would you like to create or modify root user 'cn=Proxy User,cn=Root DNs,cn=config' so that it is available for this Directory Proxy Server? (yes / no) [yes]:

```
Created 'cn=Proxy User,cn=Root DNs,cn=config'
Testing 'cn=Proxy User,cn=Root DNs,cn=config' privileges...Done
Setting replication set name .....
```

This server is currently configured for replication set 'dataset1'. Would you like to reconfigure this server for replication set 'set-1'? (yes / no) [no]:

```
Setting replication set name ..... Done
Verifying backend 'dc=example,dc=com' ..... Done
Verifying backend 'ou=people,dc=example,dc=com' ..... Done
```

Enter another server in 'newyork'

```
1)Remove newyork-set1.example.com:389
b)back
q)quit
```

Enter a host:port or choose a menu item [Press ENTER when finished entering servers]: newyork-set2.example.com:389

Assign server newyork-set2.example.com:389 to handle requests for one or more of the defined sets of data:

- 1)dc=example,dc=com
- 2)ou=people,dc=example,dc=com; Server Set 1
- 3)ou=people,dc=example,dc=com; Server Set 2

Enter one or more choices separated by commas: 1,3

You may choose a single entry-balanced data set with which newyork-set2.example.com:389 will replicate data with other servers

- 1)ou=people,dc=example,dc=com; Server Set 2
- 2)None, data will not be replicated

Enter choice: 1

Testing connection to newyork-set2.example.com:389 ..... Done

Testing 'cn=Proxy User,cn=Root DNs,cn=config' access.... Denied

Would you like to create or modify root user 'cn=Proxy User,cn=Root DNs,cn=config' so that it is available for this Directory Proxy Server? (yes / no) [yes]:

Created 'cn=Proxy User,cn=Root DNs,cn=config'

Testing 'cn=Proxy User,cn=Root DNs,cn=config' privileges...DoneSetting replication set name .....

This server is currently configured for replication set 'dataset2'. Would you like to reconfigure this server for replication set 'set-2'? (yes / no) [no]:

Setting replication set name ..... Done

Verifying backend 'dc=example,dc=com' ..... Done

Verifying backend 'ou=people,dc=example,dc=com' ..... Done

Enter another server in 'newyork'

- 1)Remove newyork-set1.example.com:389
- 2)Remove newyork-set2.example.com:389
- b)back
- q)quit

Enter a host:port or choose a menu item [Press ENTER when finished entering servers]:

>>>> >>>> Configuration Summary

External Server Security: None

Proxy User DN: cn=Proxy User,cn=Root DNs,cn=config

Location austin

Failover Order: newyork

Servers: austin-set1.example.com:389,  
austin-set2.example.com:389

```
Location newyork
  Failover Order: austin
  Servers: newyork-set1.example.com:389,
          newyork-set2.example.com:389

Base DN: dc=example,dc=com
  Servers: austin-set1.example.com:389,
          austin-set2.example.com:389,
          newyork-set1.example.com:389,
          newyork-set2.example.com:389

Base DN: ou=people,dc=example,dc=com
  Entry Balancing Base: ou=people,dc=example,dc=com
  Server Set 1: austin-set1.example.com:389,
               newyork-set1.example.com:389
  Server Set 2: austin-set2.example.com:389,
               newyork-set2.example.com:389
  Index Attributes: uid (primed,unique)
  Prime RDN Index: Yes

NOTE: The Directory Proxy Server must be restarted after this
tool has completed to have index priming take place

b)back
q)quit
w)write configuration

Enter choice [w]

>>>> Write Configuration

The configuration will be written to a 'dsconfig' batch file
that can be used to configure other Directory Proxy Servers

Writing Directory Proxy Server configuration to
/proxy/dps-cfg.txt.....Done

We enter yes to apply our configuration changes to the directory proxy server.

Apply these configuration changes to the local Directory Proxy Server?
(yes /no) [yes]:

How do you want to connect to the Directory Proxy Server on localhost?
1) LDAP
2) LDAP with SSL
3) LDAP with StartTLS
Enter choice [1]:

Administrator user bind DN [cn=Directory Manager]:

Password for user 'cn=Directory Manager':

Creating Locations ..... Done
Updating Failover Locations ..... Done
Updating Global Configuration ..... Done
Creating Health Checks ..... Done
Creating External Servers ..... Done
```

```
Creating Load-Balancing Algorithm for dc=example,dc=com .... Done
Creating Request Processor for dc=example,dc=com ..... Done
Creating Subtree View for dc=example,dc=com ..... Done
Updating Client Connection Policy for dc=example,dc=com ..... Done
Creating Load-Balancing Algorithm for ou=people,dc=example,dc=com;
Server Set 1 ..... Done
Creating Request Processor for ou=people,dc=example,dc=com; Server Set
1...Done
Creating Load-Balancing Algorithm for ou=people,dc=example,dc=com;
Server Set 2 .... Done
Creating Request Processor for ou=people,dc=example,dc=com; Server Set
2...Done
Creating Entry Balancing Request Processor for ou=people,dc=example,
dc=com ..... Done
Creating Placement Algorithm for ou=people,dc=example,dc=com .... Done
Creating Global Attribute Indexes for ou=people,dc=example,dc=com
..... Done
Creating Subtree View for ou=people,dc=example,dc=com ..... Done
Updating Client Connection Policy for ou=people,dc=example,dc=com
.....Done
```

See `/logs/create-initial-proxy-config.log` for a detailed log of this operation

To see basic server configuration status and configuration you can launch `/bin/status`

## Configuring the Placement Algorithm Using a Batch File

Now, we configure the placement algorithm using a batch file. We want to place new entries added through the proxy via LDAP ADD operations into the least used dataset. We do this using an entry count placement algorithm. To change the placement algorithm from round robin to entry count, we first create and enable an entry count placement algorithm configuration object and then disable the existing round robin placement algorithm. Our batch file, `dsconfig.post-setup`, contains the `dsconfig` commands required to create the entry-count placement algorithm and delete the old round robin algorithm.

The batch file contains comments to explain each `dsconfig` command. Note that in this example, line wrapping is used for clarity. The `dsconfig` command requires that the full command be provided on a single line.

The batch file itself looks as follows:

```
root@austin-proxy1:more ../dsconfig.post-setup

# This dsconfig operation creates the entry-count placement
# algorithm with the default behavior of adding entries to the
# smallest backend dataset first.

dsconfig create-placement-algorithm
--processor-name ou_people_dc_example_dc_com-eb-req-processor
--algorithm-name entry-count --type entry-counter --set enabled:true

# Note that once the entry-count placement algorithm is created
# and enabled, we can delete the round-robin algorithm.
# Since an entry-balancing proxy must always have a placement
```

```
# algorithm, we add a second algorithm and then delete the
# original round-robin algorithm created during the setup
# procedure.

dsconfig set-placement-algorithm-prop
--processor-name ou_people_dc_example_dc_com-eb-req-processor
--algorithm-name round-robin --set enabled:false

# At this point, LDAP ADD operations will be forwarded to an external
# server representing the dataset with the least number of entries.
```

We execute the batch file as follows:

```
root@austin-proxy1: bin/dsconfig --no-prompt
--bindDN "cn=directory manager" --bindPassword password
--port 389 --batch-file ../dsconfig.post-setup
Batch file '../dsconfig.post-setup' contains 2 commands

Executing: create-placement-algorithm --no-prompt
--bindDN "cn=directory manager" --bindPassword pass
--port 1389
--processor-name ou_people_dc_example_dc_com-eb-req-processor
--algorithm-name entry-count --type entry-counter --set enabled:true

Executing: delete-placement-algorithm --no-prompt
--bindDN "cn=directory manager" --bindPassword pass
--port 1389
--processor-name ou_people_dc_example_dc_com-eb-req-processor
--algorithm-name round-robin --set enabled:false
```

Once the batch file has executed, a new entry-counter placement algorithm, called **entry-count**, has been created, and the old round-robin placement algorithm, **round-robin**, has been deleted.

---

# 9

## Troubleshooting the Directory Proxy Server

This chapter provides the common problems and potential solutions that might occur when running UnboundID® Directory Proxy Server. It is primarily targeted at cases in which the directory proxy server is running on Solaris™ or Linux® systems, but much of the information can be useful on other platforms as well.

This chapter presents the following information:

- Debugging the Directory Proxy Server
- Working with the Troubleshooting Tools
- Common Problems and Potential Solutions

### Debugging the Directory Proxy Server

You can enable the JVM debugging options to track garbage collection data for your system. The options can impact JVM performance, but they provide valuable data to tune your server when troubleshooting garbage collection issues. You can add these options to the utility that you are testing by editing the `java.properties` file.

```
-XX:+PrintGCDetails  
-XX:+PrintTenuringDistribution  
-XX:+PrintGCApplicationConcurrentTime  
-XX:+PrintGCApplicationStoppedTime  
-XX:+PrintGCDateStamps
```

### Working with the Troubleshooting Tools

If problems arise with the directory proxy server (whether from issues in the directory proxy server itself or a supporting component, like the JVM, operating system, or hardware), then it is essential to be able to diagnose the problem quickly to determine the underlying cause and the best course of action to take towards resolving it.

