
NetIQ® LDAP Proxy 1.5

Administration Guide

May 2019

Legal Notice

For information about legal notices, trademarks, disclaimers, warranties, export and other use restrictions, U.S. Government rights, patent policy, and FIPS compliance, see <https://www.netiq.com/company/legal/>.

Copyright © 2016 NetIQ Corporation, a Micro Focus company. All Rights Reserved.

Contents

About this Book and the Library	5
About NetIQ Corporation	7
1 Introduction	9
How LDAP Proxy Solves Business Challenges	9
How LDAP Proxy Works	10
LDAP Proxy Components and Their Features	11
LDAP Proxy	11
LDAP Proxy Manager	14
The LDAP Proxy Environment	15
2 Configuring NetIQ LDAP Proxy	17
Setting Up a Basic LDAP Proxy Configuration	17
Creating a New Configuration File	18
Configuring Secured Communication Using TLS Parameters	21
Configuring Certificate Information	21
Configuring Ciphers	23
Configuring Protocols	24
Configuring Listeners	25
Configuration Parameters for Listeners	26
Configuring Listener on a Secure Port	27
Configuring Listener on a Non-Secure Port	28
Configuring Additional Listeners	28
Examples	30
Configuring Back-End Servers	31
Configuration Parameters For Back-End Server	32
Configuring Back-End Server on a Secure Port	33
Configuring Back-End Server on a Non-Secure Port	34
Configuring Additional Back-End Servers	35
Examples	37
Configuring Server Groups	38
Configuring Policies	42
Adding a New Policy	43
Generic Configuration Parameters of Policy	43
Use Cases with Examples	52
Handling Attribute OIDs in Policies	69
Enabling Auditing	69
Configuring Proxy Paths	69
Configuring Audit Events Using XDAS	70
Configuring Audit Events	78
Configuring Logging	81
Configuring the Redis Server	81
Validating Configuration File	82

3	Monitoring LDAP Proxy Processes	83
4	Configuring Monitoring and Trending Activities	85
	Configuring Monitoring Activities	85
	Managing Trend Analysis	88
	Enabling Monitoring and Trending	92
	Restoring Monitoring and Trending Configuration	95
	Restoring Trending Configuration	95
	Restoring Monitoring Configuration	96
5	Maintaining LDAP Proxy	99
	Starting LDAP Proxy	99
	Stopping LDAP Proxy	99
	Restarting LDAP Proxy	99
	Checking the Status of LDAP Proxy	99
	Backing Up the LDAP Proxy	100
A	Resolving Error	101
B	Sample Configurations	107
	Sample Entries	107
	Sample XML Files and XML Formatting	108

About this Book and the Library

The *NetIQ LDAP Proxy Administration Guide* provides an overview of NetIQ LDAP Proxy and its administration. It also describes how to configure the monitoring, analyzing, querying, and modifying directory services by using NetIQ LDAP Proxy.

Intended Audience

This guide is intended for network administrators.

Other Information in the Library

The library provides the following information resources:

Installation Guide

Describes how to install LDAP Proxy. It is intended for network administrators.

These guides are available at [NetIQ eDirectory 9.0 documentation Web site \(https://www.netiq.com/documentation/ldaproxy\)](https://www.netiq.com/documentation/ldaproxy).

About NetIQ Corporation

We are a global, enterprise software company, with a focus on the three persistent challenges in your environment: Change, complexity and risk—and how we can help you control them.

Our Viewpoint

Adapting to change and managing complexity and risk are nothing new

In fact, of all the challenges you face, these are perhaps the most prominent variables that deny you the control you need to securely measure, monitor, and manage your physical, virtual, and cloud computing environments.

Enabling critical business services, better and faster

We believe that providing as much control as possible to IT organizations is the only way to enable timelier and cost effective delivery of services. Persistent pressures like change and complexity will only continue to increase as organizations continue to change and the technologies needed to manage them become inherently more complex.

Our Philosophy

Selling intelligent solutions, not just software

In order to provide reliable control, we first make sure we understand the real-world scenarios in which IT organizations like yours operate — day in and day out. That's the only way we can develop practical, intelligent IT solutions that successfully yield proven, measurable results. And that's so much more rewarding than simply selling software.

Driving your success is our passion

We place your success at the heart of how we do business. From product inception to deployment, we understand that you need IT solutions that work well and integrate seamlessly with your existing investments; you need ongoing support and training post-deployment; and you need someone that is truly easy to work with — for a change. Ultimately, when you succeed, we all succeed.

Our Solutions

- ◆ Identity & Access Governance
- ◆ Access Management
- ◆ Security Management
- ◆ Systems & Application Management
- ◆ Workload Management
- ◆ Service Management

Contacting Sales Support

For questions about products, pricing, and capabilities, contact your local partner. If you cannot contact your partner, contact our Sales Support team.

Worldwide:	www.netiq.com/about_netiq/officelocations.asp
United States and Canada:	1-888-323-6768
Email:	info@netiq.com
Web Site:	www.netiq.com

Contacting Technical Support

For specific product issues, contact our Technical Support team.

Worldwide:	www.netiq.com/support/contactinfo.asp
North and South America:	1-713-418-5555
Europe, Middle East, and Africa:	+353 (0) 91-782 677
Email:	support@netiq.com
Web Site:	www.netiq.com/support

Contacting Documentation Support

Our goal is to provide documentation that meets your needs. If you have suggestions for improvements, click **comment on this topic** at the bottom of any page in the HTML versions of the documentation posted at www.netiq.com/documentation. You can also email Documentation-Feedback@netiq.com. We value your input and look forward to hearing from you.

Contacting the Online User Community

Qmunity, the NetIQ online community, is a collaborative network connecting you to your peers and NetIQ experts. By providing more immediate information, useful links to helpful resources, and access to NetIQ experts, Qmunity helps ensure you are mastering the knowledge you need to realize the full potential of IT investments upon which you rely. For more information, visit <http://community.netiq.com>.

1 Introduction

NetIQ LDAP Proxy is a powerful application that acts as a middleware layer between LDAP clients and LDAP directory servers. The benefits of using this proxy server include enhanced security, scalability, high availability, and direct access control to directory services. It provides features such as load balancing, failover, query filtering, data hiding, request denial, centralized auditing and monitoring, and graphical trending of LDAP activities.

- ♦ [“How LDAP Proxy Solves Business Challenges” on page 9](#)
- ♦ [“How LDAP Proxy Works” on page 10](#)
- ♦ [“LDAP Proxy Components and Their Features” on page 11](#)
- ♦ [“The LDAP Proxy Environment” on page 15](#)

How LDAP Proxy Solves Business Challenges

The LDAP Proxy server solves significant business challenges for your system:

- ♦ **High availability of back-end servers:** LDAP Proxy provides dynamic load balancing and automatic failover capabilities that ensure high availability and scalability of the directory infrastructure.
- ♦ **Enhanced security:** LDAP Proxy acts as a directory firewall by using flexible network restriction policies. These policies control the connections based on the network identity of the client application. LDAP Proxy also protects the directory infrastructure from end users.
- ♦ **Enhanced access control:** LDAP Proxy provides flexible and extensible identity-based policies. The identity can be grouped by the client's network, LDAP Bind DN, LDAP Bind DN container, and proxy listener interface. Additionally, you can have granular control over various aspects for all users or a specific set of users, including:
 - ♦ Routing connections to a specific back-end server group
 - ♦ Denying certain requests such as subtree searches with a (cn=*) filter, or allowing read-only access
 - ♦ Re-encoding requests to enforce a search time limit or size limit
 - ♦ Hiding containers and blocking certain attributes
- ♦ **Centralized auditing and live monitoring:** LDAP Proxy acts as a single point of auditing and eliminates costly back-end auditing of directory servers. Centralized live monitoring helps to generate a graphical view of the ongoing activities at the proxy server and back-end directory servers. It helps to detect potential problems before they arise, so that you can take appropriate measures. Regardless of the vendor or version of the back-end servers, you can use the same auditing and monitoring solution.
- ♦ **Graphical trend analysis:** LDAP Proxy provides a graphical view of trend data such as network traffic, load, and performance. This helps to analyze and fine-tune directory infrastructure.
- ♦ **Schema mapping:** LDAP Proxy provides schema compatibility that helps applications to work with any LDAP directory. Furthermore, schema mapping enables you to have multiple views of the same Directory Information Tree, based on identity. Therefore, applications do not need to change when the directory infrastructure changes.

- ♦ **Data consistency:** LDAP Proxy allows access to the latest directory data regardless of the distributed nature of a directory infrastructure. This is achieved by using the `request-route-dit` attribute. For more information, see [“request-route-dit:” on page 64](#).

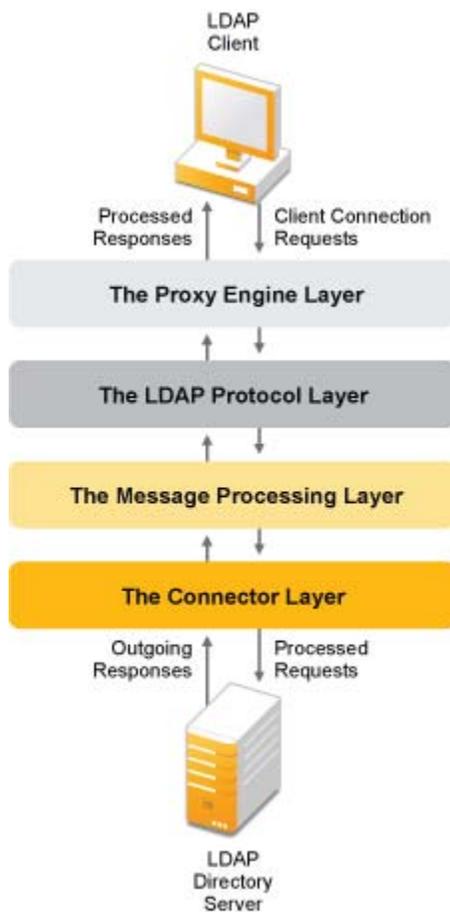
How LDAP Proxy Works

NetIQ LDAP Proxy is designed to analyze the network traffic from various interfaces and regulate requests and responses among LDAP server directories, based on policies.

Architecture

The high-level architecture of NetIQ LDAP Proxy is made up of four layers: the proxy engine layer, protocol handler layer, message processing layer, and connector layer. Each of these layers is designed to supply certain functionality for the proxy, and the architecture is extensible to allow easy adoption of new protocols and different back-end stores such as databases.

Figure 1-1 High-Level Architecture of NetIQ LDAP Proxy



1. **The Proxy Engine Layer:** A protocol-independent layer that performs several tasks:
 - ♦ Listens for client connection requests. LDAP Proxy can listen on multiple interfaces.
 - ♦ Acts as the directory firewall and filters traffic by allowing only trusted networks to establish a connection.

- ◆ Creates and monitors a session for each accepted connection.
 - ◆ Schedules and handles the connection requests.
- 2. The Protocol Handler Layer:** A protocol-specific layer. After a client connection is established, all incoming requests and outgoing responses are passed on to the protocol layer. This layer performs the following tasks:
- ◆ Decodes the LDAP requests.
 - ◆ Executes the Connection Route policy to determine the identity group of the client on every first request and subsequent Bind request. Based on the identity group, the policy determines the policies to be applied and identifies the back-end server group to which the connection needs to be routed.
 - ◆ Dispatches the incoming LDAP requests to the underlying message processing layer.
 - ◆ Receives the LDAP responses from the message processing layer and forwards the response to the LDAP client.
 - ◆ Collects monitoring statistics for proxy listeners.
- 3. The Message Processing Layer:** An optional layer that evaluates the policies. This layer performs the following tasks:
- ◆ Receives the requests from the protocol layer and responses from the connector layer.
 - ◆ Evaluates the associated policy for an incoming request/outgoing response.
 - ◆ Dispatches the requests and responses to the next level based on the policy defined.
- 4. The Connector Layer:** Acts as an interface that forwards the processed requests to the appropriate directory server. This layer performs the following tasks:
- ◆ Receives the requests from the message processing layer and forwards them to the appropriate back-end server.
 - ◆ Provides load balancing and failover.
 - ◆ Chains the requests if a referral response is received.
 - ◆ Decodes the LDAP responses received from the back-end server and dispatches the responses to the message processing layer.
 - ◆ Collects monitoring statistics for back-end servers.
 - ◆ Provides connection pooling to enhance performance.

LDAP Proxy Components and Their Features

There are several components behind the functionality and design of the NetIQ LDAP Proxy and LDAP directory servers.

- ◆ [“LDAP Proxy” on page 11](#)
- ◆ [“LDAP Proxy Manager” on page 14](#)

LDAP Proxy

- ◆ [“Listener” on page 12](#)
- ◆ [“Back-End Server” on page 12](#)
- ◆ [“Back-End Server Group” on page 12](#)
- ◆ [“Policy” on page 12](#)

Listener

A listener is the network interface where the LDAP Proxy listens for incoming requests. The proxy is capable of listening on multiple interfaces, and any number of listeners can be configured for LDAP Proxy.

Back-End Server

A back-end server is a directory server to which LDAP Proxy is connected. The proxy intercepts the requests to the back-end servers and processes the requests based on certain policies, and then forwards the requests to the back-end servers.

Back-End Server Group

The back-end servers that are configured for LDAP Proxy must be grouped as server groups. A server group is made up of one or more back-end servers to which the proxy sends requests. All the servers in a server group must host the same tree view.

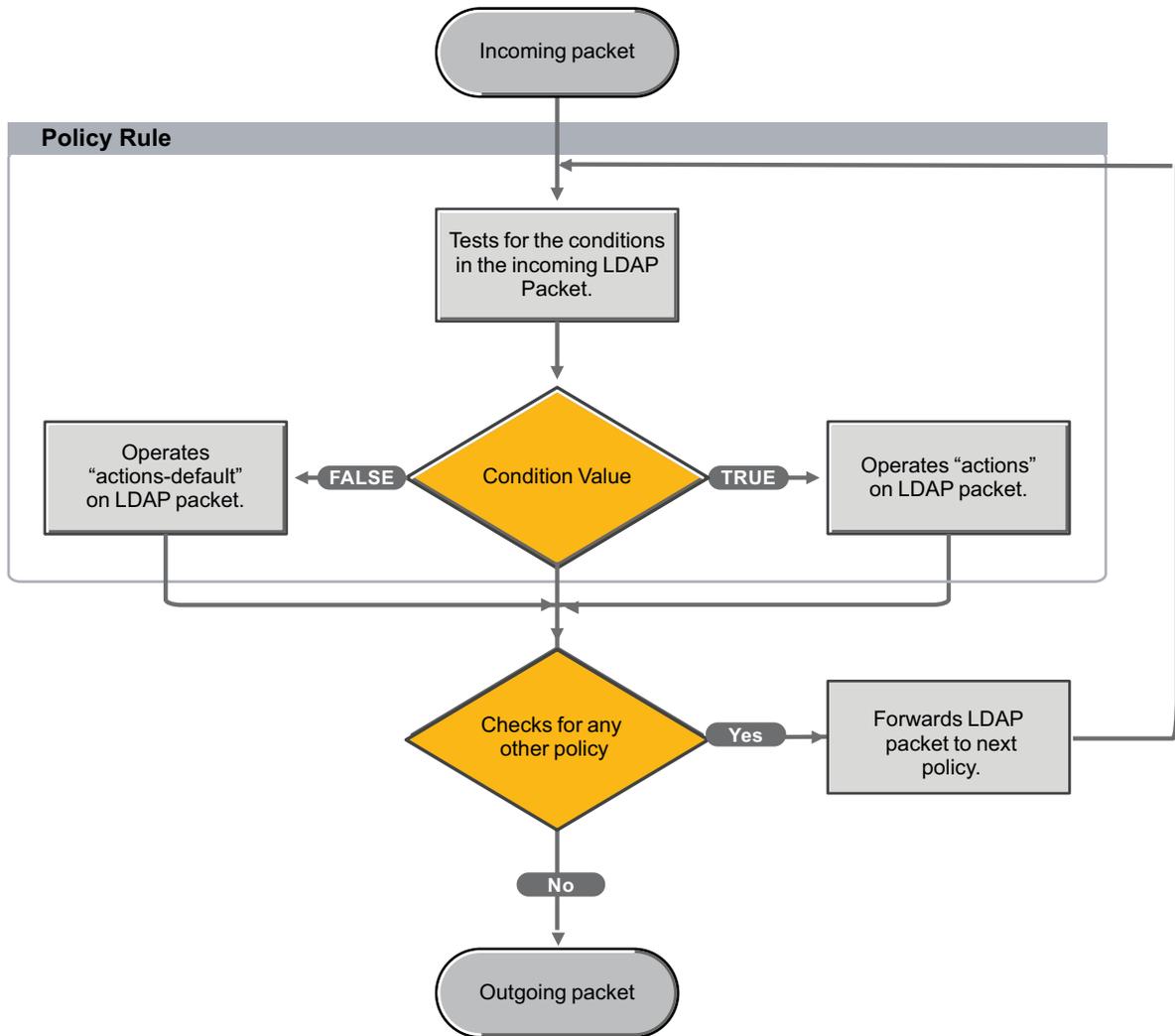
Policy

A policy is a rule that contains a set of conditions that are evaluated and the actions that are performed when the condition is true or false.

The policies that can be configured for LDAP Proxy enable the proxy to analyze and act on the incoming requests and outgoing responses, based on the rules defined when the proxy was configured. Every request or response is sequentially passed to and processed by all the policies defined.

[Figure 1-2](#) illustrates how a request is processed by policies.

Figure 1-2 Applying Policies to Requests and Responses



Currently, NetIQ LDAP Proxy supports the following policies:

- ♦ “Client Network Policy” on page 13
- ♦ “Operation Restriction Policy” on page 14
- ♦ “Map Schema Policy” on page 14
- ♦ “Search Request Policy” on page 14
- ♦ “Connection Route Policy” on page 14
- ♦ “Replace String Policy” on page 14

Client Network Policy

The Client Network policy is an optional policy that acts as a directory firewall. Before establishing a new connection from a client, the proxy executes this policy and, based on the network parameters, the connection is either accepted or rejected.

Operation Restriction Policy

The Operation Restriction policy is an optional policy that is used to restrict certain LDAP operations. LDAP operations that can be restricted are Bind, Search, Modify, Add, Delete, Moddn, Compare, and extended requests.

Map Schema Policy

The Map Schema policy is an optional policy that is used to map the back-end server schema to the application-specific schema.

Search Request Policy

The Search Request policy is an optional policy that is used to perform specific operations based on the directory tree specified in the policy. This policy is applied to an incoming search request, and after the request is evaluated, the policy performs operations including modifying the incoming search request and denying the request.

Connection Route Policy

The Connection Route policy is a mandatory policy that is used to route an incoming request to the appropriate back-end server group. Based on the conditions specified, the proxy determines the client identity, applies associated policies, and routes the request to the server. At least one Connection Route policy must be configured.

Replace String Policy

The Replace String policy is an optional policy that is used to replace a string sequence in the attribute values of a directory.

LDAP Proxy Manager

The LDAP Proxy Manager (NLPManager) is a graphical utility that enables you to monitor, analyze, and manage LDAP events.

Using NLPManager

You use NLPManager to manage files and events:

- ♦ Manage the `nlpconf.xml` configuration file used by NetIQ LDAP Proxy and configure the proxy according to your requirement.
- ♦ Create a new XML configuration file and configure LDAP Proxy. For more information, refer to [“Creating a New Configuration File” on page 18](#).
- ♦ Configure the events to be monitored. For more information, refer to [“Configuring Monitoring Activities” on page 85](#).
- ♦ Manage the LDAP events for trend analysis. For more information, refer to [“Managing Trend Analysis” on page 88](#).

NOTE: NLPManager does not allow you to configure policies for LDAP Proxy.

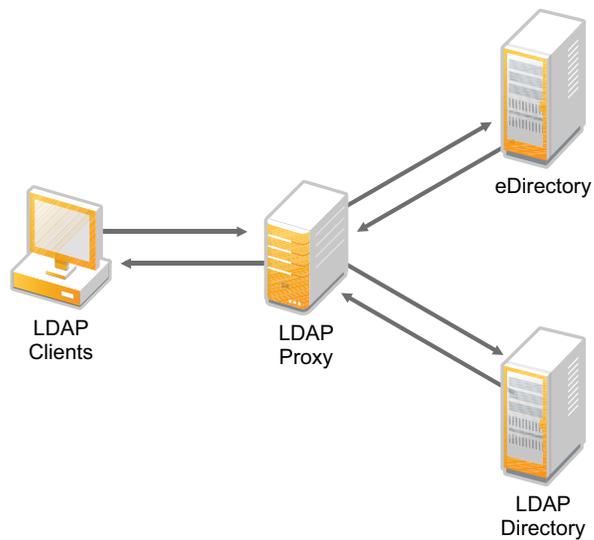
The LDAP Proxy Environment

NetIQ LDAP Proxy is completely transparent and can be easily integrated with an existing directory infrastructure. It is extremely easy to deploy, manage, and customize with any LDAP directory.

The LDAP Proxy server requests for various LDAP events from the back-end directories (for example eDirectory or any other LDAP directory) and sends the events back to the LDAP Clients.

Figure 1-3 is a graphical representation of the NetIQ LDAP Proxy environment.

Figure 1-3 NetIQ LDAP Proxy Environment



2 Configuring NetIQ LDAP Proxy

This section describes how to setup a basic LDAP Proxy configuration and perform common configuration tasks.

- ♦ [“Setting Up a Basic LDAP Proxy Configuration” on page 17](#)
- ♦ [“Creating a New Configuration File” on page 18](#)
- ♦ [“Configuring Secured Communication Using TLS Parameters” on page 21](#)
- ♦ [“Configuring Listeners” on page 25](#)
- ♦ [“Configuring Back-End Servers” on page 31](#)
- ♦ [“Configuring Policies” on page 42](#)
- ♦ [“Enabling Auditing” on page 69](#)
- ♦ [“Configuring Logging” on page 81](#)
- ♦ [“Configuring the Redis Server” on page 81](#)
- ♦ [“Validating Configuration File” on page 82](#)

Setting Up a Basic LDAP Proxy Configuration

The initial setup for LDAP Proxy consists of installing LDAP Proxy files and NLPManager and setting up the listener, back-end server, and connection route policy for your directory server in the `nlpconf.xml` file. LDAP Proxy bundles a sample `nlpconf.xml` file with the installation package located in the `/etc/opt/novell/ldaproxy/conf` directory.

LDAP Proxy can be customized by configuring additional listeners, back-end servers, back-end server groups, and policies.

- ♦ **Listener:** The IP address and the port number where the proxy listens for incoming requests. By default, LDAP Proxy is configured to listen on all interfaces. However, you can customize it to listen only on specific interfaces.
- ♦ **Back-end server:** The IP address or domain name and port number of the system on which the back-end server is installed. At least one back-end server must be configured. However, if you plan to facilitate load balancing and fault tolerance, a minimum of two back-end servers must be configured.
- ♦ **Connection route policy:** Specifies where the connections are to be routed to. A minimum of one Connection Route policy must be configured. For more information, see [“Configuring a Connection Route Policy to Block Anonymous Binds” on page 66](#) in the *NetIQ LDAP Proxy 1.5 Administration Guide*.

The `<list-policy>` node in the `nlpconf.xml` file contains a sample Connection Route policy that defines where LDAP Proxy must route the incoming connections. Do not delete this node because there must be at least one Connection Route policy defined in the minimum configuration.

LDAP Proxy can be customized by configuring additional listeners, back-end servers, and back-end server groups. You can also define additional policies to customize LDAP Proxy to filter requests, map schemas etc. Optionally, you can also define the proxy paths and monitoring events. After modifying the `nlpconf.xml` file, save the file and start the `nlpd` service for the changes to take effect.

LDAP Proxy can be configured on both secure and non-secure ports to communicate with the back-end directory. You make these configuration changes for the Listener and Back-end server components in the `nlpconf.xml` file. The following sections provide specific instructions for configuring secure and non-secure connections.

Creating a New Configuration File

You can create an XML configuration file through NLPManager. However, to use the newly created file to configure NetIQ LDAP Proxy, you must name the file as `nlpconf.xml` and place it in the `/etc/opt/novell/ldapproxy/conf` directory on the machine where you installed LDAP Proxy.

- 1 Run the `./NLPManager` command to start the NLPManager.
- 2 To start a new configuration file, do one of the following:

- ◆ Click the  icon.
- ◆ In the **Provisioning** menu, click **New Configuration**.

The New LDAP Proxy Configuration Project window is displayed.

- 3 Specify the following.

- ◆ **Filename:** A name for the new configuration file.
- ◆ **Enter or select the parent folder:** The location where you want to save the configuration file.

The proxy configuration is displayed in the Project Explorer pane.

- 4 Click **Finish**.

- 5 Add listeners:

- 5a Click the **Listeners** option in the **Project Explorer** pane.

The **Listeners** tab is displayed in the Editor pane.

- 5b To add a listener, click the  icon.

The Add New Listener window is displayed.

- 5c Specify a name to identify the listener you are configuring and then click **OK**.

The name must be a unique alphanumeric value.

The listener configuration fields are displayed in the Editor pane.

- 5d Specify the following:

- ◆ **Address Type:** The address type of the interface where the listener is going to listen for requests.

To provide the IP address of the system where you installed the LDAP Proxy, select **IPv4** or **IPv6**.

To provide the domain name of the system where you installed the LDAP Proxy, select **DNS**.

- ◆ **Address:** The value of the IP address or domain name, depending on the address type you have specified.
- ◆ **Protocol:** Specify either **ldap** or **ldaps**.
- ◆ **Port:** The port number of the listener interface.

- ♦ **Certificate File Name:** The name of the certificate file, if the protocol specified is **ldaps**.

Ensure that you have placed the certificate file in the `/etc/opt/novell/ldaproxy/conf/ssl/private` directory.

5e Add more listeners, repeat [Step 5b](#) to [Step 5d](#).

5f Click **Provisioning** > **Save** to save your changes.

6 To add back-end servers:

6a Click the **Backend Servers** option in the **Project Explorer** pane.

6b To add a back-end server, click the  icon.

The Add New Backend Server window is displayed.

6c Specify a name to identify the back-end server you are configuring and click **OK**. The name must be a unique alphanumeric value.

The back-end server configuration fields are displayed in the Editor pane.

6d Specify the following configuration details:

- ♦ **Address Type:** The address type of the interface through which the directory server receives the requests from LDAP Proxy.

NOTE: All mandatory fields are marked in red.

To provide the IP address of the LDAP directory server, select **IPv4** or **IPv6**.

To provide the domain name of the LDAP directory server, select **DNS**.

- ♦ **Address:** The value of the IP address or domain name, depending on the address type you have specified.
- ♦ **Protocol:** Specify either **ldap** or **ldaps**.
If you specify the protocol as **ldaps**, it is mandatory to place the certificate file in the `conf/ssl/trustedcert` directory.
- ♦ **Port:** The port number of the interface.

The following optional fields can also be configured to enhance the performance of the back-end server:

- ♦ **Maximum Connections:** The maximum number of connections that could be handled by the back-end server.
- ♦ **Capability:** The capability of the back-end server relative to the other servers. For example, if the capability of a back-end server is 2, it can be loaded two times more than the other servers.
- ♦ **Connection Pool:** Specify if a connection pool must be created. If you select this field, then specify the pool size value in the **Start Pool Size** field.
- ♦ **Start Pool Size:** The number of connections to be created so that the connections can be reused for incoming requests. The value must always be less than the maximum connections value.
- ♦ **Use Anonymous Login:** Specify if anonymous login is required to create a connection pool. If anonymous bind is disabled on a particular server, then to nullify the connection identity you must specify the User Distinguished Name (user DN) in the associated **Bind DN** field.
- ♦ **Bind DN:** The Bind DN to be used to nullify a connection identity.
- ♦ **Health Check:** Whether a health check must be performed to detect a slow server. If you select this field, you must specify the Bind DN and Maximum Response Time.

- ♦ **Bind DN:** The User DN on which the health check must be performed.
 - ♦ **Maximum Response Time:** The maximum time within which a bind request must receive a response.
- 6e** Specify the time interval for performing a health check on all the back-end servers:
- 6e1** Click the [Health Check Interval Configuration](#) drop-down list.
 - 6e2** Specify the time interval for performing health checks to detect slow or unavailable back-end servers. By default, the value is 60.
- 6f** To add more back-end servers, follow [Step 6b](#) to [Step 6d](#).
- 6g** Click **Provisioning** > **Save** to save the changes.
- 7** To add back-end server groups:
- 7a** Click the **Backend Server Groups** option in the **Project Explorer** pane.
The **Backend Server Group** tab is displayed.
 - 7b** To add a server group, click the  icon.
The Add New Server Group window is displayed.
 - 7c** Specify a name to identify the back-end server group you are configuring and click **OK**.
The name must be a unique alphanumeric value.
The back-end server group configuration is displayed in the Editor pane.
 - 7d** Specify the following:
 - ♦ **Load Balancing:** Specify whether the type of load balancing is **Connection Based** or **Dynamic**.
 - ♦ **Selected Servers:** The back-end servers to be defined in the server group. You can use the arrow buttons to sort servers between the **Selected Servers** and **Available Servers** lists.
The back-end servers configured in a group must host the same DIT.
 - 7e** To add more server groups, repeat [Step 7b](#) to [Step 7d](#).
 - 7f** Click **Provisioning** > **Save** to save the changes.
- 8** To use this configuration file to configure LDAP Proxy:
- 8a** Rename the newly created XML file as `nlpconf.xml`.
 - 8b** Place the `nlpconf.xml` file in the `/etc/opt/novell/ldaproxy/conf` directory on the machine where you installed LDAP Proxy.
The default `nlpconf.xml` file is replaced with the newly created configuration file.

IMPORTANT: NLPManager is not recommended to configure LDAP Proxy for complex scenarios. NLPManager should only be used for monitoring the LDAP events. You must define your configuration manually in the `nlpconf.xml` file.

Configuring Secured Communication Using TLS Parameters

LDAP Proxy allows you to configure various Transport Layer Security parameters required for TLS communication.

You modify the `<tls-opts>` node in the `nlpconf.xml` file to define the certificate information, ciphers, and protocols that you plan to use for TLS communication. For example, the `<tls-opts>` node should look like this:

```
<service protocol="ldaps">
  <addr-ipv4>x.x.x.x</addr-ipv4>
  <port>636</port>
  <tls-opts>
    <certificate-file-name>CertificateFileName.pem</certificate-file-name>
    <ciphers>CipherString</ciphers>
    <protocol>ProtocolString</protocol>
  </tls-opts>
</service>
```

Example:

```
<service protocol="ldaps">
  <addr-ipv4>1.1.1.1</addr-ipv4>
  <port>636</port>
  <tls-opts>
    <certificate-file-name>servercertificate.pem</certificate-file-name>
    <ciphers>HIGH</ciphers>
    <protocol>+TLSv1.2</protocol>
  </tls-opts>
</service>
```

In the above example, certificate information is defined as `servercertificate.pem`, cipher is defined as `HIGH`, and the protocol is defined as `+TLSv1.2`. The interface is defined by IP address `1.1.1.1` and port `636`.

NOTE

- The `<tls-opts>` is an optional node. If the protocol is specified as LDAPS under the listener configuration, you must specify the certificate as indicated in this example.
- The certificate information can be defined inside the `<tls-opts>` node if you are using TLS. Otherwise, define it under the listener configuration. For more information, see [“Configuring Certificate Information” on page 21](#).

You can configure the following parameters for LDAP Proxy:

- [“Configuring Certificate Information” on page 21](#)
- [“Configuring Ciphers” on page 23](#)
- [“Configuring Protocols” on page 24](#)

Configuring Certificate Information

If you specify the protocol as LDAPS under the `<service>` tag of the `nlpconf.xml` file while configuring listeners, you must provide the certificate file information. The LDAP Proxy listener needs be configured with a X.509 certificate. This certificate should be in a PKCS#12 file along with the

private key and all the CA certificates in the chain. Also, ensure that the LDAP Proxy server's IP address or DNS name resides in the Common Name (CN) field of the certificate's subject name or in the Subject Alternative Name list. You can obtain a PKCS#12 certificate file for the LDAP Proxy listener from any of the following Certificate Authority (CA).

- ◆ eDirectory
 - Create a LDAP Proxy server certificate in eDirectory and export it using the Certificate Access plug-in of iManager. For more information, see [Understanding the Certificate Server](#) in the *eDirectory Administration Guide*.
- ◆ Active Directory Microsoft Management Console (MMC) Certificate snap-in
 - Refer to the Active Directory documentation for more information.
- ◆ Third-party CA
 - Refer to the third-party documentation for more information.

To decrypt the private key information and store the information in the local secret store, run the `nlpcert` utility.

- 1 Export the library path by using the following command:

```
. /opt/novell/ldapproxy/bin/nlppath
```

- 2 Run the `nlpcert` utility by using the following command:

```
nlpcert -i <infile.pfx> -o <outfile.pem>
```

Option	Description
-i <inputFile>	The name of the input file. The input file should be a PKCS#12 file with encrypted private key and server certificate or a .pem file generated using the <code>nlpexportcert</code> utility.
--infile=<inputFile>	
-p <password>	(Optional) The password of the private key.
--password=<password>	
-c, --convert	The option to convert the .pem files that were generated using the <code>nlpexportcert</code> utility.
-v, --version	The version of the output file generated by the <code>nlpcert</code> utility.
-o <outputFile>	The name of the output file where server certificate will be stored in .pem format.
--outfile=<outputFile>	

NOTE: While upgrading to 1.5.2 version, the `nlpcert` utility automatically converts the existing certificate files that were created by the `nlpexportcert` utility.

Examples:

- ◆ To create a .pem file for LDAP Proxy from a PKCS#12 file, run the following command:


```
nlpcert -i server_cert.pfx -o private-cert.pem
```
- ◆ To convert a .pem file created by `nlpexportcert` utility, run the following:
 - The `nlp-install` script automatically converts the existing certificates to the new format using the `nlpcert` utility during the upgrade process only if the server certificates are present in the `/etc/opt/novell/ldapproxy/conf/ssl/private` directory. Otherwise, these certificates are not automatically converted. To convert the certificates, run the following command:

```
nlpcert -i exportcert.pem -c -o server_cert.pem
```

NOTE: Regardless of whether you are creating a new certificate or converting an existing certificate, move `server_cert.pem` to `/etc/opt/novell/ldaproxy/conf/ssl/private` directory, and use `server_cert.pem` in the `<certificate-file-name>` tag of `nlpconf.xml`.

IMPORTANT: The certificate file created by `nlpcert` utility on one server cannot be used to configure LDAP Proxy on another server because the private key is stored in local secret store with the file name as the key. Also, you are not recommended to rename the `.pem` file generated by `nlpcert` utility.

Configuring Ciphers

You can configure your own list of ciphers using the OpenSSL Cipher List Format during the TLS Communication. The following are a few examples of using Cipher List Format:

- ◆ For RSA certificates: `!CAMELLIA:!DH:!SRP:!MD5:HIGH+aRSA`
- ◆ For ECDSA certificates: `HIGH+aECDSA`
- ◆ For Suite B 128-bit compliant cipher suite with ECDSA certificates: `ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256`
- ◆ For Suite B 192-bit compliant cipher suite with ECDSA certificates: `ECDHE-ECDSA-AES128-GCM-SHA256`

For more information on Cipher List Format, refer to the [OpenSSL Ciphers \(https://www.openssl.org/docs/man1.0.2/apps/ciphers.html\)](https://www.openssl.org/docs/man1.0.2/apps/ciphers.html) documentation.

Example 1: Configuring Ciphers for a Listener

```
<listener id-listener="listener1">
  <service protocol="ldaps">
    <addr-ipv4>1.1.1.1</addr-ipv4>
    <port>636</port>
    <tls-opts>
      <certificate-file-name>servercertificate.pem</certificate-file-name>
      <ciphers>HIGH</ciphers>
      <protocol>+TLSv1.2</protocol>
    </tls-opts>
  </service>
</listener>
```

In this example, listener is defined as `listener1` which uses `servercertificate.pem` certificate file and allows `HIGH` strength ciphers only. The listener accepts connections on IP address `1.1.1.1` and port `636`.

Example 2: Configuring Ciphers for a Back-End Server

```
<backend-server id-backend-server="Backend1">
  <service protocol="ldaps">
    <addr-ipv4>1.1.1.1</addr-ipv4>
    <port>636</port>
    <tls-opts>
      <ciphers>HIGH</ciphers>
      <protocol>+TLSv1.2</protocol>
    </tls-opts>
  </service>
</backend-server>
```

In this example, back-end server is defined as `Backend1` and LDAP Proxy connects to this back-end server at IP address `1.1.1.1` and port `636` using `HIGH` strength ciphers.

Configuring Protocols

LDAP Proxy gives you the flexibility to configure the list of protocols required during the TLS communication with Proxy server and LDAP server. To control the list of protocols, define the `<protocol>` tag under the `<tls-opts>` node in the `nlpconf.xml` file. You can configure the following protocol strings with LDAP Proxy:

- ◆ SSLv3
- ◆ TLSv1.0
- ◆ TLSv1.1
- ◆ TLSv1.2
- ◆ ALL

Each protocol string should be preceded by a "+" or a "-" symbol. The "+" symbol indicates that the protocol string(s) are allowed and the "-" symbol indicates that the protocol string(s) are not allowed to get configured with LDAP Proxy. If you have not configured any protocols yet, all supported protocols except `SSLv3` are allowed by default. The following table lists a few TLS protocol configurations:

Protocol Configuration	Description
+TLSv1.2	Allows only TLSv1.2
+ALL-TLSv1.0	Allows all except TLSv1.0
+ALL-TLSv1.2-TLSv1.1	Allows SSLv3 and TLSv1.0
+ALL	Allows SSLv3, TLSv1.0, TLSv1.1, TLSv1.2

Example 1: Configuring Protocols for a Listener

```
<listener id-listener="listener1">
  <service protocol="ldaps">
    <addr-ipv4>1.1.1.1</addr-ipv4>
    <port>636</port>
    <tls-opts>
      <certificate-file-name>CertificateFileName.pem</certificate-file-name>
      <ciphers>HIGH</ciphers>
      <protocol>+TLSv1.2</protocol>
    </tls-opts>
  </service>
</listener>
```

In this example, `listener1` accepts client connections only if they use TLS protocol `TLSv1.2`.

Example 2: Configuring Protocols for a Back-End Server

```
<backend-server id-backend-server="Backend1">
  <service protocol="ldaps">
    <addr-ipv4>1.1.1.1</addr-ipv4>
    <port>636</port>
    <tls-opts>
      <ciphers>ECDSA</ciphers>
      <protocol>+ALL-SSLv3</protocol>
    </tls-opts>
  </service>
</backend-server>
```

In this example, the protocol is defined as `+ALL-SSLv3` which means LDAP Proxy can connect to the back-end server using the `TLSv1.0`, `1.1` or `1.2`. During the SSL handshake LDAP Proxy and the back-end server negotiate the highest protocol version that both support.

Configuring Listeners

A listener is the network interface where the LDAP Proxy listens for incoming requests. Using proxy you can configure any number of listeners to listen on to multiple interfaces.

Each listener is made up of interface information that is a combination of an IP address and a port number or a domain name and port number. You must also provide service protocol information indicating either LDAPS or LDAP, which means that it is either a secure or clear-text interface. By default, LDAP Proxy listens on all interfaces.

The `<list-listener>` node in the configuration file lists all the listeners configured for the proxy. The additional listeners must be defined in this node.