This section provides information about how to perform this analysis to help ensure that the problem is resolved as quickly as possible. It targets cases in which the directory proxy server is running on Solaris™ or Linux® systems, but much of the information can be useful on other platforms as well.

## Working with the Collect Support Data Tool

The directory proxy server itself provides a significant amount of information about its current state including any problems that it has encountered during processing. If a problem occurs, the first step is to run the `collect-support-data` tool in the `bin` directory. The tool aggregates all relevant support files into a zip file that administrators can send to UnboundID for analysis. The tool also runs data collector utilities, such as `jps`, `jstack`, and `jstat` plus other diagnostic tools for Solaris and Linux machines, and bundles the results in the zip file.

The tool may only archive portions of certain log files to conserve space, so that the resulting support archive does not include hundreds of megabytes of log content.

Table 1 presents a summary of the data collectors that the `collect-support-data` tool archives in zip format. If an error occurs during processing, you can rerun the specific data collector command and send the results to UnboundID.

**TABLE 11-1. Data Collectors for the collect-support-data Tool**

| Data Collector    | Description                                                                                                                                                                                                            |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| config Directory  | Archives the config directory, which holds the servers configuration settings.                                                                                                                                         |
| logs Directory    | Archives the logs directory.                                                                                                                                                                                           |
| lib Directory     | Archives several configurable files in the lib directory, but not the libraries themselves. The directory holds the directory core server code plus the jar file as well as Oracle Berkeley DB Java Edition libraries. |
| monitor.ldif      | Archives all monitor entries.                                                                                                                                                                                          |
| rootdse.ldif      | Archives the rootDSE entry.                                                                                                                                                                                            |
| Directory Listing | Archives a list of all files in the specified directories and their sizes.                                                                                                                                             |
| jps               | Java Virtual Machine Process status tool. Reports information on the JVM.                                                                                                                                              |
| jstack            | Java Virtual Machine Stack Trace. Prints the stack traces of threads for the Java process.                                                                                                                             |
| jstat             | Java Virtual Machine Statistics Monitoring Tool. Displays performance statistics for the JVM.                                                                                                                          |
| jinfo             | Displays the Java configuration information for the Java process.                                                                                                                                                      |
| system-info       | Prints system, machine, and operating system information.<br>Solaris, Linux: <code>uname -a</code>                                                                                                                     |
| ps                | Prints a snapshot of the current active processes.<br>Solaris, Linux: <code>ps -elyZf</code>                                                                                                                           |



**TABLE 11-1. Data Collectors for the collect-support-data Tool**

| <b>Data Collector</b> | <b>Description</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| zonename              | Prints the name of the current zone.<br>Solaris: zonename.                                                                                                                                                                                                                                                                                                                                                                                                                      |
| zoneadm               | Prints the name of the current configured in verbose mode.<br>Solaris: zoneadm list -cv                                                                                                                                                                                                                                                                                                                                                                                         |
| df                    | Prints the amount of available disk space for filesystems in 1024-byte units.<br>Solaris, Linux: df -k                                                                                                                                                                                                                                                                                                                                                                          |
| zfs                   | Prints basic ZFS information: dataset pool names, and their used, available, referenced, and mountpoint properties.<br>Solaris: zfs list                                                                                                                                                                                                                                                                                                                                        |
| zpool                 | Print a zpool's status.<br>Solaris: zpool status,<br>zpool iostat -v [interval][count]                                                                                                                                                                                                                                                                                                                                                                                          |
| fmdump                | Prints the log files managed by the Solaris Fault Manager.<br>Solaris: fmdump -eV                                                                                                                                                                                                                                                                                                                                                                                               |
| prtconf               | Prints the system configuration information.<br>Solaris: prtconf                                                                                                                                                                                                                                                                                                                                                                                                                |
| prtdiag               | Prints the system diagnostic information.<br>Solaris: prtdiag                                                                                                                                                                                                                                                                                                                                                                                                                   |
| etc-system            | Prints the kernel parameters.<br>Solaris: cat /etc/system                                                                                                                                                                                                                                                                                                                                                                                                                       |
| messages              | Archives the messages generated by the directory proxy server.<br>Solaris: tail -100000c /var/adm/messages<br>Linux: tail -100000c /var/log/messages<br>The Messages data collector requires root or admin group privileges on Linux machines. UnboundID recommends that you run the tool with these privileges. If administrators running the tool do not have the privileges to run a particular data collector, they will get an error in the collector out log or zip file. |
| netstat-int           | Prints the state of all network interfaces.<br>Solaris, Linux: netstat -i                                                                                                                                                                                                                                                                                                                                                                                                       |
| netstat-sum           | Prints the statistics for each protocol.<br>Solaris, Linux: netstat -s                                                                                                                                                                                                                                                                                                                                                                                                          |
| netstat-rn            | Prints the kernel routing table.<br>Solaris, Linux: netstat -rn                                                                                                                                                                                                                                                                                                                                                                                                                 |
| ifconfig              | Prints information on all interfaces.<br>Solaris: ifconfig -au<br>Linux: ifconfig -a                                                                                                                                                                                                                                                                                                                                                                                            |
| uptime                | Prints the time the server has been up and active.<br>Solaris, Linux: uptime                                                                                                                                                                                                                                                                                                                                                                                                    |
| dmesg                 | Prints the message buffer of the kernel.<br>Solaris, Linux: dmesg                                                                                                                                                                                                                                                                                                                                                                                                               |

**TABLE 11-1. Data Collectors for the collect-support-data Tool**

| <b>Data Collector</b> | <b>Description</b>                                                                                                                                                                                                   |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| patchadd-p            | Prints the patches added to the system if any.<br>Solaris: patchadd -p                                                                                                                                               |
| cpuinfo               | Prints processor information.<br>Linux: cat /proc/cpuinfo                                                                                                                                                            |
| meminfo               | Prints the memory usage information.<br>Linux: cat /proc/meminfo                                                                                                                                                     |
| vmstat                | Prints information about virtual memory statistics.<br>Solaris, Linux:<br>vmstat {reportIntervalSeconds} {reportIntervalCount}                                                                                       |
| iostat                | Prints disk I/O and CPU utilization information.<br>Solaris: iostat -xn<br>Linux (through sysstat RPM on Red Hat):<br>iostat -x {reportIntervalSeconds} {reportCount}                                                |
| mpstat                | Prints performance statistics for all logical processors.<br>Solaris: mpstat {reportIntervalSeconds} {reportCount}<br>Linux (through sysstat RPM on Red Hat):<br>mpstat -P ALL {reportIntervalSeconds} {reportCount} |
| pstack                | Prints an execution stack trace on an active process specified by the pid.<br>Solaris: pstack {pid}<br>Linux: pstack {pid}                                                                                           |
| prstat                | Prints out information on all active processes on the system.<br>Solaris: prstat -L -n 1000 -p {pid} {reportIntervalSeconds} {reportCount}                                                                           |
| top                   | Prints a list of active processes and how much CPU and memory each process is using.<br>Linux: top -b -H -d {reportIntervalSeconds} -n {reportCount} -p {pid}                                                        |

## Available Tool Options

The `collect-support-data` tool has some important options that you should be aware of:

- **-R, --serverRoot {serverRootDir}**. Specifies the server root where the server from which to gather information is installed. By default, this is the server installed in the location from which this tool is invoked.
- **--noLdap**. Specifies that no effort should be made to collect any information over LDAP. This option should only be used if the server is completely unresponsive or will not start and only as a last resort.
- **--pid {pid}**. Specifies the ID of an additional process from which information is to be collected. This option is useful for troubleshooting external server tools and can be specified multiple times for each external server, respectively.

- **--sequential**. Use this option to diagnose “Out of Memory” errors. The tool collects data in parallel to minimize the collection time necessary for some analysis utilities. This option specifies that data collection should be run sequentially as opposed to in parallel. This action has the effect of reducing the initial memory footprint of this tool at a cost of taking longer to complete.
- **--reportCount {count}**. Specifies the number of reports generated for commands that support sampling (for example, vmstat, iostat, or mpstat). A value of 0 (zero) indicates that no reports will be generated for these commands. If this option is not specified, it defaults to 10.
- **--reportInterval {interval}**. Specifies the number of seconds between reports for commands that support sampling (for example, mpstat). This option must have a value greater than 0 (zero). If this option is not specified, it default to 1.

### To Run the Collect Support Data Tool

1. Go to the installation directory.

```
$ cd /UnboundID-Proxy
```

2. Use the collect-support-data tool. Make sure to include the host, port number, bind DN, and bind password.

```
$ bin/collect-support-data --hostname 127.0.0.1 --port 389 \  
--bindDN "cn=Directory Manager" --bindPassword secret \  
--serverRoot /opt/UnboundID-Proxy --pid 1234
```

3. Email the zip file to UnboundID Support.

|             |                                                                                                           |
|-------------|-----------------------------------------------------------------------------------------------------------|
| <b>Note</b> | You might need to change the .zip file extension to something like .zap to get through your mail servers. |
|-------------|-----------------------------------------------------------------------------------------------------------|

## Troubleshooting Tools Provided by the Directory Proxy Server

UnboundID Directory Proxy Server provides a set of tools that can also be used to obtain information for diagnosing and solving problems.