For instance, if you want to define `listener1` to use the LDAP protocol. You also want to define the IP address as `192.168.1.1` and the port as `389`. Any request coming through this interface must be processed through a Connection Route policy identified as `<ref-policy-connection-route>conn-route-policy</ref-policy-connection-route>`. To do this, you can define your configuration as follows:

```

<list-listener>
  <listener id-listener="listener1">
    <service protocol="ldap">
      <addr-ipv4>192.168.1.1</addr-ipv4>
      <port>389</port>
    </service>
    <ref-policy-connection-route>conn-route-policy</ref-policy-connection-route>
  </listener>
</list-listener>

```

- ♦ [“Configuration Parameters for Listeners” on page 26](#)
- ♦ [“Configuring Listener on a Secure Port” on page 27](#)
- ♦ [“Configuring Listener on a Non-Secure Port” on page 28](#)
- ♦ [“Configuring Additional Listeners” on page 28](#)
- ♦ [“Examples” on page 30](#)

Configuration Parameters for Listeners

Use the following elements and attributes to define the listener configuration:

<listener>: Specifies that the element configured is a listener. This element has the following attributes:

- ♦ **id-listener**: The identity of the listener. The value of this attribute is used to refer to the listener. It must be a unique alphanumeric value, such as `listener1`.

The `<listener>` element must have the following child elements:

- ♦ **<service>**: Specifies how the listener listens for incoming requests. It must have the following attributes:
 - ♦ **protocol**: The protocol that the listener uses to listen for incoming requests. The attribute value can be `ldap` or `ldaps`. For more information, see [“Configuring Protocols” on page 24](#).

NOTE: If you specify the protocol as `ldaps`, you must specify the certificate information. Use the `<certificate-file-name>` element to specify the name of the file that contains the certificate information.

The `<service>` element can have the following child elements:

- ♦ **<addr-ipv4>/<addr-ipv6>**: The IP address of the system on which LDAP Proxy is installed.
- ♦ **<port>**: The port on which the listener listens for incoming requests.
- ♦ **<addr-dns>**: The domain name of the system on which LDAP Proxy is installed. In [Example 1](#), the value is `server1.example.com`.
- ♦ **<tls-opts>**: A configuration option to specify the Transport Layer Security (TLS) parameters when protocol is defined as LDAPS under the `<service>` tag for the listener configuration. For more information, see [“Configuring Secured Communication Using TLS Parameters” on page 21](#).
- ♦ **<ref-policy-client-network>**: Any request coming through this listener must be processed by using the Client Network policy. The term `ref` in this element indicates that this element is actually a pointer to a policy called `policy-client-network`. For more information about the Client Network policy, refer to [“Accepting or Denying a Client Connection \(Client Network Policy\)” on page 52](#).

- ♦ **<ref-policy-connection-route>**: Any request coming through this listener must be processed by using the Connection Route policy. The term `ref` in this element indicates that this element is actually a pointer to a policy of type `policy-connection-route`. For more information about the Connection Route policy, refer to “[Routing an Incoming Request to a Back-End Server Group \(Connection Route Policy\)](#)” on page 63.

The value shown in [Example 1](#) is `anonymous-policy`. It means a policy identified as the `anonymous-policy` must be applied to all requests coming through the port specified in the relevant listener configuration.

- ♦ **<certificate-file-name>**: The name of the file that contains the certificate information. If the proxy is going to listen on a secure port, you must specify certificate information.

NOTE: Ensure that you place the specified certificate file in the `/etc/opt/novell/ldaproxy/conf/ssl/private` directory. The certificate should be in the `pem` format.

For more information about how to export certificate file information, refer to “[Configuring Certificate Information](#)” on page 21.

Configuring Listener on a Secure Port

LDAP Proxy can be configured to use the TLS protocol to provide data privacy and integrity between the Proxy server and the LDAP client and back-end server. This configuration ensures data confidentiality and integrity protection.

To configure a listener on a secure port, perform the following steps:

- 1 Configure the LDAP Proxy listener with a `x.509` certificate.

The certificate should reside in a `PKCS#12` file along with the private key and all the CA certificates in the chain. Also, ensure that the LDAP Proxy server's IP address or DNS name is present in the Common Name (CN) field of the certificate's subject name or in the Subject Alternative Name list. You can obtain a `PKCS#12` certificate file for the LDAP Proxy listener from any of the following Certificate Authority (CA):

- ♦ eDirectory

Create a LDAP Proxy server certificate in eDirectory and export it using the Certificate Access plug-in of iManager. For more information, see [Understanding the Certificate Server](#) in the *eDirectory Administration Guide*.

- ♦ Active Directory Microsoft Management Console (MMC) Certificate snap-in

Refer to the Active Directory documentation for more information.

- ♦ Third-party CA

Refer to the third-party documentation for more information.

- 2 Export the library path by using the following command:

```
. /opt/novell/ldaproxy/bin/nlppath
```

- 3 Import the certificates and keys present in the `PKCS#12` files by using the `nlpcert` utility:

```
nlpcert -i server_cert.pfx -o private-cert.pem
```

In this example, `server_cert.pfx` is the downloaded `PKCS#12` certificate file and `private-cert.pem` is the converted `pem` certificate file.

- 4 Copy the `private-cert.pem` to `/etc/opt/novell/ldaproxy/conf/ssl/private` directory.
- 5 Specify the `private-cert.pem` in the `<certificate-file-name>` tag of `nlpconf.xml`.

Ensure that you define the protocol as `ldaps` under the `<service protocol>` tag as shown below:

```
<listener id-listener="listener1">
  <service protocol="ldaps">
    <addr-ipv4>0.0.0.0</addr-ipv4>
    <port>636</port>
  </service>
  <certificate-file-name>private-cert.pem</certificate-file-name>
  <ref-policy-connection-route>conn-route-policy</ref-policy-connection-route>
</listener>
```

- 6 Save the `nlpconf.xml` file.

Configuring Listener on a Non-Secure Port

To configure a listener on a non-secure port, complete the following steps:

- 1 Open the `nlpconf.xml` file from the `/etc/opt/novell/ldapproxy/conf` directory.
- 2 Verify that the listener configuration is defined as shown below:

```
<listener id-listener="listener1">
  <service protocol="ldap">
    <addr-ipv4>0.0.0.0</addr-ipv4>
    <port>389</port>
  </service>
  <ref-policy-connection-route>conn-route-policy</ref-policy-connection-route>
</listener>
```

NOTE: Ensure that you define the protocol as `ldap` under the `<service protocol>` tag.

- 3 Save the `nlpconf.xml` file.

Configuring Additional Listeners

You can configure additional listeners using both `nlpconf.xml` file or NLPManager utility.

Manually Configuring Additional Listeners

- 1 Open the `nlpconf.xml` file from the `/etc/opt/novell/ldapproxy/conf` directory in any XML editor.
- 2 To add a listener to the existing configuration, create an instance of the following within the `<list-listener>` node:

```
<listener id-listener="listener1">
  <service protocol="ldap">
    <addr-ipv6>[2015::37]</addr-ipv6>
    <port>4489</port>
  </service>
  <ref-policy-request-route>anonymous-policy</ref-policy-request-route>
</listener>
```

- 3 Define the following in the newly created instance:
 - ◆ The name to identify the listener you are configuring.
 - ◆ Provide either the IP address or the domain name of the system on which you have installed LDAP Proxy.
 - ◆ The protocol as either LDAP or LDAPS.
 - ◆ The port number of the interface.
 - ◆ The name of the certificate file, if you specify the protocol as LDAPS.
 - ◆ The Client Network policies and Connection Route policies that must be applied to the incoming requests. Multiple Connection Route policies can be configured on the listener, based on the identity.

For information about the elements and attributes that are used to define these parameters, refer to [“Configuration Parameters for Listeners” on page 26](#).

- 4 To add more listeners, repeat [Step 2](#) to [Step 3](#).
- 5 Save the `nlpconf.xml` file.

Configuring Additional Listeners Using NLPManager

You can configure additional listeners for the LDAP Proxy configuration by using the **Listeners** tab.

- 1 Run the `./NLPManager` command to start NLPManager.
- 2 To open the `nlpconf.xml` file, in the **Provisioning** menu, click **Open Configuration**:
The Open dialog box appears.
- 3 Browse to the `nlpconf.xml` file in the `/etc/opt/novell/ldapproxy/conf` directory and click **Open**.

The `conf` directory is available on the machine where you installed LDAP Proxy.

If NLPManager and LDAP Proxy are installed on different machines, you must either save the `nlpconf.xml` file on the machine where you installed NLPManager or map a network drive to the machine where you installed the proxy.

The proxy configuration is displayed in the Project Explorer pane.

- 4 Click the **Listeners** option in the **Project Explorer pane**.
The **Listeners** tab is displayed. To change the listener configuration, change this setting.
- 5 To add a listener, click the  icon.
The Add New Listener window is displayed.
- 6 Specify a name to identify the listener you are configuring and click **OK**.
The name must be a unique alphanumeric value.
The listener configuration fields are displayed in the Editor pane.
- 7 Specify the following:

- ◆ **Address Type:** The address type of the interface where the listener must listen for requests.
To provide the IP address of the system where you installed the LDAP Proxy, select **IPv4** or **IPv6**.
To provide the domain name of the system where you installed the LDAP Proxy, select **DNS**.
- ◆ **Address:** The value of the IP address or domain name, depending on the address type you have specified.

- ◆ **Protocol:** Specify either **ldap** or **ldaps**.
- ◆ **Port:** The port number of the listener interface.
- ◆ **Certificate File Name:** The name of the certificate file, if the protocol specified is **ldaps**.

Ensure that you have placed the certificate file in the `/etc/opt/novell/ldapproxy/conf/ssl/trustedcert` directory.

- 8 To add more listeners, repeat [Step 6](#) and [Step 7](#).
- 9 To delete a listener, select the listener from the list and click the  icon.
- 10 Click **Provisioning** > **Save** to save the changes.

Examples

- ◆ [“Example 1” on page 30](#)
- ◆ [“Example 2” on page 31](#)

Example 1

```
<list-listener>
<listener id-listener="listener1">
  <service protocol="ldaps">
    <addr-ipv4>192.168.1.1</addr-ipv4>
    <port>636</port>
  </service>
  <certificate-file-name>private-cert.pem</certificate-file-name>
  <ref-policy-connection-route>admin-policy</ref-policy-connection-route>
</listener>
<listener id-listener="listener2">
  <service protocol="ldap">
    <addr-dns>server1.example.com</addr-dns>
    <port>389</port>
  </service>
  <ref-policy-connection-route>anonymous-policy</ref-policy-connection-route>
</listener>
</list-listener>
```

In Example 1, two listeners are defined as `listener1` and `listener2`. `listener1` is defined to use the `ldaps` protocol to listen for incoming request on the system. The interface is defined by IP address `192.168.1.1` and port `636`. This node also specifies that a Connection Route policy identified as `admin-policy` is to be applied to all requests coming through the specified port, and also specifies the filename of the certificate to be used by the protocol. `listener2` is defined to use the `ldap` protocol, and the interface is defined by domain name `server1.example.com` and port `389`. It also routes requests to a Connection Route policy defined as `anonymous-policy`.

Example 2

```
<list-listener>
  <listener id-listener="listener3">
    <service protocol="ldaps">
      <addr-dns>server1.example.com</addr-dns>
      <port>636</port>
    </service>
    <certificate-file-name>private-cert1.pem</certificate-file-name>
    <ref-policy-connection-route>admin-policy</ref-policy-connection-route>
  </listener>
  <listener id-listener="listener4">
    <service protocol="ldap">
      <addr-dns>server1.example.com</addr-dns>
      <port>1389</port>
    </service>
    <ref-policy-connection-route>admin-policy</ref-policy-connection-route>
  </listener>
</list-listener>
```

In Example 2, two listeners are defined as `listener3` and `listener4`. `listener3` is defined to use the `ldaps` protocol to listen for incoming request on the system. The interface is defined by domain name `server1.example.com` and port `636`. This node also specifies that a Connection Route policy identified as `admin-policy` is to be applied to all requests coming through the specified port, and also specifies the filename of the certificate to be used by the protocol. `listener4` is defined to use the `ldap` protocol, and the interface is defined by domain name `server1.example.com` and port `1389`. It also routes requests to a Connection Route policy defined as `admin-policy`.

Configuring Back-End Servers

A back-end server is a directory server to which LDAP Proxy is connected. Using the back-end server, Proxy intercepts and processes the requests based on certain policies, and then forwards the requests to the back-end servers.

To facilitate the load balancing and fault tolerance feature of LDAP Proxy, a minimum of two back-end servers must be configured to LDAP Proxy. Periodically, a health check must be performed on the directory server to identify any performance degradation.

You can configure additional back-end servers for the proxy configuration depending on your needs. The `<list-backend-server>` node in the configuration file lists all the back-end servers configured for the proxy. The additional back-end servers must be defined in this node.

For instance, if you want to define a back-end server, `Backend1`, to use the LDAP protocol. The back-end server listens on IP address `192.168.1.3` and port `389` for incoming requests. You can define the configuration as follows:

```
<list-backend-server health-check-interval-secs="7200">
  <backend-server id-backend-server="Backend1">
    <service protocol="ldap">
      <addr-ipv4>192.168.1.3</addr-ipv4>
      <port>389</port>
    </service>
  </backend-server>
</list-backend-server>
```

- ◆ [“Configuration Parameters For Back-End Server” on page 32](#)
- ◆ [“Configuring Back-End Server on a Secure Port” on page 33](#)

- ◆ “Configuring Back-End Server on a Non-Secure Port” on page 34
- ◆ “Configuring Additional Back-End Servers” on page 35
- ◆ “Examples” on page 37
- ◆ “Configuring Server Groups” on page 38

Configuration Parameters For Back-End Server

The following elements and attributes that are used to configure back-end servers:

<backend-server>: Specifies that the element configured is a back-end server. This element can have the following attributes defined:

- ◆ **id-backend-server**: The identity of the back-end server. The value of this attribute defines the server. It must be a unique alphanumeric value.
- ◆ **max-connections**: The maximum number of connections that are handled by the back-end server. This is an optional attribute.
In [Example 1](#), the attribute value is 5000. This indicates that the `Backend1` server can handle 5000 connections.
- ◆ **capability**: The capability of the back-end server relative to the other servers. In [“Example 1” on page 30](#), the capability of the back-end server `Backend 1` is 1 and the capability of the back-end server `Backend 2` is 5. In this case, `Backend 2` can be loaded five times more than the `Backend 1`. This is an optional attribute.

The `<backend-server>` element can have the following child elements:

- ◆ **<service>**: Specifies how LDAP Proxy sends requests to the back-end server. It must have the following attributes:
 - ◆ **protocol**: The protocol that the proxy server uses to send requests to the back-end server. The attribute value can be `ldap` or `ldaps`.

NOTE: If you specify the protocol as `ldaps`, you must place the certificate file in the `/etc/opt/novell/ldaproxy/conf/ssl/trustedcert` directory.

The `<service>` element can have the following child elements:

- ◆ **<addr-ipv4>**: The IP address of the system on which the back-end server is installed.
- ◆ **<port>**: The port on which the back-end server receives requests.
- ◆ **<addr-dns>**: The domain name of the system where the back-end server is installed.
- ◆ **<tls-opts>**: A configuration option to specify the Transport Layer Security (TLS) parameters when protocol is LDAPS. For more information, see [“Configuring Secured Communication Using TLS Parameters” on page 21](#).

NOTE: If the TLS configuration of LDAP Proxy disables a version of protocol and enables the lower version, Proxy will use the lower version of the protocol to connect to the backend server. If the backend server is not configured with this lower version of the protocol, TLS Handshake fails. For example, if your backend server does not allow `SSLv3` and LDAP Proxy is configured with `+SSLv3+TLSv1.1`, then Proxy will use `SSLv3` to connect to the backend server and connection will fail.

- ◆ **<connection-pool>**: The number of LDAP connections that are cached and maintained by the proxy server so that the connections are reused when the proxy server receives future request.

The `<connection-pool>` element can have the following child elements:

- ◆ **<start-pool-size>**: Specifies the number of LDAP connections that are cached and maintained by the proxy server. The value must always be less than the `max-connections` attribute value. For instance, in [Example 1](#), the `max-connections` value is 5000, whereas the `connection-pool` value specified is 256.
- ◆ **<bind-dn>**: If anonymous bind is disabled on a particular server, then to nullify the connection identity you must specify the User Distinguished Name (user DN). To nullify a connection with a particular `bind dn`, specify the required DN.

Use the `passwdstore` utility to set the user DN password.

```
passwdstore [-a username] [-w password]
```

Replace `username` with the user DN for authentication and `password` is the user DN password for authentication.

IMPORTANT: Ensure that you specify the correct password, because if the authentication fails, the user account might be locked.

For example:

```
passwdstore -a admin -w pass
```

NOTE: It is not recommended to use `admin` DN to nullify a connection. Ideally, it should be a DN with the least privileges.

- ◆ **<health-check>**: Performs periodic health checks to determine the response time of the back-end server. This is an optional element.
If you specify this parameter, the proxy periodically sends an LDAP Bind request to the back-end server and calculates the response time of the request.
To specify the response time of the back-end server, you must use the following attribute:
 - ◆ **max-response-time-ms**: The maximum time (in milliseconds) within which a back-end server must respond when it receives an LDAP Bind request. If it does not respond within the specified time, the back-end server is identified as a slow server. In [Example 1](#), the attribute value is 5000. This indicates that the `Backend1` server must respond to any request within 5000 milliseconds.
 - ◆ **<req-ldap-bind>**: The DN with which the Bind request must be performed to detect a server that is slow to respond.

Configuring Back-End Server on a Secure Port

To configure a back-end server on a secure port, complete the following steps:

- 1 Obtain the root CA certificate in the pem format from the LDAP server for configuring the back-end directory (eDirectory, Active Directory and OpenLDAP).
 - ◆ eDirectory: See [Understanding the Certificate Server](#) in the *eDirectory Administration Guide*.
 - ◆ Active Directory: Refer to the Active Directory documentation.
 - ◆ OpenLDAP: Refer to the OpenLDAP documentation.
 - ◆ 3rd party CA: Refer to the third-party documentation.
- 2 Copy the root CA certificate to the `/etc/opt/novell/ldaproxy/conf/ssl/trustedcert/` directory.

- 3 Replace the `x.x.x.x` in the `<addr-ipv4>` field with your LDAP server IP address and provide the LDAP server secure port in the `nlpconf.xml` file.

Ensure that you define the protocol as `ldaps` under the `<service protocol>` tag as shown below:

```
<list-backend-server health-check-interval-secs="60">
  <backend-server id-backend-server="Backend1">
    <service protocol="ldaps">
      <addr-ipv4>x.x.x.x</addr-ipv4>
      <port>636</port>
    </service>
  </backend-server>
  <backend-server id-backend-server="Backend2">
    <service protocol="ldaps">
      <addr-ipv4>x.x.x.x</addr-ipv4>
      <port>636</port>
    </service>
  </backend-server>
</list-backend-server>
```

- 4 Save the `nlpconf.xml` file.

NOTE: LDAP Proxy reads the `nlpconf.xml` file during startup. For the configuration changes to take effect, restart LDAP Proxy. For more information, see [“Restarting LDAP Proxy” on page 99](#).

Configuring Back-End Server on a Non-Secure Port

To configure the back-end server on a non-secure port, complete the following steps:

- 1 Open the `nlpconf.xml` file from the `/etc/opt/novell/ldapproxy/conf` directory.
- 2 Under the `<backend-server>` configuration, provide the IP address and the non-secure port of your LDAP server in the `<addr-ipv4>` and `<port>` entries.

Ensure that you define the protocol as `ldap` under the `<service protocol>` tag as shown below:

```
<list-backend-server health-check-interval-secs="60">
  <backend-server id-backend-server="Backend1">
    <service protocol="ldap">
      <addr-ipv4>x.x.x.x</addr-ipv4>
      <port>389</port>
    </service>
  </backend-server>
  <backend-server id-backend-server="Backend2">
    <service protocol="ldap">
      <addr-ipv4>x.x.x.x</addr-ipv4>
      <port>389</port>
    </service>
  </backend-server>
</list-backend-server>
```

- 3 Save the `nlpconf.xml` file.