### Server Version Information

If it becomes necessary to contact UnboundID or an authorized support provider, then it will be important to provide precise information about the version of the directory proxy server software that is in use. If the server is running, then this information can be obtained from the "cn=Version,cn=monitor" entry. It can also be obtained using the command:

```
$ bin/start-proxy --fullVersion
```

This command outputs a number of important pieces of information, including:

- Major, minor, and point version numbers for the server.
- Source revision number from which the server was built.
- Time stamp indicating the time that the server was built.
- Optional list of issue numbers for patches that have been provided to solve problems encountered with the server.

## LDIF Connection Handler

The directory proxy server provides an LDIF connection handler that provides a way to request operations that does not require any network communication with the server. This can be particularly helpful if a configuration problem or bug in the server has left a connection handler unusable, or if all worker threads are busy processing operations.

The LDIF connection handler is enabled by default and looks for LDIF files to be placed in the `config/auto-process-ldif` directory. This directory does not exist by default, but if it is created and an LDIF file is placed in it containing one or more changes to be processed, then those changes will be applied.

Any changes that can be made over LDAP can be applied through the LDIF connection handler. It is primarily intended for administrative operations like updating the server configuration or scheduling tasks, although other types of changes (including changes to data contained in the server) can be processed. As the LDIF file is processed, a new file is written with comments for each change providing information about the result of processing that change.

## Embedded Profiler

If the directory proxy server appears to be running slowly, then it is helpful to know what operations are being processed in the server. The JVM Stack Trace monitor entry can be used to obtain a point-in-time snapshot of what the server is doing, but in many cases, it might be useful to have information collected over a period of time.

The embedded profiler is configured so that it is always available but is not active by default so that it has no impact on the performance of the running server. Even when it is running, it has a relatively small impact on performance, but it is recommended that it remain inactive when it is not needed. It can be controlled using the `dsconfig` tool or the web administration console by managing the "Profiler" configuration object in the "Plugin" object type, available at the standard complexity level. The `profile-action` property for this configuration object can have one of the following values:

- `start` – Indicates that the embedded profiler should start capturing data in the background.
- `stop` – Indicates that the embedded profiler should stop capturing data and write the information that it has collected to a `logs/profile{timestamp}` file.
- `cancel` – Indicates that the embedded profiler should stop capturing data and discard any information that it has collected.

Any profiling data that has been captured can be examined using the `profiler-viewer` tool. This tool can operate in either a text-based mode, in which case it dump a formatted text repre-

sensation of the profile data to standard output, or it can be used in a graphical mode that allows the information to be more easily understood.

### To Invoke the Profile Viewer in Text-based Mode

```
$ bin/profile-viewer --fileName logs/profile.20080101000000Z
```

### To Invoke the Profile View in GUI Mode

```
$ bin/profile-viewer --fileName logs/profile.20080101000000Z \  
--useGUI
```

## Troubleshooting Resources for Java Applications

Because the UnboundID Proxy Server is written entirely in Java, it is possible to use standard Java debugging and instrumentation tools when troubleshooting problems with the directory proxy server. In many cases, obtaining the full benefit of these tools requires access to the directory proxy server source code. These Java tools should be used under the advisement of UnboundID support.

### Java Troubleshooting Documentation

There are a number of documents providing general information about troubleshooting Java-based applications. Some of these documents include:

- <http://www.oracle.com/technetwork/java/javase/index-138283.html> – Troubleshooting Java SE
- <http://www.oracle.com/technetwork/java/javase/index-137495.html> – Troubleshooting Guide for Java SE 6 with HotSpot VM
- <http://www.sun.com/bigadmin/hubs/java/troubleshoot/> – BigAdmin Page on Java SE Troubleshooting
- <http://www.oracle.com/technetwork/java/javase/tools6-unix-139447.html> – Tools for troubleshooting Java on Solaris and Linux

### Java Troubleshooting Tools

The Java Development Kit provides a number of very useful tools to obtain information about Java applications and diagnosing problems. These tools are not included with the Java Runtime Environment™ (JRE), so the full Java Development Environment™ (JDK) should always be installed and used to run the UnboundID Proxy Server.

## jps

The `jps` tool is a Java-specific version of the UNIX® `ps` tool. It can be used to obtain a list of all Java processes currently running and their respective process identifiers. When invoked by a non-root user, it will list only Java processes running as that user. When invoked by a root user, then it lists all Java processes on the system.

This tool can be used to see if the directory proxy server is running and if a process ID has been assigned to it. This process ID can be used in conjunction with other tools to perform further analysis.

This tool can be run without any arguments, but some of the more useful arguments that include:

- `-v` – Includes the arguments passed to the JVM for the processes that are listed.
- `-m` – Includes the arguments passed to the main method for the processes that are listed.
- `-l` – (lowercase L). Include the fully qualified name for the main class rather than only the base class name.

Additional documentation for the `jps` tool is available at:

<http://download.oracle.com/javase/6/docs/technotes/tools/share/jps.html>

## jstack

The `jstack` tool is used to obtain a stack trace of a running Java process, or optionally from a core file generated if the JVM happens to crash. A stack trace can be extremely valuable when trying to debug a problems, because it provides information about all threads running and exactly what each is doing at the point in time that the stack trace was obtained.

Stack traces are helpful when diagnosing problems in which the server appears to be hung or behaving slowly. Java stack traces are generally more helpful than native stack traces, because Java threads can have user-friendly names (as do the threads used by the UnboundID Proxy Server), and the frame of the stack trace may include the line number of the source file to which it corresponds. This is useful when diagnosing problems and often allows them to be identified and resolved quickly.

In order to obtain a stack trace from a running JVM, use the command:

```
jstack {processID}
```

where *{processID}* is the process ID of the target JVM as returned by the `jps` command. To obtain a stack trace from a core file from a Java process, use the command:

```
jstack {pathToJava} {pathToCore}
```

where *{pathToJava}* is the path to the java command from which the core file was created, and *{pathToCore}* is the path to the core file to examine. In either case, the stack trace is written to

standard output and includes the names and call stacks for each of the threads that were active in the JVM.

In many cases, no additional options are necessary. The "-l" option can be added to obtain a long listing, which includes additional information about locks owned by the threads. The "-m" option can be used to include native frames in the stack trace.

Additional documentation for the jstack tool is available at <http://download.oracle.com/javase/6/docs/technotes/tools/share/jstack.html>.

## jmap

The jmap tool is used to obtain information about the memory consumed by the JVM. It is very similar to the native pmap tool provided by many operating systems. As with the jstack tool, jmap can be invoked against a running Java process by providing the process ID, or against a core file, like:

```
jmap {processID}
```

or

```
jmap {pathToJava} {pathToCore}
```

Some of the additional arguments include:

- **-dump:live,format=b,file=filename** – Dump the live heap data to a file that can be examined by the jhat tool
- **-heap** – Provides a summary of the memory used in the Java heap, along with information about the garbage collection algorithm in use.
- **-histo:live** – Provides a count of the number of objects of each type contained in the heap. If the “:live” portion is included, then only live objects are included; otherwise, the count include objects that are no longer in use and are garbage collected.

Additional information about the jmap tool can be found at <http://download.oracle.com/javase/6/docs/technotes/tools/share/jmap.html>.

## jhat

The jhat (Java Heap Analysis Tool) utility provides the ability to analyze the contents of the Java heap. It can be used to analyze a heap dump file, which is generated if the directory proxy server encounters an out of memory error (as a result of the "-XX:+HeapDumpOnOutOfMemoryError" JVM option) or from the use of the jmap command with the "-dump" option.

The jhat tool actually acts as a web server that can be accessed by a browser in order to query the contents of the heap. Several predefined queries are available to help determine the types of objects consuming significant amounts of heap space, and it also provides a custom query language (OQL, the Object Query Language) for performing more advanced types of analysis.

The `jhat` tool can be launched with the path to the heap dump file, like:

```
jhat /path/to/heap.dump
```

This command causes the `jhat` web server to begin listening on port 7000. It can be accessed in a browser at `http://localhost:7000` (or `http://address:7000` from a remote system). An alternate port number can be specified using the `-port` option, like:

```
jhat -port 1234 /path/to/heap.dump
```

In order to issue custom OQL searches, access the web interface using the URL `http://localhost:7000/oql/` (note the trailing slash must be provided). Additional information about the OQL syntax may be obtained in the web interface at `http://localhost:7000/oqlhelp/`. Additional information for the `jhat` tool may be found at `http://download.oracle.com/javase/6/docs/technotes/tools/share/jhat.html`.