NOTE: LDAP Proxy reads the `nlpconf.xml` file during startup. For the configuration changes to take effect, restart LDAP Proxy. For more information, see [“Restarting LDAP Proxy” on page 99](#).

Configuring Additional Back-End Servers

Manually Configuring Additional Back-End Servers

To add a back-end server in the `nlpconf.xml` file, perform the following steps:

- 1 Open the `nlpconf.xml` file from the `/etc/opt/novell/ldaproxy/conf` directory in any XML editor.
- 2 To add a back-end server to the existing configuration, create an instance of the following section within the `<list-backend-server>` node:

```
<backend-server id-backend-server="Backend1">  
  <service protocol="ldap">  
    <addr-ipv4>x.x.x.x</addr-ipv4>  
    <port>389</port>  
  </service>  
</backend-server>
```

- 3 Specify the following required information in the newly created instance:
 - ◆ The name to identify the back-end servers you are configuring.
 - ◆ The IP address or the domain name of the system on which the back-end server is installed.
 - ◆ The protocol as either `ldap` or `ldaps`.
 - ◆ The port number on which the back-end server receives requests.
- 4 (Optional) Define the following optional parameters to enhance the performance of the back-end server:
 - ◆ The maximum time within which a request must receive a response.
 - ◆ The maximum number of connections that are handled by the back-end server.
 - ◆ The capability of the back-end server relative to the other servers. For example, if the capability of a back-end server is 2, it can be loaded two times more than the other servers.
 - ◆ The number of the connection pool to be created.

For information about the elements/attributes that are used to define these parameters, refer to [“Configuration Parameters For Back-End Server” on page 32](#).

- 5 (Optional) Specify the time interval for performing a health check on all the listed back-end servers. This parameter is defined at the `<list-backend-server>` level, as shown in the sample configuration.
- 6 To add more back-end servers, repeat [Step 2](#) to [Step 4](#).
- 7 Save the `nlpconf.xml` file.

Configuring Additional Back-End Servers Using NLPManager

You can configure additional back-end servers in the proxy configuration, by using the **Backend Servers** tab.

- 1 Run the `./NLPManager` command to start NLPManager.
- 2 To open the `nlpconf.xml` file, in the **Provisioning** menu, click **Open Configuration**:
The Open dialog box appears.
- 3 Browse to the `nlpconf.xml` file in the `/etc/opt/novell/ldaproxy/conf` directory and click **Open**. The `conf` directory is available on the machine where you installed LDAP Proxy.

If NLPManager and LDAP Proxy are installed on different machines, you must either save the `nlpconf.xml` file on the machine where you installed NLPManager or map a network drive to the machine where you installed proxy.

The proxy configuration is displayed in the Project Explorer pane.

- 4 Click the **Backend Servers** option.

The **Backend Servers** tab is displayed.

- 5 To add a back-end server, click the  icon.

The Add New Backend Server window is displayed.

- 6 Specify a name to identify the back-end server you are configuring and click **OK**. The name must be a unique alphanumeric value.

The back-end server configuration fields are displayed in the Editor pane.

- 7 Specify a time interval for performing a health check on all the listed back-end servers:

- 7a Click the **Health Check Interval Configuration** drop-down list.

- 7b Specify a time interval to perform health checks to detect slow or unavailable back-end servers. By default, the interval is set to 60 seconds.

- 8 Specify the following configuration details:

- ◆ **Address Type:** The address type of the interface through which the directory servers receive the requests from LDAP Proxy.

To provide the IP address of the LDAP directory server, select **IPv4** or **IPv6**.

To provide the domain name of the LDAP directory server, select **DNS**.

- ◆ **Address:** The value of the IP address or domain name, depending on the address type you specified.

- ◆ **Protocol:** Specify either **ldap** or **ldaps**.

If you specify the protocol as **ldaps**, You must place the certificate file in the `conf/ssl/trustedcert` directory.

- ◆ **Port:** The port number of the directory server.

- 9 (Optional) To enhance the performance of the back-end server, configure the following optional fields:

- ◆ **Maximum Connections:** The maximum number of connections that can be handled by the back-end server.

- ◆ **Capability:** The capability of the back-end server relative to the other servers. For example, if the capability of a back-end server is 2, it can be loaded two times more than the other servers.

- ◆ **Connection Pool:** Specify if a connection pool must be created. Then specify the pool size value in the **Start Pool Size** field.

- ◆ **Use Anonymous Login:** Specify if anonymous login is required to create a connection pool. If Anonymous Bind is disabled on a particular server, you must specify the User Distinguished Name (user DN) in the associated **Bind DN** field to nullify the connection identity.

- ◆ **Bind DN:** Use to nullify a connection identity.

- ◆ **Start Pool Size:** The number of connections to be created so that these connections can be reused for incoming requests. The value must always be less than the maximum connections value you specify.

- ◆ **Health Check:** Whether a health check must be performed to detect a slow server. If you select this field, you must specify the **Bind DN** and **Maximum Response Time**.

- ♦ **Bind DN:** The User DN on which the health check must be performed.
 - ♦ **Maximum Response Time:** The maximum time within which a request must receive a response.
- 10 To add more back-end servers, repeat [Step 5](#) to [Step 8](#).
 - 11 To delete a back-end server, select the server from the list and click the  icon.
 - 12 Click **Provisioning** > **Save** to save the changes.

Examples

- ♦ [“Example 1” on page 37](#)
- ♦ [“Example 2” on page 38](#)

Example 1

```
<list-backend-server health-check-interval-secs="7200">
  <backend-server id-backend-server="Backend1" capability="1" max-
connections="5000">
    <service protocol="ldap">
      <addr-ipv4>192.168.1.4</addr-ipv4>
      <port>389</port>
    </service>
    <connection-pool>
      <start-pool-size>256</start-pool-size>
    </connection-pool>
    <health-check max-response-time-ms="5000">
      <req-ldap-bind>
        <bind-dn>cn=dummy,o=my_company<bind_dn>
      </req-ldap-bind>
    </health-check>
  </backend-server>
</list-backend-server>
```

In Example 1, the back-end server is identified as `Backend1`. It is defined to use the LDAP protocol for communication with the back-end server. The interface is defined by IP address `192.168.1.4` and port `389`. This example also specifies to perform a health check every `7200` seconds, the capability as `1`, and max-connections to be allowed as `5000`. The connection-pool size is `256`. It also defines a bind request to detect a slow server. The max-response time specified is `5000` milliseconds and the User DN is `cn=dummy,o=novell`.

```
<list-backend-server health-check-interval-secs="7200">
  <backend-server id-backend-server="Backend1" capability="1" max
connections="5000">
    <service protocol="ldap">
      <addr-ipv6>[2015::37]</addr-ipv6>
      <port>389</port>
    </service>
    <connection-pool>
      <start-pool-size>256</start-pool-size>
    </connection-pool>
    <health-check max-response-time-ms="5000">
      <req-ldap-bind>
        <bind-dn>cn=user1,o=company1<bind_dn>
      </req-ldap-bind>
    </health-check>
  </backend-server>
```

```

    <backend-server id-backend-server="Backend2" capability="5" max
connections="7000">>
      <service protocol="ldap">
        <addr-ipv4>192.168.1.1</addr-ipv4>
        <port>1389</port>
      </service>
      <connection-pool>
        <start-pool-size>256</start-pool-size>
      </connection-pool>
      <health-check max-response-time-ms="5000">
        <req-ldap-bind>
          <bind-dn>cn=user2,o=company2</bind-dn>
        </req-ldap-bind>
      </health-check>
    </backend-server>
  </list-backend-server>

```

In Example 1, the back-end servers are identified as `Backend1` and `Backend1`. They are defined to use the LDAP protocol for communication with the back-end server. The interface is defined by IPv6 and IPv4 addresses respectively on ports 389 and 1389. This example also specifies to perform a health check every 7200 seconds. It also specifies the capability as 1 and 5, max-connections to be allowed as 5000 and the connection-pool size is 256. It also defines a bind request to detect a slow server. The max-response time specified is 5000 milliseconds and the User DNs are `cn=user1` and `o=company1`, and `cn=user2` and `o=company2`.

Example 2

```

<list-backend-server>
  <backend-server id-backend-server="Backend1" max-connections="3000">
    <service protocol="ldaps">
      <addr-ipv4>192.168.1.1</addr-ipv4>
      <port>636</port>
    </service>
  </backend-server>
  <backend-server id-backend-server="Backend2">
    <service protocol="ldap">
      <addr-ipv4>192.168.1.3</addr-ipv4>
      <port>3389</port>
    </service>
  </backend-server>
</list-backend-server>

```

In Example 2, two back-end servers are defined as `Backend1` and `Backend2`. `Backend1` is defined to use the `ldaps` protocol and the interface is defined by IP address `192.168.1.1` and port `636`. It also specifies the max-connections to be allowed as 3000. `Backend2` is defined to use the `ldap` protocol, and the IP address `192.168.1.3`, and the port `3389`.

Configuring Server Groups

Server group consists of a number of back-end servers that are configured for LDAP Proxy. Configuring servers into server groups enables the proxy to balance the load between the servers (load balancing) and route requests from a failed server to an active server.

Configuring servers into server groups enables the proxy to balance the load between the servers (load balancing) and route requests around a failed server to an active server (failover).

LDAP Proxy supports both connection-based and dynamic load balancing. When a new connection request is received, the load balancer determines the destination back-end server by calculating the load on each back-end server within a group and identifying the least loaded server and routes the new connection to it. All subsequent requests received for that connection are routed to the same back-end server until the connection is terminated.

In a connection-based load balancing, the load is calculated based on following two factors:

- ◆ The number of active connections
- ◆ The relative capability weight of each back-end server

When all the servers are of equal capability, the connections are routed in a round-robin fashion.

During proxy configuration, you must specify the relative capability weight of each back-end server in the group. Relative capability weight can be determined based on the hardware configuration of the server.

In dynamic load balancing, the load is calculated based on the following two factors:

- ◆ The total number of outstanding and pending requests on each back-end server
- ◆ The current average response time of each back-end server, which is calculated periodically by performing health checks

The factors used for dynamic load balancing provide a more accurate indication of the performance of the back-end servers within a group. Therefore, dynamic load balancing is preferred to connection-based load balancing. For information about how to configure back-end server groups, refer to [“Configuring Server Groups” on page 38](#).

The `<list-load-balancer>` node in the configuration file lists all the back-end server groups configured for the proxy. Additional back-end server groups must be defined in this node.

For instance, if you want to define a back-end server group, `connld`, to be configured with back-end servers `Backend1` and `Backend3` as a part of this connection-based server group. You can define the configuration, as follows:

```
<list-load-balancer>
  <lb-conn-based id-load-balancer="connld">
    <ref-backend-server>Backend1</ref-backend-server>
    <ref-backend-server>Backend3</ref-backend-server>
  </lb-conn-based>
</list-load-balancer>
```

- ◆ [“Configuration Parameters for Server Groups” on page 39](#)
- ◆ [“Configuring Additional Server Groups” on page 40](#)
- ◆ [“Example.” on page 42](#)

Configuration Parameters for Server Groups

The following elements and parameters are used to configure back-end server groups:

- ◆ **<lb-conn-based>**: The configured element is a connection-based load balancer.
- ◆ **<lb-dynamic-load-based>**: The configured element is a dynamic load balancer.

Both the `<lb-conn-based>` and `<lb-dynamic-load-based>` elements must have the following attribute:

- ♦ **id-load-balancer:** The identity of the load balancer (back-end server group). This is a mandatory attribute and its value is used to refer to the load balancer. It must be a unique alphanumeric value. In the sample configuration, the back-end server is identified as `connld`.

Both the `<lb-conn-based>` and `<lb-dynamic-load-based>` elements must have the following child element:

- ♦ **<ref-backend-server>:** The back-end server to be grouped in the defined back-end server group. The term `ref` in this element indicates that this element is actually a pointer to a back-end server. For instance, the sample configuration indicates that the specified connection-based server group is made up of the `Backend1` and `Backend3` back-end servers.
- ♦ **<lb-priority-based>:** The configured element is a priority-based load balancer.

You can configure a priority-based load balancer with a set of servers given in a specific order and have the load balancer to always route to the high priority servers when they are up, as shown in the following example:

```
<lb-priority-based id-load-balancer="backend-grp1">
  <ref-backend-server>Backend2</ref-backend-server>
  <ref-backend-server>Backend1</ref-backend-server>
</lb-priority-based>
<lb-priority-based id-load-balancer="backend-grp2">
  <ref-backend-server>Backend1</ref-backend-server>
</lb-priority-based>
```

In the preceding example, the load balancer `backend-grp1` with server `Backend2` is the highest priority server and `Backend1` has the second highest priority. If you choose `backend-grp1`, it will always route to server `Backend1`. The load balancer `backend-grp2` will always route to `Backend1` and if `Backend1` is not available, the operation fails.

Configuring Additional Server Groups

Manually Configuring Additional Server Groups

To add a back-end server group manually, perform the following steps:

- 1 Open the `nlpconf.xml` file from the `/etc/opt/novell/ldaproxy/conf` directory in any XML editor.
- 2 To add a back-end server group to the existing configuration, create an instance of the following section within the `<list-load-balancer>` node:

```
<list-load-balancer>
  <lb-conn-based id-load-balancer="connld">
    <ref-backend-server>Backend1</ref-backend-server>
    <ref-backend-server>Backend3</ref-backend-server>
  </lb-conn-based>
</list-load-balancer>
```

- 3 In the newly created instance, provide a name for the new back-end server group.
- 4 Provide the name of the back-end servers that you want to configure in this group. If you have more than two servers, add additional `<ref-backend-server>` elements to define each back-end server.

For information about the elements/attributes that are used to define these parameters, refer to [“Configuration Parameters for Server Groups” on page 39](#).

- 5 To add more back-end server groups, repeat [Step 2](#) to [Step 4](#).
- 6 Save the `nlpconf.xml` file.

Configuring Additional Server Groups Using NLPManager

You can configure additional back-end server groups to the proxy configuration by using the **Backend Server Group** tab.

- 1 Run the `./NLPManager` command to start NLPManager.
- 2 To open the `nlpconf.xml` file, do one of the following:
 - ♦ Click the  icon.
 - ♦ In the **Provisioning** menu, click **Open Configuration**.

The Open dialog box appears.

- 3 Browse to the `nlpconf.xml` file in the `/etc/opt/novell/ldaproxy/conf` directory and click **Open**.

The `conf` directory is available on the machine where you installed LDAP Proxy.

If NLPManager and LDAP Proxy are installed on different machines, you must either save the `nlpconf.xml` on the machine where you have installed NLPManager or map a network drive to the machine where you installed the proxy.

The proxy configuration is displayed in the Project Explorer pane.

- 4 Click the **Backend Server Groups** option in the **Project Explorer** pane.

The **Backend Server Group** tab is displayed. By default, one server group is defined, and both of the default back-end servers are defined in this group. You can add more groups and include other servers in each group.

- 5 To add a server group, click the  icon.

The Add New Server Group window is displayed.

- 6 Specify a name to identify the back-end server group you are configuring and click **OK**. The name must be a unique alphanumeric value.

The server group configuration is displayed.

- 7 Specify the following:

- ♦ **Load Balancing:** Whether the type of load balancing is **Connection Based**, **Dynamic**, or **Priority Based**.
- ♦ **Selected Servers:** The back-end servers to be defined in the server group. You can use the arrow buttons to sort servers between the **Selected Servers** and **Available Servers** lists.

- 8 To add more server groups, follow [Step 5](#) to [Step 7](#).

- 9 To delete a server group, select the server group and click the  icon.

- 10 Click **Provisioning** > **Save** to save the changes.

To configure message policies, refer to [“Configuring Policies” on page 42](#).

Example:

```
<list-load-balancer>
  <lb-conn-based id-load-balancer="connld">
    <ref-backend-server>Backend1</ref-backend-server>
    <ref-backend-server>Backend3</ref-backend-server>
  </lb-conn-based>
  <lb-dynamic-load-based id-load-balancer="dyld">
    <ref-backend-server>Backend2</ref-backend-server>
    <ref-backend-server>Backend4</ref-backend-server>
  </lb-dynamic-load-based>
</list-load-balancer>
```

In this example, two back-end server groups are defined as `connld` and `dyld`. `connld` indicates that a connection-based load balancing is performed between `Backend1` and `Backend3`. `dyld` indicates that dynamic load balancing is performed between `Backend2` and `Backend4`.

Configuring Policies

LDAP Proxy uses policies to analyze and act on the incoming requests and outgoing responses. Every request or response is sequentially passed to and processed by the policies defined.

The `<list-policy>` node of the XML configuration file lists all the policies configured for the LDAP Proxy. All the additional policies must be defined in this node.

Every policy defined while configuring the proxy has a rule associated with it. The rule is made up of the following elements:

- ◆ A condition or a group of conditions.
- ◆ An action that must be performed on the incoming request and outgoing responses if the condition evaluates to true.
- ◆ A default action that must be performed if the condition evaluates to false.

NOTE: You must not configure a policy with more than one action at a time. For example, if you configure a Search Request policy with the `do-modify-search` and `do-restrict-view` actions and then restart the LDAP Proxy, the first action is overwritten by the second action.

The following table provides high-level information about the policies which can be configured along with various components of the Proxy server.

Policy Name	Listener	Policy Engine	Back-end Groups
Connection Route Policy	✓		✓
Search Request Policy		✓	
Map Schema Policy		✓	
Replace String Policy		✓	
Client Network Policy	✓		

Policy Name	Listener	Policy Engine	Back-end Groups
Operation Restriction Policy		✓	

Adding a New Policy

To add a policy, follow these steps:

- 1 Open the `nlpconf.xml` file from the `/etc/opt/novell/ldapproxy/conf` directory in any XML editor.
- 2 To add a policy, create an instance within the `<list-policy>` node. Use the sample configuration as a pattern. You must define the policy as the first policy in the node.
- 3 Define a name to identify the policy, a set of conditions, an action, and a default action for the policy.
- 4 To add more policies, repeat Step 2 to Step 3.
- 5 Save the `nlpconf.xml` file.

Generic Configuration Parameters of Policy

Before adding a new policy, you must understand the various configuration parameters that are used to configure policies:

id-policy: The identity of the policy. The value of this attribute is used to refer to the policy that is being configured. It must be a unique alphanumeric value. This is a mandatory attribute.

Policies configuration parameters can have the following child elements:

- ♦ **<description>**: An explanation about the policy. This is an optional element used for reference purposes.
- ♦ **<rule>**: The rule associated with the policy that is configured. Every policy has a rule.

This element can have the following child elements:

- ♦ **<conditions>**: The condition to be evaluated. Each policy can have multiple conditions defined. You can define same or different conditions multiple times based on your requirement. If a policy has multiple conditions, each condition should be combined by the logical operators such as `<and>`, `<or>`, `<not>` in the xml file.

This element can have the several child elements. For more information, see [“Understanding the Condition Elements for Policies” on page 45](#).

- ♦ **<actions>**: The action to be performed if the condition evaluates to true.
- ♦ **<actions-default>**: The default action to be performed if the condition evaluates to false.

NOTE: For Connection Route Policy, only `<do-deny>` and `<do-nothing>` action elements are supported under the `<actions-default>` condition elements.

Supported Condition and Action Elements for Policies

The below table provides the matrix of supported condition and action elements for each policy.