## jstat

The `jstat` tool is used to obtain a variety of statistical information from the JVM, much like the `vmstat` utility that can be used to obtain CPU utilization information from the operating system. The general manner to invoke it is as follows:

```
jstat {type} {processID} {interval}
```

The `{interval}` option specifies the length of time in milliseconds between lines of output. The `{processID}` option specifies the process ID of the JVM used to run the directory proxy server, which can be obtained by running `jps` as mentioned previously. The `{type}` option specifies the type of output that should be provided. Some of the most useful types include:

- `-class` – Provides information about class loading and unloading.
- `-compile` – Provides information about the activity of the JIT complex.
- `-printcompilation` – Provides information about JIT method compilation.
- `-gc` – Provides information about the activity of the garbage collector.
- `-gccapacity` – Provides information about memory region capacities.

## Java Diagnostic Information

In addition to the tools listed in the previous section, the JVM can provide additional diagnostic information in response to certain events.

### Garbage Collection Diagnostic Information

If the JVM is not properly configured, then garbage collection activity can cause significant pauses in the execution of the Java code running in it. While the `jstat` utility with the `-gc` option can be used to obtain some information about garbage collection activity, there are additional arguments that can be added to the JVM to use when running the server to provide additional detail. Those arguments include:



- `-XX:+PrintGCDetails`
- `-XX:+PrintTenuringDistribution`
- `-XX:+PrintGCApplicationConcurrentTime`
- `-XX:+PrintGCApplicationStoppedTime`
- `-XX:+PrintGCDateStamps`

To run the directory proxy server with these options, edit the `config/java.properties` file and add them to the end of the line that begins with `"start-ds.java-args"`. After the file has been saved, invoke the following command to make those new arguments take effect the next time the server is started:

```
$ bin/dsjavaproperties
```

## JVM Crash Diagnostic Information

If the JVM itself should happen to crash for some reason, then it generates a fatal error log with information about the state of the JVM at the time of the crash. By default, this file is named `hs_err_pid{processID}.log` and is written into the directory that is the base of the directory proxy server installation. This file includes information the underlying cause of the JVM crash, information about the threads running and Java heap at the time of the crash, the options provided to the JVM, environment variables that were set, and information about the underlying system.

More information about the content that may be written to this log file may be found at <http://www.oracle.com/technetwork/java/javase/index-137495.html>.

## Troubleshooting Resources in the Operating System

The underlying operating system also provides a significant amount of information that can help diagnose issues that impact the performance and the stability of the directory proxy server. In some cases, problems with the underlying system can be directly responsible for the issues seen with the directory proxy server, and in others system, tools can help narrow down the cause of the problem.

### Identifying Problems with the Underlying System

If the underlying system itself is experiencing problems, it can adversely impact the function of applications running on it. Places to look for problems in the underlying system include:

- The system log file (`/var/adm/messages` on Solaris and `/var/log/messages` on Linux). Information about faulted or degraded devices or other unusual system conditions are written there.
- On Solaris systems, if the fault management system has detected a problem with a system component, information about that problem is obtain by running the `findump` command.

- If the ZFS filesystem is in use, then the `zpool status` command provides information about read errors, write errors, or data checksum errors.

## Examining CPU Utilization

Observing CPU utilization for the directory proxy server process and the system as a whole provides clues as to the nature of the problem.

### System-Wide CPU Utilization

To investigate CPU consumption of the system as a whole, use the `vmstat` command with a time interval in seconds, like:

```
vmstat 5
```

The specific output of this command varies between different operating systems, but it includes the percentage of the time the CPU was spent executing user-space code (user time), the percentage of time spent executing kernel-space code (system time), and the percentage of time not executing any code (idle time).

If the CPUs are spending most of their time executing user-space code, the available processors are being well-utilized. If performance is poor or the server is unresponsive, it can indicate that the directory is not optimally tuned. If there is a high system time, it can indicate that the system is performing excessive disk and/or network I/O, or in some cases, there can be some other system-wide problem like an interrupt storm. If the system is mostly idle but the directory proxy server is performing poorly or is unresponsive, there can be a resource constraint elsewhere (for example, waiting on disk or memory access, or excessive lock contention), or the JVM can be performing other tasks like stop-the-world garbage collection that cannot be heavily parallelized.

### Per-CPU Utilization

To investigate CPU consumption on a per-CPU basis, use the `mpstat` command with a time interval in seconds, like:

```
mpstat 5
```

On Linux systems, it might be necessary to add `"-P ALL"` to the command, like:

```
mpstat -P ALL 5
```

Among other things, this shows the percentage of time each CPU has spent in user time, system time, and idle time. If the overall CPU utilization is relatively low but `mpstat` reports that one CPU has a much higher utilization than the others, there might be a significant bottleneck within the server or the JVM might be performing certain types of garbage collection which cannot be run in parallel. On the other hand, if CPU utilization is relatively even across all CPUs, there is likely no such bottleneck and the issue might be elsewhere.

## Per-Process Utilization

To investigate CPU consumption on a per-process basis, use the `prstat` tool on Solaris or the `top` utility on Linux. If a process other than the Java process used to run the directory proxy server is consuming a significant amount of available CPU, it might be interfering with the ability of the directory proxy server to run effectively.

If the `mpstat` command showed that one CPU was much more heavily utilized than the others, it might be useful to identify the thread with the highest CPU utilization as it is likely the one that is a bottleneck preventing other threads from processing. On Solaris, this can be achieved by using the `prstat` command with the `"-L"` option, like:

```
prstat -L -p {processID}
```

This command will cause each thread to be displayed on a separate line, with the LWPID (lightweight process identifier) displayed as the last item on each line, separated from the process name by a slash. The thread that is currently consuming the largest amount of CPU will be displayed at the top of the list, and the `pstack` command can be used to identify which thread is responsible.

## Examining Disk Utilization

If the underlying system has a very high disk utilization, it can adversely impact directory proxy server performance. It could raise concerns for server stability if excessive disk I/O inhibits the ability of the cleaner threads to keep the database size under control.

The `iostat` tool may be used to obtain information about the disk activity on the system. On Solaris systems, this should be invoked using the `"-x"` and `"-n"` arguments, like:

```
iostat -x -n 5
```

On Linux systems, `iostat` should be invoked with the `"-x"` argument, like:

```
iostat -x 5
```

A number of different types of information will be displayed, but in order to obtain an initial feel for how busy the underlying disks are, look at the `"%b"` column on Solaris and the `"%util"` column on Linux. Both of these fields show the percentage of the time that the underlying disks are actively servicing I/O requests. A system with a high disk utilization likely exhibits poor directory proxy server performance.

If the high disk utilization is on one or more disks that are used to provide swap space for the system, the system might not have enough free memory to process requests. As a result, it might have started swapping blocks of memory that have not been used recently to disk. This can cause very poor server performance. It is important to ensure that the server is configured appropriately to avoid this condition. If this problem occurs on a regular basis, then the server is likely configured to use too much memory. If swapping is not normally a problem but it

does arise, then check to see if there are any other processes running, which are consuming a significant amount of memory, and check for other potential causes of significant memory consumption (for example, large files in a `tmpfs` filesystem).

On Solaris systems using ZFS, you can use the `zpool iostat {interval}` command to obtain information about I/O activity on a per-pool basis. While this command provides a useful display of the number of read and write operations and the amount of data being read from and written to the disks, it does not actually show how busy the underlying disks. As a result, the `zpool iostat` command is generally not as useful as the traditional `iostat` command for identifying potential I/O bottlenecks.

## Examining Process Details

There are a number of tools provided by the operating system that can help examine a process in detail.

### **ps**

The standard `ps` tool can be used to provide a range of information about a particular process. For example, the command:

```
ps -fly -p {processID}
```

can be used to display the state of the process, the name of the user running the process, its process ID and parent process ID, the priority and nice value, resident and virtual memory sizes, the start time, the execution time, and the process name with arguments.

Note that for a process with a large number of arguments, the standard `ps` command displays only a limited set of the arguments based on available space in the terminal window. In that case, the BSD version of the `ps` command (available on Solaris as `/usr/ucb/ps`) can be used to obtain the full command with all arguments, like:

```
/usr/ucb/ps auxwww {processID}
```

### **pstack**

The `pstack` command can be used to obtain a native stack trace of all threads in a process. While a native stack trace might not be as user-friendly as a Java stack trace obtained using `jstack`, it includes threads that are not available in a Java stack trace. For example, the command displays those threads used to perform garbage collection and other housekeeping tasks. The general usage for the `pstack` command is:

```
pstack {processID}
```

## dbx / gdb

A process debugger provides the ability to examine a process in detail. Like `pstack`, a debugger can obtain a stack trace for all threads in the process, but it also provides the ability to examine a process (or core file) in much greater detail, including observing the contents of memory at a specified address and the values of CPU registers in different frames of execution. The GNU debugger `gdb` is widely-used on Linux systems and is available on Solaris, but the Sun Studio debugger `dbx` is generally preferred over `gdb` on Solaris.