Policy Name	Supported Condition Elements	Supported Action Elements
Connection Route Policy	<ul style="list-style-type: none"> ◆ <if-network-addr> ◆ <if-ip-addr> ◆ <if-port> ◆ <if-bind-dn> ◆ <if-bind-dn-container> ◆ <if-srch-scope> ◆ <if-srch-base> ◆ <if-srch-filter> ◆ <if-srch-selection-attr> ◆ <if-regex-match> <p>NOTE: While using the <if-srch-*> conditions in the Connection Route Policy, you must always use the <if-message-type> condition. For more information, see “Configuring a Connection Route Policy for Routing a Request Based on Search Request” on page 67.</p>	<ul style="list-style-type: none"> ◆ <do-use-route> ◆ <do-deny>
Search Request Policy	<ul style="list-style-type: none"> ◆ <if-srch-scope> ◆ <if-srch-base> ◆ <if-srch-filter> ◆ <if-srch-selection-attr> ◆ <if-regex-match> 	<ul style="list-style-type: none"> ◆ <do-allow> ◆ <do-deny> ◆ <do-modify-search> ◆ <do-restrict-view> ◆ <do-send-monitor-response>
Map Schema Policy	<ul style="list-style-type: none"> ◆ <if-network-addr> ◆ <if-ip-addr> ◆ <if-port> ◆ <if-bind-dn> ◆ <if-bind-dn-container> ◆ <if-message-type> 	<ul style="list-style-type: none"> ◆ <do-map-schema> ◆ <do-nothing>
Replace String Policy	<ul style="list-style-type: none"> ◆ <if-network-addr> ◆ <if-ip-addr> ◆ <if-port> ◆ <if-message-type> 	<ul style="list-style-type: none"> ◆ <do-replace-string> ◆ <do-nothing>
Client Network Policy	<ul style="list-style-type: none"> ◆ <if-network-addr> ◆ <if-ip-addr> ◆ <if-port> 	<ul style="list-style-type: none"> ◆ <do-allow> ◆ <do-deny>
Operation Restriction Policy	<ul style="list-style-type: none"> ◆ <if-message-type> 	<ul style="list-style-type: none"> ◆ <do-allow> ◆ <do-deny>

Understanding the Condition Elements for Policies

Each policy can have one or more conditions defined. The following condition elements are used by various policies:

<if-ip-addr>

This condition is used to match an individual IP address. You must define the attribute `op` specifying the matching operation. The allowed values for `op` are `equal` or `not-equal`. This condition supports both `ipv4` and `ipv6` addresses. Provide `ipv6` address within square brackets.

For instance, you can define the `<if-ip-addr>` condition as follows:

```
<if-ip-addr op="equal">151.155.123.12</if-ip-addr> or  
<if-ip-addr op="equal">[2015::34]</if-ip-addr>
```

<if-port>

This condition is used to match an individual TCP Port number in the range of 1-65536. You must define the attribute `op` specifying the matching operation. The allowed values for `op` are `equal`, `not-equal`, `less-or-equal`, or `greater-or-equal`.

For instance, you can define the `<if-port>` condition as follows:

```
<if-port op="equal">389</if-port>
```

<if-network-addr>

This condition is used to match an IP subnet. You must define the attribute `op` specifying the matching operation. The allowed values for `op` are `equal` or `not-equal`. This condition supports both `IPv4` and `IPv6` subnets. Provide `IPv6` address within square brackets. This element can have the following child elements:

- ♦ **<network-addr>**: This is a mandatory element which is used to specify the subnet address.
You must define one of the following two elements:
- ♦ **<subnet-mask>**: Provide the subnet mask as a value to this element.
- ♦ **<subnet-bits>**: Provide the subnet mask as a value to this element. This element must have a valid subnet bits value.

NOTE: If you are using a `IPv6` network address, use only the `<subnet-bits>` element.

For instance, you can define the `<if-network-addr>` condition as follows:

- ♦ **IPv4 subnets:**

```
<if-network-addr>  
  <network-addr>164.99.178.0</network-addr>  
  <subnet-bits>24</subnet-bits>  
</if-network-addr>
```

- ♦ **IPv6 subnets:**

```
<if-network-addr>  
  <network-addr>[2015::f4]</network-addr>  
  <subnet-bits>64</subnet-bits>  
</if-network-addr>
```

<if-bind-dn-container>

This condition is used to match the LDAP bind DN container in an incoming bind request. You must define the attribute `op` specifying the matching operation. The allowed values for `op` are `equal`, `not-equal`.

You can also define value for the optional attribute `match`. The allowed values are `case-ignore` and `case-exact`. If the attribute `match` is not defined, the default value will be `case-ignore`.

For instance, you can define the `<if-bind-dn-container>` condition as follows:

```
<if-bind-dn-container op="equal" match="case-ignore">ou=provo,o=novell</if-bind-dn-container>
```

<if-bind-dn>

This condition is used to match the LDAP bind DN in an incoming bind request. You must define the attribute `op` specifying the matching operation. The allowed values for `op` are `equal`, `not-equal`.

You can also define value for the optional attribute `match`. The allowed values are `case-ignore` and `case-exact`. If the attribute `match` is not defined, the default value will be `case-ignore`.

For instance, you can define the `<if-bind-dn>` condition as follows:

```
<if-bind-dn op="equal" match="case-ignore">cn=admin,ou=provo,o=novell</if-bind-dn>
```

<if-regex-match>

This condition is used to evaluate search filters in a search request using a regular expression. This new condition allows you to use POSIX Extended Regular Expressions to evaluate the search filters. The allowed value for `field` is `srch-filter`.

For instance, you can define the `<if-regex-match>` condition as follows:

```
<if-regex-match field="srch-filter"> [POSIX Extended Regular Expression] </if-regex-match>
```

The below example allows `mail=` searches only when the domain is `.com`. For other domains, the request is blocked.

```
<if-regex-match field="srch-filter">mail=\w*@\w*\com</if-regex-match>
```

<if-srch-selection-attr>

This condition is used to match the LDAP search selection attribute in an incoming search request. You must define the attribute `op` specifying the matching operation. The allowed values for `op` are `equal`, `not-equal`.

You can also define value for the optional attribute `match`. The allowed values are `case-ignore` and `case-exact`. If the attribute `match` is not defined, the default value will be `case-ignore`.

For instance, you can define the `<if-search-selection-attr>` condition as follows:

```
<if-srch-selection-attr op="equal" match="case-ignore">acl</if-srch-selection-attr>
```

<if-srch-base>

This condition is used to match the LDAP search base in an incoming search request. You must define the attribute `op` specifying the matching operation. The allowed values for `op` are `equal`, `not-equal`.

You can also define value for the optional attribute `match`. The allowed values are `case-ignre` and `case-exact`. If the attribute `match` is not defined, the default value will be `case-ignore`.

For instance, you can define the `<if-search-base>` condition as follows:

```
<if-srch-base op="equal" match="case-ignore">o=novell</if-srch-base>
```

<if-srch-filter>

This condition is used to match the LDAP search filter in an incoming search request. You must define the attribute `op` specifying the matching operation. The allowed values for `op` are `equal`, `not-equal`.

You must also define the `filter-type` attribute to match on the standard LDAP search filter types. Correspondingly the allowed values for this attribute are `substrings`, `greater-or-equal`, `less-or-equal`, `present`, `approx-match` or `equality-match`.

This condition has an optional child element:

- ◆ **<filter-attribute>**: This element is used to match on the attributes present in the LDAP search filter. You can also define value for the optional attribute `match`. The allowed values are `case-ignre` and `case-exact`. If the attribute `match` is not defined, the default value will be `case-ignore`.

If `filter-attribute` element is not specified, the condition will apply on all LDAP search requests irrespective of the attributes.

For instance, you can define the `<if-srch-filter>` condition as follows:

```
<if-srch-filter filter-type="present" op="equal">  
  <filter-attribute match="case-ignore">cn</filter-attribute>  
</if-srch-filter>
```

<if-srch-scope>

This condition is used to match the LDAP search scope in an incoming search request. You must define the attribute `op` specifying the matching operation. The allowed values for `op` are `equal`, `not-equal`.

The allowed values for this condition are `base-object`, `one-level`, and `sub-tree`, corresponding to the standard LDAP search scopes.

For instance, you can define the `<if-srch-scope>` condition as follows:

```
<if-srch-scope op="equal">sub-tree</if-srch-scope>
```

<if-message-type>

This condition is used to match the LDAP message type in an incoming request. You must define the attribute `op` specifying the matching operation. The allowed values for `op` are `equal`, `not-equal`. The allowed values for this condition are mentioned below corresponding to the standard LDAP message types:

- ◆ `ldap-bind-request`

- ◆ ldap-bind-response
- ◆ ldap-search-request
- ◆ ldap-search-result-entry-response
- ◆ ldap-search-result-reference-response
- ◆ ldap-search-result-done-response
- ◆ ldap-modify-request
- ◆ ldap-modify-response
- ◆ ldap-add-request
- ◆ ldap-add-response
- ◆ ldap-delete-request
- ◆ ldap-delete-response
- ◆ ldap-moddn-request
- ◆ ldap-moddn-response
- ◆ ldap-compare-request
- ◆ ldap-compare-response
- ◆ ldap-extended-request
- ◆ ldap-extended-response
- ◆ ldap-abandon-request
- ◆ ldap-intermediate-response
- ◆ ldap-unbind-request

For instance, you can define the `<if-message-type>` condition as follows:

```
<if-message-type op="equal">ldap-add-request</if-message-type>
```

Understanding the Action Elements for Policies

Each policy must have one action and one default action defined. When the condition evaluates to true, the specified action will be performed. When the condition evaluates to false, the default action will be performed. The following actions are allowed to be defined as Actions and Default Actions:

<do-allow>

This action element is used to allow the connection flow through to the next policy in the configuration without modifying the LDAP request. This element doesn't have any child elements.

<do-nothing>

This action element serves the same functionality as `<do-allow>`. In some policies, `<do-nothing>` element is allowed. For more information, see [“Supported Condition and Action Elements for Policies” on page 43](#).

<do-deny>

This action element is used to stop the connection flow to the next policy. Client will see an LDAP error message saying `Insufficient access rights` when this action element is executed.

<do-map-schema>

This action element is used to map the schema based on attributes. This element has a child element called `attributes`. The child element `attributes` can have multiple child elements called `map-attribute`. Each `map-attribute` defines a schema mapping on one attribute.

For instance, you can define the `<do-map-schema>` element as follows:

```
<do-map-schema>
  <attributes>
    <map-attribute name="cn">FirstName</map-attribute>
    <map-attribute name="sn">LastName</map-attribute>
  </attributes>
</do-map-schema>
```

In the above example, the schema mapping is defined as follows:

- ◆ The `cn` attribute maps to the `FirstName` attribute.
- ◆ The `sn` attribute maps to the `LastName` attribute.

<do-modify-search>

This element is used to modify an LDAP search request. This element allows the following child elements:

- ◆ **<description>**: An explanation about the modify search request.
- ◆ **<base>**: This element is used to specify modifications that are done on the search base. You must define the attribute `op` specifying the modify operation. The allowed values for `op` are `replace` and `append`.
- ◆ **<scope>**: This element is used to specify the scope of the search request. The three types of search scope are `base-object`, `one-level`, and `sub-tree`.
- ◆ **<time-limit>**: This element is used to specify the time limit of the search request.
- ◆ **<size-limit>**: This element is used to specify the size limit of the search request.
- ◆ **<derefalias-reset>**: This element is used to reset the alias dereferencing value set by the client. Corresponding to the standard LDAP `derefalias` options- `never`, `always`, `search`, `find`, the allowed values for this element are 0, 1, 2, 3 respectively.
- ◆ **<selection-attributes>**: This element is used to specify modifications to the LDAP search selection attribute list. Using this element, you can add, delete or replace attributes in the selection list. Correspondingly the following three child element are allowed in this element:
 - ◆ `<add>`
 - ◆ `<delete>`
 - ◆ `<replace>`

Each of the above elements allows any number of `<value>` child elements to specify the attribute names which are to be added, deleted or replaced. The `<replace>` element has a mandatory `name` attribute to specify the selection attribute to be replaced.

For instance, you can define selection attributes as follows:

```

<do-modify-search>
  <base op="append">ou=sales,o=novell</base>
  <scope>subtree</scope>
  <time-limit>20</time-limit>
  <size-limit>120</size-limit>
  <derefalias-reset>2</derefalias-reset>
  <selection-attributes>
    <add>
      <value>localEntryID</value>
    </add>
    <delete>
      <value>creationTimeStamp</value>
    </delete>
    <replace name="*">
      <value>cn</value>
      <value>sn</value>
    </replace>
  </selection-attributes>
</do-modify-search>

```

<do-restrict-view>

This element is used to modify the LDAP search result in order to restrict the result entries to return specific elements. You can define the following child elements to modify the LDAP search result:

- ♦ **<attributes>**: Shows the attributes of the directory tree.
- ♦ **<objectclasses>**: Shows the objectclass of the directory tree.
- ♦ **<containers>**: Shows the containers of the directory tree.

Each of the above elements allows any number of <value> child elements. You must define the attribute `op` specifying the operation. The allowed values for `op` are `show-only` and `hide`.

For instance, you can define the `<do-restrict-view>` action as follows:

```

<do-restrict-view>
  <attributes op="show-only">
    <value>cn</value>
    <value>objectClass</value>
  </attributes>
  <objectclass op="show-only">
    <value>inetOrgPerson</value>
    <value>groupOfNames</value>
  </objectclass>
  <container op="hide">
    <value>o=security</value>
    <value>o=support</value>
  </container>
</do-restrict-view>

```

<do-send-monitor-response>

This is a special action element which is used to enable the live monitoring feature of LDAP Proxy. This element must be used in a policy connected to a dedicated listener and this listener can be used for live monitoring of the Proxy server. For more information on how to use NLPManager on this listener, see [“Configuring Monitoring Activities” on page 85](#).

<do-use-route>

This action element is used to route the incoming LDAP requests to a load balancer. This element also allows us to connect multiple policies before routing the request to a load balancer. The following child elements are allowed in this policy:

- ♦ **<ref-policy>**: This child element is used to connect other policies using their policy id. All policies except Client Network Policy are allowed. Multiple `<ref-policy>` elements can be used to connect multiple policies. The order of `<ref-policy>` elements will determine the order of the policy execution.

NOTE: All id-policy that you have mentioned in the above element, must be predefined within the `<list-policy>` element.

- ♦ **<ref-load-balancer>**: This element is used to route the incoming request to a back-end server group. The value specified here should be the id of a load balancer element defined earlier within the `<list-load-balancer>`.
- ♦ **<ref-hash-balancers>** : This element is used to configure a hash-based route policy with a group of load balancers. The value of this element must be a common pattern on the load balancer ids. This element has two attributes; `num-buckets` and `hash-route-all`. `num-buckets` takes a numerical value specifying the number of hash-buckets.

`hash-route-all` is a boolean attribute. If this is set to true, all LDAP incoming requests including binds are routed through the hash policy.

NOTE: For a given Connection Route policy, you can have either have the `<ref-load-balancer>` element or the `<ref-hash-balancers>` element and *not* both.

<do-replace-string>

This element is used to match & replace strings on the attribute values in the LDAP request and response. The operation will be performed on the attribute(s) which are specified in the `<attributes>` child element. The `<attributes>` child element can have multiple `<value>` elements specifying each attribute. The LDAP DN attribute is always operated on irrespective of the attributes specified.

The actual match & replace operation is specified using an `<order>` element. The `<order>` element can have multiple `<replace>` elements. Each `<replace>` element has a pair of `<from>` and `<to>` elements. Each `<replace>` element is used to perform a single match replace operation. In LDAP request, when the value specified in `<from>` is matched and replaced with the values specified in `<to>`. In an LDAP response, `<to>` element is matched and replaced with the value in the `<from>` element. Each replace operation specified within the `<order>` element happens on the LDAP request in the same order. In an LDAP response, the operations are performed in the reverse order.

For instance, you can define the `<do-replace-string>` as follows:

```

<do-replace-string>
  <attributes>
    <value>manager</value>
    <value>seeAlso</value>
  </attributes>
  <order>
    <replace>
      <from>o=company</from>
      <to>ou=marketing,o=company</to>
    </replace>
    <replace>
      <from>o=something</from>
      <to>ou=something,o=company</to>
    </replace>
  </order>
</do-replace-string>

```

Use Cases with Examples

You can perform the following tasks using policies for LDAP Proxy:

- ♦ [“Accepting or Denying a Client Connection \(Client Network Policy\)”](#) on page 52
- ♦ [“Restricting LDAP Operations \(Operation Restriction Policy\)”](#) on page 55
- ♦ [“Configuring Access Control Based on Users \(Operation Restriction Policy\)”](#) on page 56
- ♦ [“Mapping the Back-End Server Schema with the Application-Specific Schema \(Map Schema Policy\)”](#) on page 57
- ♦ [“Mapping the Schema Based on the Network and Users \(Map Schema Policy\)”](#) on page 57
- ♦ [“Evaluating Incoming Search Request \(Search Request Policy\)”](#) on page 58
- ♦ [“Setting a Search Base for User Identities \(Search Request Policy\)”](#) on page 62
- ♦ [“Preventing Wild Card Search Filters \(Search Request Policy\)”](#) on page 63
- ♦ [“Routing an Incoming Request to a Back-End Server Group \(Connection Route Policy\)”](#) on page 63
- ♦ [“Additional Configuration Parameters for Connection Route Policy”](#) on page 64
- ♦ [“Replacing a String Sequence in Directory Attribute Values \(Replace String Policy\)”](#) on page 68

Accepting or Denying a Client Connection (Client Network Policy)

The Client Network policy is an optional policy that acts as a directory firewall. This policy helps to accept/reject clients depending on the IP/subnet addresses. Configuring this policy, you can either allow Proxy to establish connection to a client that resides in a particular internal network or reject the requests that are coming from a specific IP address in a particular location subnet.

The Client Network policy must be defined in the `<list-policy>` node of the XML configuration file.

For instance, if you want to define a simple Client Network policy that has one condition and its relevant action and default action. Any incoming client requests from a network with an IP address equal to 192.168.1.0 and with subnet bits equal to 24 must be allowed to establish a connection.

From LDAP Proxy 1.5 onwards, it is possible to specify IPv6 addresses of specific hosts or subnets to which client network restriction policy should apply, as shown in Example 3. This works similar to IPv4 addresses.

You can define the configuration as follows:

```

<list-policy>
<policy-client-network id-policy="client-policy">
  <rule>
    <comment>Allow clients from a particular network</comment>
    <conditions>
      <if-network-addr op="equal">
        <network-addr>192.168.1.0</network-addr>
        <subnet-mask>255.255.255.0</subnet-mask>
      </if-network-addr>
      <if-network-addr op="equal">
        <network-addr>[2021::89]</network-addr>
        <subnet-bits>64</subnet-bits>
      </if-network-addr>
    </conditions>
    <actions>
      <do-allow/>
    </actions>
    <actions-default>
      <do-deny/>
    </actions-default>
  </rule>
</policy-client-network>
</list-policy>

```

Examples

- ◆ [“Example 1” on page 53](#)
- ◆ [“Example 2” on page 54](#)
- ◆ [“Example 3” on page 54](#)

Example 1

```

<list-policy>
<policy-client-network id-policy="client-policy">
  <rule>
    <comment>Allow clients with network IP address as 164.99.148.12 and subnet-bits
as 24 to establish a connection</comment>
    <conditions>
      <or>
        <if-network-addr op="equal">
          <network-addr>192.168.5.0</network-addr>
          <subnet-bits>24</subnet-bits>
        </if-network-addr>
        <and>
          <if-ip-addr op="equal">151.155.123.12</if-ip-addr>
          <if-port op="less-or-equal">1024</if-port>
        </and>
      </or>
    </conditions>
    <actions>
      <do-allow/>
    </actions>
    <actions-default>
      <do-deny/>
    </actions-default>
  </rule>
</policy-client-network>
</list-policy>

```

Example 1 specifies a Client Network policy identified as `client-policy`. Any incoming client request from either a network having an IP address equal to `192.168.1.0` and having subnet-bits as `24` or a client having an IP address `151.155.123.12` and port number less than or equal to `1024` is allowed to establish a connection.