Note that using a debugger against a live process interrupts that process and suspends its execution until it detaches from the process. In addition, when running against a live process, a debugger has the ability to actually alter the contents of the memory associated with that process, which can have adverse effects. As a result, it is recommended that the use of a process debugger be restricted to core files and only used to examine live processes under the direction of UnboundID Support staff.

## pfiles / lsof

To examine the set of files that a process is using (including special types of files, like sockets) on Solaris, you can use the `pfiles` command, like:

```
pfiles {processID}
```

On Linux systems, the `lsof` tool can be used, like:

```
lsof -p {processID}
```

## Tracing Process Execution

If a process is unresponsive but is consuming a nontrivial amount of CPU time, or if a process is consuming significantly more CPU time than is expected, it might be useful to examine the activity of that process in more detail than can be obtained using a point-in-time snapshot like you can get with `pstack` or a debugger. For example, if a process is performing a significant amount of disk reads and/or writes, it can be useful to see which files are being accessed. Similarly, if a process is consistently exiting abnormally, then beginning tracing for that process just before it exits can help provide additional information that cannot be captured in a core file (and if the process is exiting rather than being terminated for an illegal operation, then no core file may be available).

On Solaris systems, the `dtrace` tool provides an unmatched mechanism for tracing the execution of a process in extremely powerful and flexible ways, but it is also relatively complex and describing its use is beyond the scope of this document. In many cases, however, observing the system calls made by a process can reveal a great deal about what it is doing. This can be accomplished using the `truss` utility on Solaris or the `strace` tool on Linux.

The `truss` utility is very powerful and has a lot of options, but two of the most useful forms in which it may be invoked are:

- `truss -f -p {processID}` – Provides a basic overview of all system calls being made by the specified process (and any subprocesses that it creates) and their associated return values.
- `truss -fear all -p {processID}` – Provides an extremely verbose trace of all system call activity, including details about data being read from or written to files and sockets.

In both cases, the output may be written to a file instead of the terminal window by adding the `-o {path}` option. Further, rather than observing an already-running process, it is possible to have `truss` launch the process and trace execution over its entire life span by replacing `-p {processID}` with name and arguments for the command to invoke.

On Linux systems, the basic equivalent of the first `truss` variant above is:

```
strace -f -p {processID}
```

Consult the `strace` manual page for additional information about using it to trace process execution on Linux.

## Examining Network Communication

Because UnboundID Directory Proxy Server is a network-based application, it can be valuable to observe the network communication that it has with clients. The directory proxy server itself can provide details about its interaction with clients by enabling debugging for the `protocol` or `data` debug categories, but there may be a number of cases in which it is useful to view information at a much lower level. A network sniffer, like the `snoop` tool on Solaris or the `tcpdump` tool on Linux, can be used to accomplish this.

There are many options that can be used with these tools, and their corresponding manual pages will provide a more thorough explanation of their use. However, in order to perform basic tracing to show the full details of the packets received for communication on port 389 with remote host 1.2.3.4, the following commands can be used on Solaris and Linux, respectively:

```
snoop -d {interface} -r -x 0 host 1.2.3.4 port 389
```

```
tcpdump -i {interface} -n -XX -s 0 host 1.2.3.4 and port 389
```

On Solaris systems, the `snoop` command provides enhanced support for parsing LDAP communication (but only when the directory proxy server is listening on the default port of 389). By adding the `-v` argument to the `snoop` command line, a verbose breakdown of each packet will be displayed, including protocol-level information. It does not appear that the `tcpdump` tool provides support for LDAP parsing. However, in either case it is possible to write capture data to a file rather than displaying information on the terminal (using `-o {path}` with `snoop`, or `-w {path}` with `tcpdump`), so that information can be later analyzed with a graphical tool like Wireshark, which provides the ability to interpret LDAP communication on any port.

Note that enabling network tracing generally requires privileges that are not available to normal users and therefore may require root access. On Solaris systems, granting the

`net_rawaccess` privilege to a user should be sufficient to allow that user to run the `snoop` utility.

## Common Problems and Potential Solutions

If a problem does occur, information provided earlier in this chapter can help you determine the underlying cause and provide clues about how to resolve the problem. This section describes a number of different types of problems that can occur and common potential causes for them.

### The Server Will Not Run Setup

If the `setup` tool does not run properly, some of the most common reasons include:

#### A Suitable Java Environment Is Not Available

UnboundID Directory Proxy Server requires that Java 1.6.0 or higher be installed on the system and made available to the server, and it must be installed prior to running `setup`. If the `setup` tool does not detect that a suitable Java environment is available, it will refuse to run.

In order to ensure that this does not happen, the `setup` tool should be invoked with an explicitly-defined value for the `JAVA_HOME` environment variable that specifies the path to the Java installation that should be used. For example:

```
env JAVA_HOME=/ds/java ./setup
```

If this still does not work for some reason, then it can be that the value specified in the provided `JAVA_HOME` environment variable can be overridden by another environment variable. If that occurs, try the following command, which should override any other environment variables that can be set:

```
env UNBOUNDID_JAVA_HOME="/ds/java" \
ALUUNBOUNDID_JAVA_BIN="" ./setup
```

#### Unexpected Arguments Provided to the JVM

If the `setup` script attempts to launch the `java` command with an invalid set of Java arguments, it might prevent the JVM from starting. By default, no special options is provided to the JVM when running `setup`, but this might not be the case if either the `JAVA_ARGS` or `UNBOUNDIDALU_JAVA_ARGS` environment variable is set. If the `setup` tool displays an error message that indicates that the Java environment could not be started with the provided set of arguments, then invoke the following command before trying to re-run `setup`:

```
unset JAVA_ARGS UNBOUNDIDALU_JAVA_ARGS
```

## The Server Has Already Been Configured or Used

The `setup` tool is only intended to provide the initial configuration for the directory proxy server. It refuses to run if it detects that the `setup` tool has already been run, or if an attempt has been made to start the directory proxy server prior to running the `setup` tool. This is an attempt to protect an existing directory proxy server installation from being inadvertently updated in a manner that could harm an existing configuration or data set.

If the directory proxy server has been previously used and if you want to perform a fresh installation, it is recommended that you first remove the existing installation, create a new one and run `setup` in that new installation. However, if you are confident that there is nothing of value in the existing installation (for example, if a previous attempt to run `setup` failed to complete successfully for some reason but it will refuse to run again), the following steps can be used to allow the `setup` program to run:

- Remove the `config/config.ldif` file and replace it with the `config/upgrade/config.ldif.{revision}` file containing the initial configuration.
- If there are any files or subdirectories below the `db` directory, then remove them.
- If a `config/java.properties` file exists, then remove it.
- If a `lib/setup-java-home` script (or `lib\set-java-home.bat` file on Windows) exists, then remove it.

## The Server Will Not Start

If the directory proxy server does not start, then there are a number of potential causes.

### The Server or Other Administrative Tool Is Already Running

Only a single instance of the directory proxy server can run at any time from the same installation root. If an instance is already running, then subsequent attempts to start the server will fail. Similarly, some other administrative operations (for example, restoring a backup or importing data from an LDIF file) can also prevent the server from being started. In such cases, the attempt to start the server should fail with a message like:

```
The Directory Proxy Server could not acquire an exclusive lock
on file /ds/UnboundID-DS/locks/server.lock: The exclusive lock
requested for file /ds/UnboundID-DS/locks/server.lock was not
granted, which indicates that another process already holds a
shared or exclusive lock on that file. This generally means
that another instance of this server is already running
```

If the directory proxy server is not running (and is not in the process of starting up or shutting down) and there are no other tools like `restore` or `import-ldif` running that could prevent the server from being started, and the server still believes that it is running, then it is possible that a previously-held lock was not properly released. In that case, you can try removing all of the files in the `locks` directory before attempting to start the server.



If you wish to have multiple instances running at the same time on the same system, then you should create a completely separate installation in another location on the filesystem.

## There Is Not Enough Memory Available

When the directory proxy server is started, the JVM attempted to allocate all memory that it has been configured to use. If there is not enough memory available on the system, then the directory proxy server generates an error message that indicates that the server could not be started with the specified set of arguments. Note that it is possible that an invalid option was provided to the JVM (as described below), but if that same set of JVM arguments has already been used successfully to run the server, then it is more likely that the system does not have enough memory available.

There are a number of potential causes for this:

- If the amount of memory in the underlying system has changed (for example, system memory has been removed, or if the directory proxy server is running in a zone or other type of virtualized container and a change has been made to the amount of memory that container will be allowed to use), then the directory proxy server might need to be reconfigured to use a smaller amount of memory than had been previously configured.
- Another process running on the system is consuming a significant amount of memory so that there is not enough free memory available to start the server. If this is the case, then either terminate the other process to make more memory available for the directory proxy server, or reconfigure the directory proxy server to reduce the amount of memory that it attempts to use.
- The directory proxy server was just shut down and an attempt was made to immediately restart it. In some cases, if the server is configured to use a significant amount of memory, then it can take a few seconds for all of the memory that had been in use by the server, when it was previously running, to be released back to the operating system. In that case, run the `vmstat` command and wait until the amount of free memory stops growing before attempting to restart the server.
- If the system has one or more ZFS filesystems (even if the directory proxy server itself is not installed on a ZFS filesystem), but it has not been configured to limit the amount of memory that ZFS can use for caching, then it is possible that ZFS caching is holding onto a significant amount of memory and cannot release it quickly enough when it is needed by the directory proxy server. In that case, the system should be reconfigured to limit the amount of memory that ZFS is allowed to use.
- If the system is configured with one or more memory-backed filesystems for example, `tmpfs` used for `/tmp` for Solaris), then look to see if there are any large files that can be consuming a significant amount of memory in any of those locations. If so, then remove them or relocate them to a disk-based filesystem.

If nothing else works and there is still not enough free memory to allow the JVM to start, then as a last resort, try rebooting the system.

## An Invalid Java Environment or JVM Option Was Used

If an attempt to start the directory proxy server fails with an error message indicating that no valid Java environment could be found, or that indicates that the Java environment could not be started with the configured set of options, then you should first ensure that enough memory is available on the system as described above. If there is a sufficient amount of memory available, then other causes for this error can include the following:

- The Java installation that was previously used to run the server no longer exists (for example, an updated Java environment was installed and the old installation was removed). In that case, update the `config/java.properties` file to reference the path to the new Java installation and run the `bin/dsjavaproperties` command to apply that change.
- The Java installation used to run the server has been updated and the server is trying to use the correct Java installation but one or more of the options that had worked with the previous Java version no longer work with the new version. In that case, it is recommended that the server be reconfigured to use the previous Java version, so that it can be run while investigating which options should be used with the new installation.
- If an `UNBOUNDID_JAVA_HOME` or `UNBOUNDID_JAVA_BIN` environment variable is set, then its value may override the path to the Java installation used to run the server as defined in the `config/java.properties` file. Similarly, if an `UNBOUNDID_JAVA_ARGS` environment variable is set, then its value might override the arguments provided to the JVM. If this is the case, then explicitly unset the `UNBOUNDID_JAVA_HOME`, `UNBOUNDID_JAVA_BIN`, and `UNBOUNDID_JAVA_ARGS` environment variables before trying to start the server.

Note that any time the `config/java.properties` file is updated, the `bin/dsjavaproperties` tool must be run to apply the new configuration. If a problem with the previous Java configuration prevents the `bin/dsjavaproperties` tool from running properly, then it can be necessary to remove the `lib/set-java-home` script (or `lib\set-java-home.bat` file on Windows) and invoke the `bin/dsjavaproperties` tool with an explicitly-defined path to the Java environment, like:

```
env UNBOUNDID_JAVA_HOME=/ds/java bin/dsjavaproperties
```

## An Invalid Command-Line Option Was Provided

There are a small number of arguments that may be provided when running the `bin/start-proxy` command, but in most cases, none are required. If one or more command-line arguments were provided for the `bin/start-proxy` command and any of them is not recognized, then the server provides an error message indicating that an argument was not recognized and displays version information. In that case, correct or remove the invalid argument and try to start the server again.

## The Server Has an Invalid Configuration

If a change is made to the directory proxy server configuration using an officially-supported tool like `dsconfig` or the Directory Proxy Server Management Console, the server should validate that configuration before applying it. However, it is possible that a configuration change

can appear to be valid at the time that it is applied, but does not work as expected when the server is restarted. Alternately, a change in the underlying system can cause a previously-valid configuration to become invalid.

In most cases involving an invalid configuration, the directory proxy server displays (and write to the error log) a message that explains the problem, and this can be sufficient to identify the problem and understand what action needs to be taken to correct it. If for some reason the startup failure does not provide enough information to identify the problem with the configuration, then look in the `logs/config-audit.log` file to see what recent configuration changes have been made with the server online, or in the `config/archived-configs` directory to see if there might have been a recent configuration change resulting from a direct change in the configuration file itself that was not made through a supported configuration interface.

If the server does not start as a result of a recent invalid configuration change, then it can be possible to start the server using the configuration that was in place the last time that the server started successfully (for example, the "last known good" configuration). This can be achieved using the `"--useLastKnownGoodConfig"` option, like:

```
$ bin/start-proxy --useLastKnownGoodConfig
```

Note that if it has been a long time since the last time the server was started and a number of configuration changes have been made since that time, then the last known good configuration can be significantly out of date. In such cases, it can be preferable to manually repair the configuration.

If there is no last known good configuration, if the server no longer start with the last known good configuration, or if the last known good configuration is significantly out of date, then manually update the configuration by editing the `config/config.ldif` file. In that case, you should make sure that the server is offline and that you have made a copy of the existing configuration before beginning. You might wish to discuss the change with UnboundID or an authorized support representative before applying it to ensure that you understand the correct change that needs to be made.

## You Do Not Have Sufficient Permission

The directory proxy server should only be started by the user or role used to initially install the server. In most cases, if an attempt is made to start the server as a user or role other than the one used to create the initial configuration, then the server will fail to start, because the user will not have sufficient permissions to access files owned by the other user, such as, database and log files. However, if the server was initially installed as a non-root user and then the server is started by the root account, then it can no longer be possible to start the server as a non-root user because new files that are created would be owned by root and could not be written by other users.

If the server was inadvertently started by root when it is intended to be run by a non-root user, or if you wish to change the user account that should be used to run the server, then it should be sufficient to simply change ownership on all files in the directory proxy server installation, so that they are owned by the user or role under which the server should run. For example, if

the directory proxy server should be run as the "proxy" user in the "other" group, then the following command can be used to accomplish this:

```
chown -R proxy:other /UnboundID-Proxy
```

This command will likely need to be invoked by the root user.

## The Server Has Crashed or Shut Itself Down

If the directory proxy server was previously running but is no longer active, then the potential reasons include the following:

- The directory proxy server was shut down by an administrator. Unless the server was forcefully terminated (for example, using “kill -9”), then messages are written to the error log and server.out files explaining the reason for the shutdown.
- The directory proxy server was shut down when the underlying system crashed or was rebooted. If this is the case, then running the uptime command on the underlying system shows that it was recently booted.
- The directory proxy server process was terminated by the underlying operating system for some reason (for example, the out of memory killer on Linux). If this happens, then a message will be written to the system error log.
- The directory proxy server decided to shut itself down in response to a serious problem that had arisen. At present, this should only occur if the server has detected that the amount of usable disk space has become critically low, or if significant errors have been encountered during processing that left the server without any remaining worker threads to process operations. If this happens, then messages are written to the error log and server.out files (if disk space is available) to provide the reason for the shutdown.
- The JVM in which the directory proxy server was running crashed. If this happens, then the JVM should dump a fatal error log (a `hs_err_pid{processID}.log` file) and potentially a core file.

In the event that the operating system itself crashed or terminated the process, then you should work with your operating system vendor to diagnose the underlying problem. If the JVM crashed or the server shut itself down for a reason that is not clear, then contact UnboundID or your authorized support provider for further assistance.

## The Server Will Not Accept Client Connections

If the directory proxy server does not appear to be accepting connections from clients, then potential reasons include the following:

- The directory proxy server is not running.

- The directory proxy server is in the process of starting up but has not yet begun accepting client connections. If the directory proxy server is configured to automatically load database contents into memory on startup, then this can take a significant length of time (depending on the size of the database) although it writes messages to the error log as priming progresses.
- The underlying system on which the directory proxy server is installed is not running.
- The directory proxy server is running but is not reachable as a result of a network or firewall configuration problem. If that is the case, then connection attempts should time out rather than be rejected.
- If the directory proxy server is configured to allow secure communication via SSL or Start-TLS, then a problem with the key manager and/or trust manager configuration can cause connections to be rejected. If that is the case, then messages should be written to the server access log for each failed connection attempt.
- If the directory proxy server has been configured with a maximum allowed number of connections, then it can be that the maximum number of allowed client connections are already established. If that is the case, then messages should be written to the server access log for each rejected connection attempt.
- If the directory proxy server is configured to restrict access based on the address of the client, then messages should be written to the server access log for each rejected connection attempt.
- If a connection handler encounters a significant error, then it can stop listening for new requests. If this occurs, then a message should be written to the server error log with information about the problem. It can be possible to restart the connection handler potentially using the LDIF connection handler) to make it available again.

## The Server is Unresponsive

If the directory proxy server process is running and appears to be accepting connections but does not respond to requests received on those connections, then potential reasons for this behavior include:

- If all worker threads are busy processing other client requests, then new requests that arrive will be forced to wait in the work queue until a worker thread becomes available. If this is the case, then a stack trace obtained using the jstack command shows that all of the worker threads are busy and none of them are waiting for new requests to process.
- If a request handler is stuck performing some expensive processing for a client connection, then other requests sent to the server on connections associated with that request handler is forced to wait until the request handler is able to read data on those connections. If this is the case, then only some of the connections can experience this behavior (unless there is only a single request handler, in which it will impact all connections), and stack traces obtained using the jstack command shows that a request handler thread is continuously blocked rather than waiting for new requests to arrive.