Example 2

```
<list-policy>
  <policy-client-network id-policy="restrict-a-network">
    <rule>
      <conditions>
        <if-network-addr op="equal">
          <network-addr>192.168.5.0</network-addr>
          <subnet-mask>255.255.255.0</subnet-mask>
        </if-network-addr>
      </conditions>
      <actions>
        <do-deny/>
      </actions>
      <actions-default>
        <do-allow/>
      </actions-default>
    </rule>
  </policy-client-network>
</list-policy>
```

Example 2 specifies a Client Network policy identified as `restrict-a-network`. Any incoming client requests from a network having IP address equal to `192.168.0.0` and having subnet-mask as `255.255.254.0` cannot establish a connection.

Example 3

```
<policy-client-network id-policy="allow_my_company_network" disabled="false">
  <rule>
    <conditions>
      <or>
        <if-network-addr op="equal">
          <network-addr>192.168.5.0</network-addr>
          <subnet-mask>255.255.255.0</subnet-mask>
        </if-network-addr>
        <if-network-addr op="equal">
          <network-addr>[2021::89]</network-addr>
          <subnet-bits>64</subnet-bits>
        </if-network-addr>
        <if-network-addr op="equal">
          <network-addr>[2015:c5::ad]</network-addr>
          <subnet-bits>63</subnet-bits>
        </if-network-addr>
        <if-ip-addr op="equal">[2045:ec:54::de]</if-ip-addr>
        <if-ip-addr op="equal">132.0.0.0</if-ip-addr>
      </or>
    </conditions>
    <actions>
      <do-allow/>
    </actions>
    <actions-default>
      <do-deny/>
    </actions-default>
  </rule>
</policy-client-network>
```

In a Example 3, a Client Network policy identified as `allow_my_company_network`. Any incoming client requests from a network having the following IP addresses and subnet-masks can establish a connection:

- ◆ Network address 192.168.5.0 with subnet mask 255.255.255.0
- ◆ Network address [2021::89] with 64 subnet bits
- ◆ Network address [2015:c5::ad] with 63 subnet bits
- ◆ IPv6 address [2045:ec:54::de]
- ◆ IPv4 address 132.0.0.0

Restricting LDAP Operations (Operation Restriction Policy)

The Operation Restriction policy is an optional policy that is used to restrict certain LDAP operations. Using this policy you can restrict few LDAP operations such as Bind, Search, Modify, Add, Delete, Moddn, Compare, and Extended Requests.

The Operation Restriction policy must be defined in the `<list-policy>` node of the XML configuration file.

For instance, if you want to define an Operation Restriction policy, `restrict-operation`, that denies all modify operations on the back-end server and allows only read-only operations and sends them to the back-end group. The configuration can be defined as follows:

```
<list-policy>
  <policy-client-restriction id-policy="restrict operation">
    <rule>
      <conditions>
        <or>
          <if-message-type op="equal" >ldap-bind-request</if-message-type>
          <if-message-type op="equal" >ldap-add-request</if-message-type>
          <if-message-type op="equal" >ldap-modify-request</if-message-type>
          <if-message-type op="equal" >ldap-delete-request</if-message-type>
        </or>
      </conditions>
      <actions>
        <do-deny/>
      </actions>
      <actions-default>
        <do-allow/>
      </actions-default>
    </rule>
  </policy-client-restriction>
</list-policy>
```

Example

```
<list-policy>
  <policy-client-restriction id-policy="restrict operation">
    <rule>
      <conditions>
        <or>
          <if-message-type op="equal" >ldap-bind-request</if-message-type>
          <if-message-type op="equal" >ldap-search-request</if-message-type>
        </or>
      </conditions>
      <actions>
        <do-allow/>
      </actions>
      <actions-default>
        <do-deny/>
      </actions-default>
    </rule>
  </policy-client-restriction>
</list-policy>
```

This example specifies that this policy node is used before the request is sent to the load balancer. The back-end group behaves like a directory used for performing searches only.

Configuring Access Control Based on Users (Operation Restriction Policy)

The `nlpconf_AccessBasedOnUsers.xml` is a sample configuration file in the `/etc/opt/novell/ldaproxy/conf-sample` directory which configures a particular set of servers to behave as read-only for all users except users from specific groups.

```
<policy-operation-restriction id-policy="restrict_operations" disabled="false">
  <rule>
    <conditions>
      <or>
        <if-message-type op="equal">ldap-modify-request</if-message-type>
        <if-message-type op="equal">ldap-add-request</if-message-type>
        <if-message-type op="equal">ldap-delete-request</if-message-type>
        <if-message-type op="equal">ldap-moddn-request</if-message-type>
      </or>
    </conditions>
    <actions>
      <do-deny/>
    </actions>
    <actions-default>
      <do-allow/>
    </actions-default>
  </rule>
</policy-operation-restriction>

<policy-connection-route id-policy="support_users" disabled="false">
  <rule>
    <conditions>
```

```

        <if-bind-dn-container op="equal" match="case-ignore">
            ou=support,o=my_company</if-bind-dn-container>
    </conditions>
    <actions>
        <do-use-route>
            <ref-policy>restrict_operations</ref-policy>
            <ref-load-balancer>backend_grp1</ref-load-balancer>
        </do-use-route>
    </actions>
    <actions-default>
        <do-nothing/>
    </actions-default>
</rule>
</policy-connection-route>

```

Mapping the Back-End Server Schema with the Application-Specific Schema (Map Schema Policy)

The Map Schema policy is an optional policy that is used to map the back-end server schema to the application-specific schema.

The Map Schema policy must be defined in the `<list-policy>` node of the XML configuration file.

For instance, if you want to define a Map Schema policy to map the attribute names of a directory to custom attribute names. You can configure the policy as follows:

```

<policy-map-schema id-policy="schema-map">
    <rule>
        <comment>Maps the attribute-names of the directory to custom attribute-names</comment>
        <actions>
            <do-map-schema>
                <attributes>
                    <map-attribute name="cn" syntax="dn">CommonName</map-attribute>
                    <map-attribute name="c" syntax="dn">country</map-attribute>
                    <map-attribute name="o" syntax="dn">organization</map-attribute>
                </attributes>
            </do-map-schema>
        </actions>
        <actions-default>
            <do-nothing/>
        </actions-default>
    </rule>
</policy-map-schema>

```

Mapping the Schema Based on the Network and Users (Map Schema Policy)

When the identity of an incoming request is established, the Map Schema policy checks the attribute name specified in the client request and matches these names to the relevant attribute names in the back-end server schema. For example, you can use this policy to configure the proxy for following users:

- ◆ For users from `ou=finance`, `ou=hr`, `ou=sales`, `ou=marketing`, map attributes `cn` to `commonName` and `sn` to `surName`

- ♦ For users from ou=eng_dept, map attributes cn to FirstName and sn to LastName
- ♦ For users from ou=support and other users, do not map the schema

For more information about the elements and attributes that are used to define a Map Schema policy, refer to [“Generic Configuration Parameters of Policy” on page 43](#).

Example 1:

```
<policy-map-schema id-policy="fin_mktg_hr_sales_schema" disabled="false">
  <rule>
    <actions>
      <do-map-schema>
        <attributes>
          <map-attribute name="cn">commonName</map-attribute>
          <map-attribute name="sn">surName</map-attribute>
        </attributes>
      </do-map-schema>
    </actions>
    <actions-default>
      <do-nothing/>
    </actions-default>
  </rule>
</policy-map-schema>
```

This example specifies that this policy node is used to map client request attribute name `cn` to `commonName` and `sn` to `surName` for users from `ou=finance`, `ou=hr`, `ou=sales` and `ou=marketing`.

Example 2:

```
<policy-map-schema id-policy="eng_schema" disabled="false">
  <rule>
    <actions>
      <do-map-schema>
        <attributes>
          <map-attribute name="cn">FirstName</map-attribute>
          <map-attribute name="sn">LastName</map-attribute>
        </attributes>
      </do-map-schema>
    </actions>
    <actions-default>
      <do-nothing/>
    </actions-default>
  </rule>
</policy-map-schema>
```

This example specifies that this policy node is used to map client request attribute name `cn` to `FirstName` and `sn` to `LastName` for users from `ou=eng_dept`.

Evaluating Incoming Search Request (Search Request Policy)

The Search Request policy is an optional policy which evaluates an incoming search request, then process it either by modifying or by denying the request.

You must define the Search Request policy in the `<list-policy>` node of the XML configuration file.

For instance, if you want to define a Search Request policy as search-policy, with the search scope as sub-tree and the filter-type as present. You want the match attribute to be defined as case-ignore, which means that the container can be either cn or CN. If any of these conditions are found to be true, then the search request is denied. You can define the configuration as follows:

```
<list-policy>
  <policy-search-request id-policy="search-policy">
    <rule>
      <comment>deny subtree search with cn=* filter or allow attributes requests
except "acl"</comment>
      <conditions>
        <or>
          <if-srch-selection-attr op="equal" match="case-ignore">acl</if-srch-selection-
attr>
          <if-srch-scope op="equal">sub-tree</if-srch-scope>
          <if-srch-filter filter-type="present" op="equal">
            <filter-attribute match="case-ignore">cn</filter-attribute>
          </if-srch-filter>
        </or>
      </conditions>
      <actions>
        <do-deny/>
      </actions>
      <actions-default>
        <do-allow/>
      </actions-default>
    </rule>
  </policy-search-request>
</list-policy>
```

Examples

- ◆ [“Example 1” on page 59](#)
- ◆ [“Example 2” on page 60](#)
- ◆ [“Example 3” on page 60](#)

Example 1

```
<list-policy>
  <policy-search-request id-policy="search-restriction">
    <rule>
      <conditions>
        <or>
          <if-srch-base op="equal" match="case-ignore">ou=dept1,ou=dept4,o=my_company</
if-srch-base>
          <if-srch-base op="equal" match="case-ignore">ou=dept2,o=my_company</if-srch-
base>
        </or>
      </conditions>
      <actions>
        <do-allow/>
      </actions>
      <actions-default>
        <do-deny/>
      </actions-default>
    </rule>
  </policy-search-request>
</list-policy>
```

Example 1 uses the search only if the search base is either `ou=dept1`, `ou=dept4,o=my_company` or `ou=dept2,o=my_company`.

Example 2

```
<list-policy>
  <policy-search-request id-policy="cn-monitor">
    <rule>
      <actions>
        <do-send-monitor-response/>
      </actions>
      <actions-default>
        <do-allow/>
      </actions-default>
    </rule>
  </policy-search-request>
  <policy-connection-route id-policy="monitor-admin">
    <rule>
      <comment>allow all</comment>
      <conditions>
        <if-bind-dn op="equal">cn=admin,o=mycompany</if-bind-dn>
      </conditions>
      <actions>
        <do-use-route>
          <ref-policy>cn-monitor</ref-policy>
          <ref-load-balancer>back-dynld</ref-load-balance>
        </do-use-route>
      </actions>
      <actions-default>
        <do-nothing/>
      </actions-default>
    </rule>
  </policy-connection-route>
</list-policy>
```

Example 2 specifies that a monitor request `<do-send-monitor-response>` is described in a search policy, `cn-monitor`. This search policy is referred in a Connection Route policy, `monitor-admin`.

Example 3

```
<proxy-configuration>
  <proxy-paths>
    <dir-config>/etc/opt/novell/ldaproxy/conf</dir-config>
    <dir-log>/var/opt/novell/ldaproxy/log</dir-log>
  </proxy-paths>
  <list-listener>

  <!-- Listener for LDAP requests .... All the ldap request should go to this
  Listener (IP & Port) -->
    <listener id-listener="listener1">
      <service protocol="ldap">
        <addr-ipv4>192.168.1.2</addr-ipv4>
        <port>389</port>
      </service>
      <ref-policy-connection-route>conn-route-policy</ref-policy-connection-route>
    </listener>

  <!-- Listener for Live Monitor -->
    <listener id-listener="listener2">
      <service protocol="ldap">
```

```

        <addr-ipv4>192.168.1.3</addr-ipv4>
        <port>1389</port>
    </service>
    <ref-policy-connection-route>monitor-policy</ref-policy-connection-route>
</listener>

</list-listener>
<list-backend-server health-check-interval-secs="60">
    <backend-server id-backend-server="Backend1" max-connections="0"
capability="1">
        <service protocol="ldap">
            <addr-ipv4>192.168.0.111</addr-ipv4>
            <port>389</port>
        </service>
        <health-check max-response-time-ms="5000">
            <req-ldap-bind>
                <bind-dn>cn=wyld,ou=users,o=sna</bind-dn>
            </req-ldap-bind>
        </health-check>
    </backend-server>

    <backend-server id-backend-server="Backend2" max-connections="0"
capability="1">
        <service protocol="ldap">
            <addr-ipv4>192.168.0.112</addr-ipv4>
            <port>389</port>
        </service>
        <health-check max-response-time-ms="5000">
            <req-ldap-bind>
                <bind-dn>cn=wyld,ou=users,o=sna</bind-dn>
            </req-ldap-bind>
        </health-check>
    </backend-server>
</list-backend-server>

<list-load-balancer>
    <lb-conn-based id-load-balancer="backend-grp1">
        <ref-backend-server>Backend1</ref-backend-server>
        <ref-backend-server>Backend2</ref-backend-server>
    </lb-conn-based>
</list-load-balancer>
<list-policy>
    <policy-search-request id-policy="monitor_request">
        <rule>
            <action>
                <do-send-monitor-response/>
            </action>
            <action-default>
                <do-allow/>
            </action-default>
        </rule>
    </policy-search-request>

    <policy-connection-route id-policy="conn-route-policy" disabled="false">
        <rule>
            <comment>
                Route all connections to the backend-grp1
            </comment>
            <actions>
                <do-use-route>

```

```

        <ref-load-balancer>backend-grpl</ref-load-balancer>
    </do-use-route>
</actions>
<actions-default>
    <do-nothing/>
</actions-default>
</rule>
</policy-connection-route>

<policy-connection-route id-policy="monitor-policy" disabled="false">
    <rule>
        <comment>
            Policy for Live Monitor
        </comment>
        <actions>
            <do-use-route>
                <ref-policy>monitor_request</ref-policy>
                <ref-load-balancer>backend-grpl</ref-load-balancer>
            </do-use-route>
        </actions>
        <actions-default>
            <do-nothing/>
        </actions-default>
    </rule>
</policy-connection-route>
</list-policy>
</proxy-configuration>

```

Example 3 specifies that a listener listener2 is configured for live monitoring. listener2 includes a Connection Route policy monitor-policy. This policy includes the monitor_request policy for live monitoring, which contains the monitor request policy <do-send-monitor-response>.

Setting a Search Base for User Identities (Search Request Policy)

The nlpconf_SearchBasedOnUserIdentity.xml is a sample configuration file available in the /etc/opt/novell/ldaproxy/conf-sample directory which enables a proxy server to automatically append a search base to default containers when a base is supplied. For more information on configuring a Search Request Policy, refer to ["Evaluating Incoming Search Request \(Search Request Policy\)" on page 58](#).

For instance, if you want to define a Search Restriction Policy to enable Proxy server to replace ou=finance and o=my_company, define your configuration as follows:

```

</policy-search-request>
    <policy-search-request id-policy="finance_search_help" disabled="false">
        <rule>
            <conditions>
                <if-srch-base op="equal" match="case-ignore"/>
            </conditions>
            <actions>
                <do-modify-search>
                    <base op="replace">ou=finance,o=my_company</base>
                </do-modify-search>
            </actions>
            <actions-default>
                <do-allow/>
            </actions-default>
        </rule>
    </policy-search-request>

```

In the above example, the proxy server will automatically replace `ou=finance, o=my_company` as a search base to default containers when a base is supplied.

Preventing Wild Card Search Filters (Search Request Policy)

The `nlpconf_PreventWildSearch.xml` is a sample configuration file in the `/etc/opt/novell/ldaproxy/conf-sample` directory which enables the Proxy server to prevent substring searches reaching to the back-end servers. For more information on configuring a Search Request Policy, refer to [“Evaluating Incoming Search Request \(Search Request Policy\)” on page 58](#).

For instance, if you want to define a Search Restriction Policy to enable LDAP Proxy server to prevent substring searches such as `cn=*` reaching to the back-end servers, define your configuration as follows:

```
<policy-search-request id-policy="search_filter" disabled="false">
  <rule>
    <conditions>
      <if-srch-filter filter-type="substrings" op="equal">
        <filter-attribute match="case-ignore">cn</filter-attribute>
      </if-srch-filter>
    </conditions>
    <actions>
      <do-deny/>
    </actions>
    <actions-default>
      <do-allow/>
    </actions-default>
  </rule>
</policy-search-request>
```

In the above example, the proxy server prevents users from sending the `cn=*` type of LDAP searches.

Routing an Incoming Request to a Back-End Server Group (Connection Route Policy)

The Connection Route policy is a mandatory policy that is used to route an incoming request to the appropriate back-end server group.

By default, one Connection Route policy is defined in the `<list-policy>` node and the defined Connection Route policy is referred in the `<list-listener>` node in the configuration file. You can add more Connection Route policies to this configuration.

For instance, if you want to define a Connection Route policy, `all-clients`, to specify that an incoming request from either a network IP address `192.168.1.1` with 24 subnet bits or a base `ou=dept1, o=my_company` must be routed and analyzed by the search policy defined as `search-policy`. It is then passed on to the back-end server group called `connld`. You can define the configuration, as follows:

```

<list-policy>
  <policy-connection-route id-policy="all-clients" request-route-dit="backend-tree-
name">
    <rule>
      <conditions>
        <or>
          <if-network-addr op="equal">
            <network-addr>192.168.1.1</network-addr>
            <subnet-bits>24</subnet-bits>
          </if-network-addr>
          <if-bind-dn-container op="equal" match="case-ignore">ou=dept1,o=my_company</
if-bind-
dn-container>
          </or>
        </conditions>
        <actions>
          <do-use-route>
            <ref-policy>search-policy</ref-policy>
            <ref-load-balancer>conmlc</ref-load-balance>
          </do-use-route>
        </actions>
        <actions-default>
          <do-nothing/>
        </actions-default>
      </rule>
    </policy-connection-route>
  </list-policy>

```

Additional Configuration Parameters for Connection Route Policy

The following additional elements and attributes are required to configure Connection Route policies:

<policy-connection-route>: Specifies that the element configured is a Connection Route policy. This element must have the following attributes:

- ♦ **id-policy**: The identity of the policy. The value of this attribute is used to refer to the policy that is being configured. This is a mandatory attribute. You must specify this attribute to enable the modify entry cache configuration.
- ♦ **request-route-dit**: The name of the Directory Information Tree that the back-end server is hosting. The name must not conflict with the policy IDs. This is an optional attribute and is also referred to as ModDNCache. If this attribute is specified, the cache configuration is enabled and a request is routed to the back-end server that has the latest copy of the entry in the request.
Multiple LDAP Proxy servers can be configured to share the ModDNCache, so that information about any modification that occurs through any of the proxy is available on all the proxy servers.
- ♦ **moddn-cache-enable-for-bind**: This is an optional attribute. While specifying this attribute, you can configure the Proxy server to enable modify DN cache and have the user binds follow it, if the user object is present in the cache. However, if the user object is not present in the cache, you can configure LDAP Proxy to follow the default load balancing and routing mechanism.
- ♦ **moddn-cache-expiry-time**: This attribute is used to specify the duration for which the cache should be effective. You can set a value between 300 seconds and 86400 seconds. The cleanup thread event runs every two minutes, and if the administrator specifies 10 minutes as cache timeout, cache entries may get cleared between 10 to 12 minutes.