- If the JVM in which the directory proxy server is running is not properly configured, then it can be forced to spend a significant length of time performing garbage collection, and in severe cases, could cause significant interruptions in the execution of Java code. In such cases, a stack trace obtained from a pstack of the native process should show that most threads are ideal but at least one thread performing garbage collection is active. It is also likely that one or a small number of CPUs is 100% busy while all other CPUs are mostly idle.
- If the JVM in which the directory proxy server is running has hung for some reason, then the pstack utility should show that one or more threads are blocked and unable to make progress. In such cases, the system CPUs should be mostly idle.
- If a network or firewall configuration problem arises, then attempts to communicate with the server cannot be received by the server. In that case, a network sniffer like snoop or tcpdump should show that packets sent to the system on which the directory proxy server is running are not receiving TCP acknowledgement.
- If the system on which the directory proxy server is running has become hung or lost power with a graceful shutdown, then the behavior is often similar to that of a network or firewall configuration problem.

If it appears that the problem is with the directory proxy server software or the JVM in which it is running, then you need to work with UnboundID or your authorized support provider to fully diagnose the problem and determine the best course of action to correct it.

## The Server Is Slow to Respond to Client Requests

If the directory proxy server is running and does respond to clients, but clients take a long time to receive responses, then the problem can be attributable to one of the following reasons:

- The server is not optimally configured for the type of requests being processed, or clients are requesting inefficient operations. If this is the case, then the access log should show that operations are taking a long time to complete and they will likely be unindexed. This could result from searches that are unindexed in the backend servers. In an entry-balancing environment, there may be no global attribute index for the target attribute, which could cause all of the searches to be broadcasts. Updating the server configuration to better suit the requests, or altering the requests to make them more efficient, could help alleviate the problem.
- The server is overwhelmed with client requests and has amassed a large backlog of requests in the work queue. This can be the result of a configuration problem (for example, too few worker threads configured), or it can be necessary to provision more systems on which to run the directory proxy server software. Symptoms of this problem appear similar to those experienced when the server is asked to process inefficient requests, but looking at the details of the requests in the access log show that they are not necessarily inefficient requests.
- The global index is not configured to fully cache all of the data in the server, or the cache is not yet primed. In this case, the global index cache is loaded from other systems over the

network, resulting in high network load or an unusually high number of broadcasts. This can be resolved by configuring the server to fully cache all index data, and to load database contents into memory on startup. If the underlying system does not have enough memory to fully cache the entire data set, then it might not be possible to achieve optimal performance for operations that need data which is not contained in the cache.

- If the JVM is not properly configured, then it will need to perform frequent garbage collection and periodically pause execution of the Java code that it is running. In that case, the server error log should report that the server has detected a number of pauses and can include tuning recommendations to help alleviate the problem.
- If the directory proxy server is configured to use a large percentage of the memory in the system, then it is possible that the system has gotten low on available memory and has begun swapping. In this case, `iostat` should report very high utilization for disks used to hold swap space, and commands like “`swap -l`” on Solaris or “`cat /proc/meminfo`” on Linux can report a large amount of swap memory in use.
- If another process on the system is consuming a significant amount of CPU time, then it can adversely impact the ability of the directory proxy server to process requests efficiently. Isolating the processes (for example, using processor sets) or separating them onto different systems can help eliminate this problem.

## The Server Returns Error Responses to Client Requests

If a large number of client requests are receiving error responses, then potential reasons include:

- If clients are requesting operations that legitimately should fail (for example, they are targeting entries that do not exist, are attempting to update entries in a way that would violate the server schema, or are performing some other type of inappropriate operation), then the problem is likely with the client and not the server.
- If the directory proxy server work queue is configured with a maximum capacity and that capacity has been reached, then the server begins rejecting all new requests until space is available in the work queue. In this case, it might be necessary to alter the server configuration or the client requests or both, so that they can be processed more frequently, or it might be necessary to add additional server instances to handle some of the workload.
- If an internal error occurs within the server while processing a client request, then the server terminate the connection to the client and logs a message about the problem that occurred. This should not happen under normal circumstances, so it is likely necessary to work with UnboundID or your authorized support provider to diagnose and correct the problem.

## The Server Routes Requests in an Unexpected Manner

If the server routes requests in an unexpected way, the directory proxy server’s global index may be out of date. Reload the global index using the `reload-index` tool. You can reload all

of the configured indexes in the global index or just a specific index. For example, you can reload the global index as follows:

```
reload-index --bindPassword password \  
--baseDN "dc=example,dc=com"
```

## Problems with the Directory Proxy Server Management Console

If a problem arises when trying to use the directory proxy server management console, then potential reasons for the problem may include the following:

- The web application container used to host the console is not running. If an error occurs while trying to start it, then consult the logs for the web application container.
- If a problem occurs while trying to authenticate to the web application container, then make sure that the target directory proxy server is online. If it is online, then the access log may provide information about the reasons for the authentication failure.
- If a problem occurs while attempting to interact with a directory proxy server instance using the Directory Proxy Server Management Console, then the access and error logs for that directory proxy server instance might provide additional information about the underlying problem.

## Global Index Growing Too Large

If the global index appears to be growing too large, you can reload from the backend directory servers. Use the `--fromDS` option, overriding the configuration of the `prime-index-source` property. You can do this on a one off basis if the global index appears to be growing too large, using the `reload-index` tool as follows:

```
reload-index --bindPassword password --baseDN "dc=example,dc=com" \  
--fromDS
```

## Forgotten Proxy User Password

If you have forgotten the password you set for the `cn=Proxy User` entry, you can work around the problem as follows:

- You can temporarily add a second password to the proxy user entry so that you can transition all of the directory proxy server instances to the new password. However, you should have multiple passwords on the `cn=Proxy User` entry for the shortest time possible.
- If you do not know the clear-text value, then you can use the encrypted value when configuring the new Directory Proxy Server. The encryption scheme allows reversible passwords that are stored in the server configuration so that they can be decrypted by any server instance.



- You can create a new root user in the directory server instances with the appropriate set of privileges and have the new directory proxy server instance use that account to authenticate. Since it is not a good idea to have an account for which you do not know the password, you may want to update all of the other directory proxy server instances to use the new account.
- You can use a protocol analyzer like snoop or Wireshark, or a tool like the LDAPDecoder provided with SLAMD, to capture the password from the network communication. When used in proxy mode, the SLAMD LDAPDecoder tool can handle SSL-encrypted communication, and any of these methods can handle clear-text LDAP.

## Providing Information for Support Cases

If a problem arises that you are unable to fully diagnose and correct on your own, then contact UnboundID or your authorized support provider for assistance. In order to ensure that the problem can be addressed as quickly as possible, you should run the `collect-support-data` tool in the `bin` directory to provide all of the information that the support personnel may need to fully understand the underlying cause. The tool aggregates all relevant support files into a zip file that administrators can send to UnboundID for analysis. For more information about this tool, see “Working with the Collect Support Data Tool” on page 288.