The following is a sample configuration that uses the `moddn-cache-expiry-time` attribute:

```
<policy-connection-route id-policy="conn-route-policy" request-route-
dit="ldap_proxy_tree" moddn-cache-enabled-for-bind="true" moddn-cache-expiry-
time="1800">
  <rule>
    Rule definition...
  </rule>
  <persistent-moddn-cache>
    <redis redis-auth-username="redis_admin">
      <addr-ipv4>192.168.1.2</addr-ipv4>
      <port>6379</port>
    </redis>
  </persistent-moddn-cache>
</policy-connection-route>
```

In the preceding example, a connection route policy `conn-route-policy` is defined and the `ModDNCache` and `moddn-cache-enabled-for-bind` attributes are enabled. An expiry time of 1800 seconds for the cache is also defined.

The `<policy-connection-route>` element can also have the following additional child elements:

- ◆ **<persistent-moddn-cache>**: If the LDAP Proxy server is restarted, the in-memory `ModDNCache` is erased. This causes further modify requests to be routed to any of the back-end servers, depending on the configuration of the load balancer. You can store the `ModDNCache` in a persistent storage to prevent loss of the data even if the LDAP Proxy server is restarted.

For information about configuring the Redis server, see [“Configuring the Redis Server” on page 81](#).

- ◆ **redis-auth-username**: This attribute is used when redis is configured for authentication. You must specify the username of the redis administrator and ensure that the password for this user is stored in the local secret store using the `passwdstore` utility.
- ◆ **<addr-ipv4>/<addr-ipv6>**: The IP address of the system on which LDAP Proxy is installed.
- ◆ **<port>**: The port on which the listener listens for incoming requests.

The following is a sample configuration of a persistable `ModDNCache` interface:

```
<list-policy>
  <policy-connection-route id-policy="conn-route-policy" request-route-
dit="ldap_proxy_tree" >
    <rule>
      Rule definition...
    </rule>
    <persistent-moddn-cache>
      <redis redis-auth-username="redis_admin">
        <addr-ipv4>127.0.0.2</addr-ipv4>
        <port>6379</port>
      </redis>
    </persistent-moddn-cache>
  </policy-connection-route>
```

Configuring a Hash-Based Route Policy

Hash-Based Route Policy is an optional policy which allows you to choose the load balancers based on priority. You can associate a connection route policy with a group of load balancers. The type of the load balancers may vary based on your need. You can have a group of priority based load balancers or a mix of different types of load balancers, as shown in the following example:

```

<list-load-balancer>
  <lb-conn-based id-load-balancer="backend-grp1">
    <ref-backend-server>Backend1</ref-backend-server>
    <ref-backend-server>Backend2</ref-backend-server>
  </lb-conn-based>

  <lb-dynamaic-load-based id-load-balancer="backend-grp2">
    <ref-backend-server>Backend3</ref-backend-server>
    <ref-backend-server>Backend4</ref-backend-server>
  </lb-conn-based>

  <lb-priority-based id-load-balancer="backend-grp3">
    <ref-backend-server>Backend2</ref-backend-server>
    <ref-backend-server>Backend1</ref-backend-server>
  </lb-priority-based>

</list-load-balancer>
<policy-connection-route id-policy="conn-route-policy">
  <rule>
    <actions>
      <do-use-route>
        <ref-hash-balancers num-buckets="3">
          backend-gr
        </ref-hash-balancers>
      </do-use-route>
    </actions>
    <actions-default>
      <do-nothing/>
    </actions-default>
  </rule>
</policy-connection-route>

```

In the above sample, the configuration includes a connection route policy `conn-route-policy` with a hash balancer pattern `backend-grp`. When this policy is configured, the LDAP Proxy expects three load balancers: `backend-grp1`, `backend-grp2`, and `backend-grp3`. If they are not available, then LDAP Proxy fails to start.

If you want to hash route all requests (including the bind requests), set the `hash-route-all` attribute to `true`, as shown in the following example:

```

<ref-hash-balancers num-buckets="3" hash-route-all="true">
  backend-grp
</ref-hash-balancers>

```

Configuring a Connection Route Policy to Block Anonymous Binds

To configure a Route Policy to block anonymous binds, set a condition in the connection route policy, as shown in the following example:

```

<policy-connection-route id-policy="conn-route-policy">
  <rule>
    <conditions>
      <if-bind-dn op="not-equal"></if-bind-dn>
    </conditions>
    <actions>
      <do-use-route>
        <ref-load-balancer>backend-grp1</ref-load-balancer>
      </do-use-route>
    </actions>
    <actions-default>
      <do-nothing/>
    </actions-default>
  </rule>
</policy-connection-route>

```

In the above example, a Connection Route policy identified as id-policy includes a condition to block anonymous binds and route all connections to the back-end server identified as backend-grp1. If bind dn is blank, the LDAP proxy will not do anything.

Configuring a Connection Route Policy for Routing a Request Based on Search Request

Routing based on search requests is evaluated only if the search request is on anonymous connection. In-order to handle the bind, policy chain should have at least one route which will be used for the non-search requests. To configure a connection route policy based on a search request is shown in the below example:

```

<policy-connection-route id-policy="sample_conn">
  <rule>
    <conditions>
      <or>
        <and>
          <if-message-type op="equal">ldap-search-request</if-message-type>
          <if-srch-base op="equal">o=acme</if-srch-base>
        </and>
        <if-message-type op="not-equal">ldap-search-request</if-message-type>
      </or>
    </conditions>
    <actions>
      <do-use-route>
        <ref-load-balancer>Default_LoadBalancer</ref-load-balancer>
      </do-use-route>
    </actions>
    <actions-default>
      <do-nothing/>
    </actions-default>
  </rule>
</policy-connection-route>

```

In the above example, anonymous search requests go to the Default_LoadBalancer when the search base is o=acme. Any request other than a search request, passes the condition.

Replacing a String Sequence in Directory Attribute Values (Replace String Policy)

The Replace String policy is an optional policy that is used to replace a string sequence in the attribute values of a directory.

The Replace String policy must be defined in the `<list-policy>` node of the XML configuration file.

The Replace String policy replaces the values from the object DN and the attributes specified in the policy. For instance, if you want to define a Replace String policy to replace the DN, manager, and also attribute values from `o=company` to `ou=marketing`, `o=company` and `o=subsidiary` to `o=company`, you can configure the policy, as follows:

```
<list-policy>
  <policy-replace-string id-policy="replace-string" disabled="false">
    <rule>
      <conditions>
        <or>
          <if-message-type op="equal">ldap-add-request</if-message-type>
          <if-message-type op="equal">ldap-add-response</if-message-type>
          <if-message-type op="equal">ldap-delete-request</if-message-type>
          <if-message-type op="equal">ldap-modify-request</if-message-type>
          <if-message-type op="equal">ldap-moddn-request</if-message-type>
          <if-message-type op="equal">ldap-search-request</if-message-type>
          <if-message-type op="equal">ldap-search-result-entry-response</if-message-
type>
          <if-message-type op="equal">ldap-search-result-entry-referenceresponse</if-
message-type>
          <if-message-type op="equal">ldap-bind-request</if-message-type>
          <if-message-type op="equal">ldap-bind-response</if-message-type>
          <if-message-type op="equal">ldap-compare-request</if-message-type>
          <if-message-type op="equal">ldap-search-result-done-response</if-message-
type>
        </or>
      </conditions>
      <actions>
        <do-replace-string>
          <attributes>
            <value>manager</value>
            <value>seeAlso</value>
          </attributes>
          <order>
            <!-- In the oreder. Will come out after first hit -->
            <replace>
              <from>o=comapny</from>
              <to>ou=marketing,o=company</to>
            </replace>
          </order>
        </do-replace-string>
      </actions>
    </rule>
  </policy-replace-string>
</list-policy>
```

```

    </replace>
    <replace>
      <from>o=subsidiary</from>
      <to>o=company</to>
    </replace>
  </order>
</do-replace-string>
</actions>
<actions-default>
  <do-nothing/>
</actions-default>
</rule>
</policy-replace-string>
</list-policy>

```

Handling Attribute OIDs in Policies

As indicated in RFC 4512 section 2.5, attributes can also be referred to by their OIDs. This means that policies can be easily bypassed when attribute OIDs are used in the incoming request.

LDAP Proxy provides a map file for default attribute names, which is called `nlp-schemaconf.xml`, located in the `/etc/opt/novell/ldapproxy/conf` directory. This file contains attribute name OID maps for the default schema provided by directories such as NetIQ eDirectory, Active Directory, Sun ONE, IBM Tivoli, and Oracle OID.

However, OIDs related to custom schemas are not supported and need to be handled manually. For example, to add an attribute name OID map for attributeTypes “2.16.840.1.113719.1.1.4.1.59.12 NAME 'myattribute' SYNTAX 1.3.6.1.4.1.1466.115.121.1.12”, add the following node to the `nlp-schemaconf.xml` file:

```

<attribute oid="2.16.840.1.113719.1.1.4.1.59.12 ">
  <name>myattribute</name>
</attribute>

```

Enabling Auditing

LDAP Proxy allows you to audit the activities on the proxy and back-end directory servers. For example, you can track session details, LDAP policies, and back-end activities. Proxy supports the traditional method of auditing as well as the XDAS-standards based auditing.

- ♦ [“Configuring Proxy Paths” on page 69](#)
- ♦ [“Configuring Audit Events Using XDAS” on page 70](#)
- ♦ [“Configuring Audit Events” on page 78](#)

Configuring Proxy Paths

The `<proxy-paths>` node is an optional node that defines the location of certain mandatory directories that are installed during proxy installation.

By default, the `<proxy-paths>` node is defined in the `nlpconf.xml` file as follows:

```

<proxy-paths>
  <dir-config> /etc/opt/novell/ldapproxy/conf</dir-config>
  <dir-log> /var/opt/novell/ldapproxy/log</dir-log>
</proxy-paths>

```

Configuration Parameters

The following elements and parameters are used to configure proxy paths:

- ◆ **<dir-config>**: The location of the `conf` directory. In the sample configuration, the location specified is `/etc/opt/novell/ldaproxy/conf`.
- ◆ **<dir-log>**: The location of the `log` file. In the sample configuration, the location specified is `/var/opt/novell/ldaproxy/log`.

Configuring Audit Events Using XDas

Though LDAP Proxy supports both traditional as well as the XDas standards-based auditing, NetIQ recommends that you use XDas auditing.

XDas auditing supports auditing through Syslog appender and file appender. Syslog appender supports event logging over UDP, TCP and SSL protocols. File appender supports event logging through rolling files.

The following is a sample configuration of XDas events:

```
<!--XDas configuration!-->
<proxy-xdas-config>
    <xdas-event>AUTHENTICATE_SESSION</xdas-event>
    <xdas-event>UNAUTHENTICATE_SESSION</xdas-event>
    <xdas-event>MODIFY_ACCOUNT</xdas-event>
</proxy-xdas-config>
```

The following table lists how traditional LDAP Proxy events are mapped to XDas events.

Table 2-1 Mapping LDAP Proxy Events to XDas Events

LDAP Events	Proxy Event ID	Proxy Events	XDas Events	XDas Event ID
LDAP Events	1442817	The LDAP Bind requests that are received.	AUTHENTICATE_SESSION	0.0.11.0
	1442818	The LDAP Bind responses that are sent.	AUTHENTICATE_SESSION	0.0.11.0
	1442819	The LDAP Unbind requests that are received.	UNAUTHENTICATE_SESSION	0.0.11.1
	1442820	The LDAP Search requests that are received.	QUERY_ACCOUNT, QUERY_DATA_ITEM_ATTRIBUTE, QUERY_ROLE	0.0.0.4, 0.0.2.2, 0.0.8.4
	1442821	The LDAP Search Result Entry responses that are sent.	QUERY_ACCOUNT, QUERY_DATA_ITEM_ATTRIBUTE	0.0.0.4, 0.0.2.2
	1442822	The LDAP Search Done responses that are sent	QUERY_DATA_ITEM_ATTRIBUTE	0.0.2.2

LDAP Events	Proxy Event ID	Proxy Events	XDAS Events	XDAS Event ID
	1442823	The LDAP Search Referral responses that are sent	QUERY_DATA_ITEM_ATTRIBUTE	0.0.2.2
	1442824	The LDAP Modify requests that are received	MODIFY_ACCOUNT, MODIFY_DATA_ITEM_ATTRIBUTE, MODIFY_ROLE	0.0.0.5, 0.0.2.3, 0.0.8.5
	1442825	The LDAP Modify responses that are sent	MODIFY_ACCOUNT, MODIFY_DATA_ITEM_ATTRIBUTE, MODIFY_ROLE	0.0.0.5, 0.0.2.3, 0.0.8.5
	1442826	The LDAP Add requests that are received	CREATE_ACCOUNT, CREATE_DATA_ITEM, CREATE_ROLE	0.0.0.0, 0.0.2.0, 0.0.8.0
	1442827	The LDAP Add responses that are sent.	CREATE_ACCOUNT, CREATE_DATA_ITEM, CREATE_ROLE	0.0.0.0, 0.0.2.0, 0.0.8.0
	1442828	The LDAP Delete requests that are received	DELETE_ACCOUNT, DELETE_DATA_ITEM, DELETE_ROLE	0.0.0.1, 0.0.2.1, 0.0.8.1
	1442829	The LDAP Delete responses that are sent	DELETE_ACCOUNT, DELETE_DATA_ITEM, DELETE_ROLE	0.0.0.1, 0.0.2.1, 0.0.8.1
	1442830	The LDAP Modify DN requests that are received	MODIFY_ACCOUNT, MODIFY_DATA_ITEM_ATTRIBUTE	0.0.0.5, 0.0.2.3
	1442831	The LDAP Modify DN responses that are sent	MODIFY_ACCOUNT, MODIFY_DATA_ITEM_ATTRIBUTE	0.0.0.5, 0.0.2.3
	1442832	The LDAP Compare requests that are received	QUERY_ACCOUNT, QUERY_DATA_ITEM_ATTRIBUTE, QUERY_ROLE	0.0.0.4, 0.0.2.2, 0.0.8.4
	1442833	The LDAP Compare responses that are sent.	QUERY_DATA_ITEM_ATTRIBUTE	0.0.2.2
	1442834	The LDAP Abandon requests that are received.	TERMINATE_DATA_ITEM_ASSOCIATION	0.0.6.1
	1442835	The LDAP Extended requests that are received	QUERY_DATA_ITEM_ATTRIBUTE, QUERY_ROLE	0.0.2.2, 0.0.8.4
	1442836	The LDAP Extended responses that are received.	QUERY_DATA_ITEM_ATTRIBUTE	0.0.2.2

LDAP Events	Proxy Event ID	Proxy Events	XDAS Events	XDAS Event ID
	1442837	The LDAP Extended intermediate responses that are received	QUERY_DATA_ITEM_ATTRIBUTE	0.0.2.2
	1442838	The LDAP Start TLS requests that are received	QUERY_DATA_ITEM_ATTRIBUTE	0.0.2.2
	1442839	The LDAP Start TLS responses that are sent	QUERY_DATA_ITEM_ATTRIBUTE	0.0.2.2
	1442840	The LDAP Stop TLS requests that are received	QUERY_DATA_ITEM_ATTRIBUTE	0.0.2.2
	1442841	The LDAP Unknown requests that are received.	QUERY_DATA_ITEM_ATTRIBUTE	0.0.2.2
	1442842	The LDAP Unknown responses that are received	QUERY_DATA_ITEM_ATTRIBUTE	0.0.2.2
Policy Events	1443073	The Connections that ar rejected	TERMINATE_SESSION	0.0.1.1
	1443074	The Requests that are denied	DESTROY_ACCESS_TOKEN	0.0.11.5
	1443075	The Routes that are not found for incoming requests	RESOURCE_UNAVAILABLE	0.0.9.4
	1443076	The Connection routes that are changed	MODIFY_SESSION	0.0.1.3
Back-end Events	1443329	The back- end servers whose status is changed to up.	ENABLE_SERVICE	0.0.3.5
	1443330	The back-end servers whose status is changed to down	DISABLE_SERVICE	0.0.3.4
	1443331	The back-end servers whose status is changed to slow	MODIFY_SERVICE_CONFIGURAT ION	0.0.3.3
	1443332	The servers in back-end group that are down	DISABLE_SERVICE	0.0.3.4

LDAP Events	Proxy Event ID	Proxy Events	XDAS Events	XDAS Event ID
	1443333	The back-end servers whose maximum connection limit has exceeded	REMOVE_SERVICE	0.0.3.1
	1443334	The LDAP Proxy System request sent to the back-end server	INVOKE_SERVICE	0.0.4.0
Session Events	1442561	The new sessions that are created	CREATE_SESSION	0.0.1.0
	1442562	The sessions that are terminated	TERMINATE_SESSION	0.0.1.1
	1442563	The sessions whose identity has been changed	MODIFY_SESSION	0.0.1.3
System Events	1442305	The LDAP Proxy systems that have been initialized	START_SYSTEM	0.0.9.0
	1442306	The LDAP Proxy systems that have been shut down	SHUTDOWN_SYSTEM	0.0.9.1
Event System Events	1442049	The event producers and consumers that are registered or deregistered	CONFIGURE_AUDIT_SERVICE	0.0.10.0
	1442050	The event producers and consumers that are registered or deregistered	CONFIGURE_AUDIT_SERVICE	0.0.10.0

Configuring the XDAS Audit Events

To configure XDAS audit events:

- 1 Open the `nlpconf.xml` file from the `/etc/opt/novell/ldaproxy/conf` directory in any XML editor.
- 2 Create an instance similar to the sample configuration. This `<proxy-xdas-config>` node must be defined after the `<proxy-paths>` node in the configuration file.
- 3 Use the `<xdas-event>` element to define the XDAS audit events.
For more information about the various events that can be monitored and their IDs, refer to [Table 2-5](#).
- 4 Save the `nlpconf.xml` file.

Configuring the XDASv2 Property File

When you install LDAP Proxy, the installer lays down the `xdasconfig.properties` file in the `/etc/opt/novell/ldaproxy/conf` directory.

The following is the content of the XDASv2 property file:

```
# Set the level of the root logger to DEBUG and attach appenders.
#log4j.rootLogger=debug, S, R

# Defines appender S to be a SyslogAppender.
#log4j.appender.S=org.apache.log4j.net.SyslogAppender

# Defines location of Syslog server.
#log4j.appender.S.Host=localhost
#log4j.appender.S.Port=port

# Specify protocol to be used (UDP/TCP/SSL)
#log4j.appender.S.Protocol=UDP

# Specify SSL certificate file for SSL connection.
# File path should be given with double backslash.
#log4j.appender.S.SSLCertFile=/etc/opt/novell/mycert.pem

# Minimum log-level allowed in syslog.
#log4j.appender.S.Threshold=INFO

# Defines the type of facility.
#log4j.appender.S.Facility=USER

# Defines caching for SyslogAppender.
# Inputs should be yes/no
#log4j.appender.S.CacheEnabled=no

# Cache location directory
# Directory should be available for creating cache files
#log4j.appender.S.CacheDir=/var/opt/novell/ldaproxy

# Cache File Size
# Cache File Size should be in the range of 50MB to 4000MB
#log4j.appender.S.CacheMaxFileSize=500MB

# Layout definition for appender Syslog S.
#log4j.appender.S.layout=org.apache.log4j.PatternLayout
#log4j.appender.S.layout.ConversionPattern=%c : %p%m%n

# Defines appender R to be a Rolling File Appender.
#log4j.appender.R=org.apache.log4j.RollingFileAppender

# Log file for appender R.
#log4j.appender.R.File=/var/opt/novell/ldaproxy/log/xdas-events.log

# Max size of log file for appender R.
#log4j.appender.R.MaxFileSize=100MB

# Set the maximum number of backup files to keep for appender R.
# Max can be 13. If set to zero, then there will be no backup files.
#log4j.appender.R.MaxBackupIndex=10

# Layout definition for appender Rolling log file R.
#log4j.appender.R.layout=org.apache.log4j.PatternLayout
#log4j.appender.R.layout.ConversionPattern=%d{MMM dd HH:mm:ss} %c : %p%m%n
```

Table 2-2 XDASv2 Property File

Options	ID
Syslog Appender	S
Rolling File Appender	R

The entries in the `xdasconfig.properties` file are not case sensitive, entries can appear in any order, empty lines are valid, and any line that starts with a hash (#) is commented out.