# Index

## A

### access control

#### authmethod keyword

described 174

none 174

sasl 174

simple 174

ssl 174

#### bind rule keywords

authmethod 176

dayofweek 176

dns 176

groupdn 176

ip 177

timeofday 177

userattr 177

userdn 177

#### entry balancing 168

#### global aci

described 170

#### key features 170

#### permissions

add 175

all 176

compare 175

delete 175

proxy 175

read 175

search 175

selfwrite 175

write 176

#### target keywords

extop 175

targattrfilters 175

target 175

targetattr 175

targetcontrol 175

targetfilter 175

targetscope 175

#### targeted attributes

differences between  
user and opera-  
tional 171

#### targetscope keyword

base 173

described 172

onelevel 173

subordinate 173

subtree 173

#### access control with entry bal- ancing 168

#### access log 105

#### account status notification han- dler

associating with password  
policies 143

described 140

Error log 141

disabling 141

removing notification  
type 142

#### events

account-disabled 141

account-enabled 141

account-expired 141

account-idle-locked 141

account-permanently-  
locked 141

account-reset-locked  
141

account-temporarily-

locked 141

account-unlocked 141

password-changed 141

password-expired 141

password-expiring 141

password-reset 141

#### SMTP 141

configuring 142

described 142

enabling 143

viewing 143

#### alert handlers

described 144

#### JMX

configuring 145

described 144

#### SMTP

configuring 145

described 144

#### SNMP

configuring 146

described 144

#### attribute mapping

configuring 82

deployment example 240

#### audit log 105

## C

#### client connection policy

configure with dsconfig 97

configure with the console  
97

configuring 92

definition of 15

deployment example 95,

- 269
  - setting operation rate with 94
  - setting resource limits with 94
- collecting support data 288
- collect-support-data
  - options 288
- config audit log 104
- configuration overview 18
- configuring
  - client connection policies 92
  - directory proxy server 47
  - entry balancing ACIs 168
  - health checks 71
  - LDAP external servers 76
  - load balancing 80
  - locations 54
  - proxy transformations 82
  - request processors 84
  - subtree views 90
- connection handlers
  - JMX
    - configuring 144
- create-initial-proxy-config script 48
  - deployment example 199
- create-initial-proxy-config tool 48
  - entry balancing deployment 51
  - standard deployment 49
- custom health checks 72
- D
- data collectors 288
- debug log 103
- deployment example 197
  - merging two data sets 240
  - three data centers 210
  - two data centers 197
- detaching a replication set 193
- Directory Management Console 41
  - installing 42–44
  - logging in 44–45
    - creating a global admin-istrator 44
  - uninstalling 46
  - upgrading 46
- directory management console
  - troubleshooting 312
- directory proxy server
  - component architecture of 15
  - configuration steps 18
  - configuring 47
  - features of 7
  - installing 28
  - managing 101
  - monitoring 115
  - scheduled shutdown of 37
  - starting 36
  - stopping 37
  - testing configuration of 208
  - troubleshooting 287
- directory server
  - configuring
    - using dsconfig utility 54
- DN mapping
  - configuring 82
  - deployment example 240
  - multiple source DN's 241
- dsconfig utility
  - complexity menu 54
    - Advanced 55
    - Basic 54
    - changing 55
    - Expert 55
    - Standard 54
  - using batch mode 57
  - using interactive command-line mode 54
  - using non-interactive command-line mode 56
- dsframework utility
  - creating a global administrator 44
- dsreplication tool 185
- E
- embedded profiler 292
- entry balancing
  - access control 168
  - component architecture of 16
  - configuring 51, 98
  - definition of 15
  - global index size 111
  - overview 99
  - placement algorithm 99
  - replication and 183
- entry balancing monitor entry 118
- entry balancing request processor 84
- entry counter placement algorithm 99
- error log 102
- F
- file descriptor limit
  - editing /etc/security/limits.conf 26
  - editing /etc/sysctl.conf 26
  - increasing 26
  - setting 25
  - using ulimit 26
  - verifying epoll resource limit 27
- filesystem flushes
  - setting 27
- forgotten proxy user password 312
- fromDS option 113
- G
- global administrator
  - creating 59
  - described 59
  - removing 59–60
- global index 15, 111, 114
  - managing 110
  - monitoring size of 111
  - priming 113
  - troubleshooting 311
- global index priming and 114
- H
- hash DN placement algorithm 99
- health checks

- configuring 71
  - custom 72
  - default 72
  - deployment example 221
- histogram
  - accessing 119
- I
- installation 28
  - interactive mode 28, 33
  - non-interactive mode 31, 36
- J
- Java 6 22
  - installing
    - onLinux 24
    - onWindows 24
- Java troubleshooting 293
- JConsole
  - monitoring with 120
- JMX
  - monitoring with 119
- L
- large global index 111
- LDAP
  - monitoring over 121
- LDAP external servers
  - configuring 76
  - definition of 9
  - deployment example 202
  - monitoring entry for 118
  - testing failure of 210
- LDAP health check
  - definition of 10
- LDIF connection handler 292
- LD\_PRELOAD 27
- LD\_PRELOAD\_32 27
- LD\_PRELOAD\_64 27
- libumem.so 27
- load-balancing algorithm
  - configuring 80
  - definition of 11
  - deployment example 197, 234
- locations
  - configuring 54
  - definition of 9
  - deployment example 201
- log publishers 107
  - creating new 108
  - log retention 110
  - log rotation 109
  - logs
    - default 101
    - managing 101
    - periodic stats 133
    - publishers of 107
- M
- manage-account utility
  - described 155
  - enabling 156
  - getting an account state 155
  - returning password policy 156
- manage-account utility
  - disabling 156
- monitor entries 117
  - entry balancing 118
  - LDAP external server 118
- monitoring
  - size of global index 111
- monitoring directory proxy server 115
  - monitor entries 117
  - over LDAP 121
  - status tool 116
  - using SNMP 121
  - with JConsole 120
  - with JMX 119
- monitoring interfaces 118
- N
- non-interactive mode 39
- O
- operating system
  - troubleshooting 297
- operation rate
  - defining 94
- OS-level environment variables
  - editing 27
- P
- password
  - forgotten 312
- password policies
  - assigning to a group 153–154
  - assigning to an account 153
- bypassing evaluation 157
- bypassing per user 157
- creating 152
- default properties
  - allow-user-password-changes 151
  - force-change-on-add 151
  - force-change-on-reset 151
  - lockout-failure-count 151
  - max-password-age 151
  - password-change-requires-current-password 151
  - password-expiration-warning-interval 151
  - password-history-count 151
- deleting 154
- described 149
- globally bypassing evaluation 157
- modifying 152
- preventing user modification 153
- privileges
  - bypass-pw-policy 157
- properties
  - disable-password-policy-evaluation 157
- viewing 150
- viewing specific 150
- password validators
  - Attribute Value
    - described 158
    - managing 159
  - Character Set
    - described 158
    - managing 160
  - Dictionary
    - described 158
    - managing 160
  - Length-Based

- managing 161
- Length-Based Password Validator
  - described 158
- Regular Expression
  - described 158
- Repeated Character
  - managing 162, 163
- Repeated Characters
  - described 158
- Similarity-Based
  - managing 164
- Similarity-Based Password Validator
  - described 158
- Unique Characters
  - described 158
  - managing 165
- viewing 159
- passwords
  - generating a temporary password 155
  - modifying 155
- Periodic Stats Logger
  - configuring custom 135
  - configuring multiple 135
  - description 133
  - enabling 134
- placement algorithms 99
  - entry counter 99
  - hash DN 99
  - round-robin 99
  - single set 100
- prepare-external-server tool 79
  - deployment example 211
- prime-index-source property 113
- priming from Sun Directory Server 114
- priming global index 113
- priocntl 27
- privileges
  - assigning 181
  - backend-backup 178
  - backend-restore 178
  - bypass-acl 178
  - bypass-pw-policy 178
  - bypass-read-acl 178
  - config-read 178
  - config-write 179
  - described 177
  - disabling 181
  - disconnect-client 179
  - granted to root users 180
  - jmx-notify 179
  - jmx-read 179
  - jmx-write 179
  - ldif-export 179
  - ldif-import 179
  - lockdown-mode 179
  - modify-acl 179
  - password-reset 179
  - privilege-change 179
  - proxied-auth 179
  - server-restart 179
  - server-shutdown 179
  - stream-values 180
  - unindexed-search 180
  - update-schema 180
- proxy transformations
  - attribute mapping 82
  - configuring 82
  - definition of 12
  - deployment example 240
  - DN mapping 82
  - simple to external bind 83
  - suppress attribute 82
  - suppress entry 82
  - testing 273
- proxying request processor 84
- pstack 27
- R
  - reload-index tool 111
    - fromDS option 111
  - reloading 111
- replication
  - checking status of 185
  - configuration example 186
  - detaching 193
  - detaching a replica 193
  - enabling 192
  - prerequisites 184
  - with entry balancing 183
- request processors
  - configuring 84
  - definition of 13
  - deployment example 262
  - entry balancing 84
  - proxying 84
- resource limits
  - setting 94, 114
- restricted argument 185
- revert-update utility
  - reverting an update 41
- root user
  - default privileges 64
  - modifying passwords 65
  - viewing default privileges 65
- root users
  - described 63
- rotating logs 109
- round-robin placement algorithm 99
- S
  - scheduling a shutdown 37
- scripts
  - collect-support-data 290
  - create-initial-proxy-config 48
  - prepare-external-server 79
  - profile-viewer 293
  - status 116
- server crashed 308
- server does not run 303
- server groups
  - adding an instance to a group 61
  - applying changes to an instance 62
  - applying changes to the group 61
  - creating 60–63
  - described 60
  - setting the global configuration property 61
- server health checks 71
- server not accepting client con-

- nections 308
  - server returns errors 311
  - server unresponsive 309
  - server version 291
  - server.out log 103
  - setting resource limits 94, 114
  - setup log 107
  - simple to external bind 83
  - single set placement algorithm 100
  - SNMP
    - configuring 123
    - configuring 125
    - context names 123
    - directory server 122
    - MIBs 125
      - references 126
    - overview of 121
    - traps 130
  - starting directory proxy server 36
    - at boot 37
    - background process 36
    - foreground 36
  - status tool 116
  - stopping directory proxy server 37
  - subtree view
    - definition of 14
    - deployment example 267
  - subtree views
    - configuring 90
  - Sun Directory Server 114
  - Suppress attribute 82
  - Suppress entry 82
  - systat 27
  - T
  - testing proxy configuration 208
  - testing proxy transformations 273
  - testing simulated server failure 210
  - tool log 107
  - troubleshooting directory proxy server 287
  - troubleshooting tools 287
- U
  - uninstalling directory proxy server 39
    - interactive mode 38
  - update utility
    - updating 40–41
  - user accounts
    - described 155
    - disabling 156
    - enabling 156
    - getting an account state 155
    - returning password policy 156
  - V
  - virtual attributes 146
  - W
  - web console. See Directory Management Console