NOTE: If you add or delete any event in the `nlpconf.xml` file, restart LDAP Proxy for the changes to take effect.

The following table provides an explanation of each setting in the `xdasconfig.properties` file.

Table 2-3 XDAS Settings

Setting	Description
<code>log4j.rootLogger=debug, S, R</code>	Sets the level of the root logger to debug and attaches an appender named R or S, where S specifies a Syslog appender and R specifies a Rolling File appender.
<code>log4j.appender.S=org.apache.log4j.net.SyslogAppender</code>	Specifies the appender S to be a Syslog appender.
<code>log4j.appender.S.Host=localhost</code>	Specifies the location of the Syslog server where XDAS events are logged. For example, <code>log4j.appender.S.Host=192.168.0.1</code>
<code>log4j.appender.S.Port=port</code>	The port at which the XDAS connects to the Syslog server. The port supports values from 1 to 65535. If you specify an invalid value, the port defaults to 514. If the connection between XDAS and the Syslog server fails, Identity Manager cannot log events until the connection is restored.
<code>log4j.appender.S.Protocol=UDP</code>	Specifies the protocol to use. For example, UDP, TCP, or SSL.
<code>log4j.appender.S.SSLCertFile=/etc/opt/novell/mycert.pem</code>	Specifies the SSL certificate file for the SSL connection. Use double backslashes to specify the path of the file. This is an optional setting.
<code>log4j.appender.S.Threshold=INFO</code>	Specifies the minimum log level allowed in the Syslog appender. Currently, the INFO log level is supported.
<code>log4j.appender.S.Facility=USER</code>	Specifies the type of facility. The facility is used to try to classify the message. Currently, USER facility is supported. These values may be specified as upper or lower case characters.

Setting	Description
<code>log4j.appender.S.layout=org.apache.log4j.PatternLayout</code>	Layout setting for Syslog appender.
<code>log4j.appender.S.layout.ConversionPattern=%c : %p%m%n</code>	Layout setting for Syslog appender. For information about the conversion patterns and their descriptions, see logging.apache.org .
<code>log4j.appender.R=org.apache.log4j.RollingFileAppender</code>	Specifies appender R to be a Rolling File appender
<code>log4j.appender.R.File=/var/opt/novell/ldaproxy/log/xdas-events.log</code>	The location of the log file for a Rolling File appender.
<code>log4j.appender.R.MaxFileSize=100MB</code>	The maximum size, in MBs, of the log file for a Rolling File appender. Set this value to the maximum size that the client allows.
<code>log4j.appender.R.MaxBackupIndex=10</code>	Specify the maximum number of backup files for a Rolling File appender. The maximum number of the backup files can be 10. A zero value means no backup files.
<code>log4j.appender.R.layout=org.apache.log4j.PatternLayout</code>	Layout setting for Rolling File appender.
<code>log4j.appender.R.layout.ConversionPattern=%d{MMMMdd HH:mm:ss} %c : %p%m%n</code>	Layout setting for Rolling File appender. See Table 2-4 on page 76 for simple date format patterns. For information about the conversion patterns and their descriptions, see logging.apache.org

The following examples illustrate the date and time patterns interpreted in the U.S. The given date and time are 2012-07-04 12:08:56 local time in the U.S. Pacific Time time zone.

Table 2-4 Date and Time Pattern Example

Date and Time Pattern	Result
<code>"yyyy.MM.dd G 'at' HH:mm:ss z"</code>	2012.07.04 AD at 12:08:56 PDT
<code>"EEE, MMM d, 'yy"</code>	Wed, Jul 4, '01
<code>"h:mm a"</code>	12:08 PM
<code>"hh 'o'clock' a, zzzz"</code>	12 o'clock PM, Pacific Daylight Time
<code>"K:mm a, z"</code>	0:08 PM, PDT
<code>"yyyyy.MMMMM.dd GGG hh:mm aaa"</code>	02012.July.24 AD 12:08 PM
<code>"EEE, d MMM yyyy HH:mm:ss Z"</code>	Wed, 24 Jul 2012 12:08:56 -0700
<code>"yyMMddHHmmssZ"</code>	120724120856-0700
<code>"yyyy-MM-dd'T'HH:mm:ss.SSSZ"</code>	2012-07-04T12:08:56.235-0700

Enabling Syslog Appender

You can use the Syslog appender, if you want centralize the auditing messages at one place. Additionally, a Syslog server offers better backup support in the event of a disaster.

To enable the Syslog appender, make the following changes in the `xdasxconfig.properties` file:

- 1 Change the following entry to S to attach a Syslog appender:

```
log4j.rootLogger=debug, S
```

- 2 Uncomment the following entries:

```
log4j.appender.S=org.apache.log4j.net.SyslogAppender
```

```
log4j.appender.S.Host=localhost
```

```
log4j.appender.S.Port=port
```

```
log4j.appender.S.Protocol=UDP
```

```
log4j.appender.S.SSLCertFile=/etc/opt/novell/mycert.pem
```

```
#log4j.appender.S.Threshold=INFO
```

```
#log4j.appender.S.Facility=USER
```

```
#log4j.appender.S.layout=org.apache.log4j.PatternLayout
```

```
#log4j.appender.S.layout.ConversionPattern=%c : %p%m%n
```

- 3 Restart `nlpd`.

Enabling Rolling File Appender

The File appender is preferred, if the auditing solution is limited to an individual server. Also, it is easy to bring up this solution because the number of components to be setup are few and thus, is more suited for demonstrations.

To enable the Rolling File appender, make the following changes in the `xdasxconfig.properties` file:

- 1 Change the following entry to R to attach a Rolling File appender.

```
log4j.rootLogger=debug, R
```

- 2 Uncomment the following entries:

```
log4j.appender.R=org.apache.log4j.RollingFileAppender
```

```
log4j.appender.R.File=/var/opt/novell/ldaproxy/log/xdas-events.log
```

```
log4j.appender.R.MaxFileSize=100MB
```

```
log4j.appender.R.MaxBackupIndex=10
```

```
log4j.appender.R.layout=org.apache.log4j.PatternLayout
```

```
log4j.appender.R.layout.ConversionPattern=%d{MMM dd HH:mm:ss} %c : %p%m%n
```

- 3 Restart `nlpd`.

Enabling XDAS Event Caching

LDAP Proxy allows you to optionally store XDAS events locally on the agent in a Syslog Appender cache. With events cached, if the agent cannot communicate with the auditing server, the audit events generated are retained, ensuring that audit data is not lost. The agent then attempts to re-send the cached events when the agent computer can once again communicate with the auditing server.

XDAS event caching is disabled by default. To enable event caching, complete the steps below.

- 1 On the agent computer, navigate to the location of the XDASv2 property file. The `xdasconfig.properties` file is located at `/etc/opt/novell/ldaproxy/conf/xdasconfig.properties` by default.
- 2 Use a text editor to open the `xdasconfig.properties` file.
- 3 Within the property file, navigate to the `log4j.appender.S.CacheEnabled` property and change the property value to `yes`.
- 4 If you want to cache events in a specific directory, modify the value of the `log4j.appender.S.CacheDir` property to specify the directory path. The default path is `/var/opt/novell/ldaproxy`. If you specify a directory, ensure that the directory path is a valid location on the server. If the specified path does not exist, the Syslog Appender logs events to the default location.
- 5 If you want to specify a custom file size for the cache, modify the value of the `log4j.appender.S.CacheMaxFileSize` property. The default value is 100 MB. The minimum value should be 50 MB, with a maximum value of 4 GB.
- 6 Save and close the `xdasconfig.properties` file.
- 7 Restart `nlpd`.

Configuring Audit Events

You can configure a specific set of events in traditional auditing or XDAS auditing individually and configure both these auditing systems together.

It enables you to monitor all the user activities that occur in the proxy. This helps you to track user activities including local activities such as LDAP requests, back-end server status, policy actions, configuration changes, and session details. This helps to detect and resolve potential problems before they arise, so that users are not denied access to critical services.

The proxy configuration allows you to specify the kind of events that must be audited. The following types of events can be monitored:

- ♦ LDAP Events
- ♦ Policy Events
- ♦ Back-end Events
- ♦ Session Events
- ♦ System Events
- ♦ Event System Events

You can configure all the events to be monitored by using the `<proxy-audit-config>` node in the configuration file. However, this is an optional configuration.

The following is a sample configuration for defining audit events. The events to be monitored are specified by using the `<event-id>` element. The sample configuration monitors events with event-ids 1442305 and 1442306, which means to monitor the LDAP Proxy systems that are initialized and shut down:

```
<proxy-audit-config audit-file-size-mb="512">
  <event-id>1442305</event-id>
  <event-id>1442306</event-id>
</proxy-audit-config>
```

To configure audit events:

- 1 Open the `nlpconf.xml` file from the `/etc/opt/novell/ldaproxy/conf` directory in any XML editor.
- 2 Create an instance similar to the sample configuration. This `<proxy-audit-config>` node must be defined after the `<proxy-paths>` node in the configuration file.
- 3 Use the `<event-id>` element to define the audit events.
For more information about the various events that can be monitored and their IDs, refer to [Table 2-5](#).
- 4 (Optional) Specify the file size of the audit log file in the `audit-file-size-mb` attribute. The default file size is 1 GB. If you do not want to specify the file size, you can remove this element from the configuration.
- 5 Save the `nlpconf.xml` file.

Table 2-5 Audit Events

Category	Event-id	Description
LDAP Events	1442817	The LDAP Bind requests that are received.
	1442818	The LDAP Bind responses that are sent.
	1442819	The LDAP Unbind requests that are received.
	1442820	The LDAP Search requests that are received.
	1442821	The LDAP Search Result Entry responses that are sent.
	1442822	The LDAP Search Done responses that are sent.
	1442823	The LDAP Search Referral responses that are sent.
	1442824	The LDAP Modify requests that are received.
	1442825	The LDAP Modify responses that are sent.
	1442826	The LDAP Add requests that are received.
	1442827	The LDAP Add responses that are sent.
	1442828	The LDAP Delete requests that are received.
	1442829	The LDAP Delete responses that are sent.
	1442830	The LDAP Modify DN requests that are received.
	1442831	The LDAP Modify DN responses that are sent.
	1442832	The LDAP Compare requests that are received.

Category	Event-id	Description
	1442833	The LDAP Compare responses that are sent.
	1442834	The LDAP Abandon requests that are received.
	1442835	The LDAP Extended requests that are received
	1442836	The LDAP Extended responses that are received.
	1442837	The LDAP Extended intermediate responses that are received.
	1442838	The LDAP Start TLS requests that are received.
	1442839	The LDAP Start TLS responses that are sent.
	1442840	The LDAP Stop TLS requests that are received.
	1442841	The LDAP Unknown requests that are received.
	1442842	The LDAP Unknown responses that are received.
Policy Events	1443073	The Connections that are rejected.
	1443074	The Requests that are denied.
	1443075	The Routes that are not found for incoming requests.
	1443076	The Connection routes that are changed.
Back-end Events	1443329	The back-end servers whose status is changed to up.
	1443330	The back-end servers whose status is changed to down.
	1443331	The back-end servers whose status is changed to slow.
	1443332	The servers in back-end group that are down.
	1443333	The back-end servers whose maximum connection limit has been exceeded.
	1443334	The LDAP Proxy System request sent to the back-end server.
Session Events	1442561	The new sessions that are created.
	1442562	The sessions that are terminated.
	1442563	The sessions whose identity has been changed.
System Events	1442305	The LDAP Proxy systems that have been initialized.
	1442306	The LDAP Proxy systems that have been shut down.
Event System Events	1442049	The event producers and consumers that are registered or deregistered.
	1442050	The event producers and consumers that register or deregister events.

Configuring Logging

LDAP Proxy monitors the various LDAP operations performed by the listeners and back-end servers, including the number of Bind requests received by listeners, the number of Bind requests received by back-end servers, and the number of Search requests encountered by back-end servers.

To obtain this monitored events data, LDAP Proxy enables you to configure the monitor policy that defines log files.

The `<proxy-stat-log-config>` node in the configuration file specifies the monitoring policy, such as the file-size limit of the log file being created and also the time interval for updating log files. However, this is an optional configuration.

The following is a sample `<proxy-stat-log-config>` node configuration:

```
<proxy-stat-log-config>
  <logfile-size-limit>102400</logfile-size-limit>
  <log-interval>120</log-interval>
</proxy-stat-log-config>
```

To configure the stat log:

- 1 Open the `nlpconf.xml` file from the `/etc/opt/novell/ldapproxy/conf` directory in any XML editor.
- 2 Create an instance similar to the sample configuration. This `<proxy-stat-log-config>` node must be defined after the `<proxy-audit-config>` node in the configuration file.
- 3 Specify the following:
 - ♦ **<logfile-size-limit>**: The maximum size of the log file in kilobytes (KB). The default file size is 102400.
 - ♦ **<log-interval>**: The time interval, in seconds, to log monitoring information. The default value is 60, which is also the minimum value that you can set.
- 4 Save the `nlpconf.xml` file.

Configuring the Redis Server

LDAP Proxy 1.5 includes the Redis server rpms for SUSE Linux Enterprise Server (SLES) and RedHat Linux.

- 1 Install the appropriate rpm for your operating system.
- 2 Start the Redis server, by using the `/etc/init.d/redis start` command.

By default, the Redis server configuration file is located in the `/etc/redis/redis.conf` folder.

For more information about configuring Redis server, refer to the [Redis documentation \(http://redis.io/documentation\)](http://redis.io/documentation).

Validating Configuration File

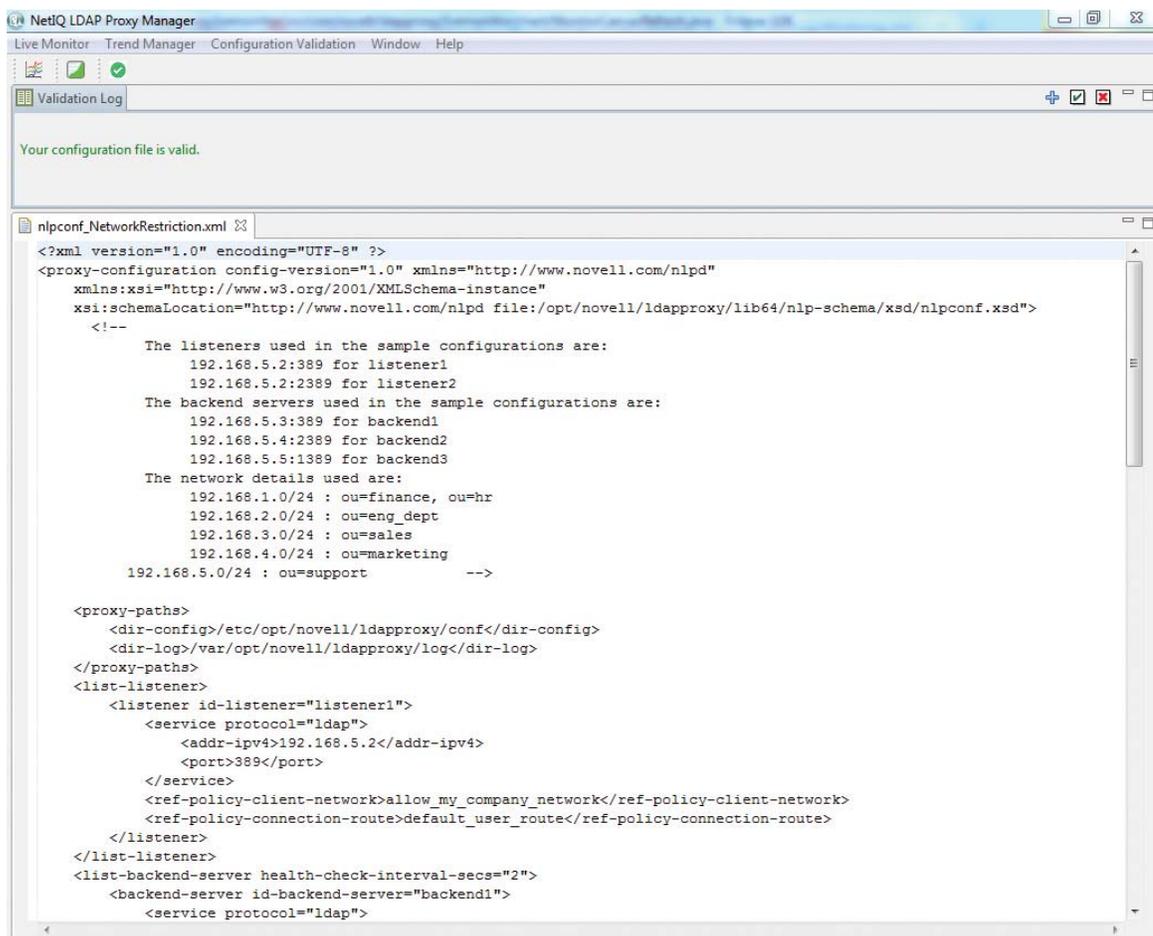
Once your configuration file is ready, you can validate the file using NLPManager. Using this option, you can check for errors (if any) in your configuration file. To validate the configuration file, perform the following steps:

- 1 Click on the Validation icon  at the top of the NLPManager utility.
- 2 Click on the Open File  icon from the **Validation Logs** tab.
- 3 Browse and upload the configuration file (XML), then click Validate  icon.
- 4 The Validation Logs section displays the details of the errors in your configuration file. If there are no error, you will see the following message displayed:

Your configuration file is valid.

- 5 Click on the Close  icon to close the Validation panel.

Figure 2-1 XML Validation Log



3 Monitoring LDAP Proxy Processes

All internal activities of NetIQ LDAP Proxy can be monitored by enabling logging and tracing.

NetIQ LDAP Proxy enables you to configure a trace based on modules. Every traced activity message is associated with a severity level, which helps you to determine how critical a message is.

- ♦ **Critical:** A critical message that needs the user's action immediately.
For example, the server ran out of memory or the listener failed to listen on a given interface.
- ♦ **Error:** An error message that does not directly affect the functioning of the LDAP proxy server.
For example, any kind of operational errors.
- ♦ **Warning:** A warning message that needs the user's attention.
For example, the back-end server is down or the maximum connection limit for back-end service is reached.
- ♦ **Info:** An informational message that can be understood by users.
For example, all module initialization messages.
- ♦ **Debug:** Debugging information that can be understood only by developers or administrators.
For example, `IN-CONN received socket error, closing LDAP connection`.

You can configure a trace for the following modules:

- ♦ **TPOOL:** Logs thread pool events.
- ♦ **SOCKET:** Logs socket events.
- ♦ **SESSION:** Logs session events.
- ♦ **MONITOR:** Logs monitor thread events.
- ♦ **BER:** Logs LDAP encoding and decoding events.
- ♦ **LDAP:** Logs LDAP events.
- ♦ **POLICY:** Logs policy events.
- ♦ **BACKEND:** Logs back-end events.
- ♦ **XML:** Logs XML events.
- ♦ **CONFIG:** Logs configuration events.
- ♦ **STAT:** Logs statistics logger events.

Additionally, you can configure certain parameters that are used to log additional control information with every message:

- ♦ **Time:** Logs the time when the activity occurred. By default, time is enabled and logged.
- ♦ **Severity:** Logs the message severity levels. By default, severity is disabled.
- ♦ **Session:** Logs the session details, including session ID and thread ID. By default, this parameter is disabled.
- ♦ **Client address:** Logs the client address where the activity occurred. By default, this parameter is disabled.
- ♦ **Inline:** Logs messages in the same thread. By default, this parameter is disabled.

To log information for these parameters, you must set the parameter to "true" while configuring the proxy trace. For example, to enable session, you define it as `session="true"`.

To enable trace configuration:

- 1 Open the `nlptraceconf.xml` file from the `/etc/opt/novell/ldaproxy/conf` directory in any XML editor.

The `conf` directory is available on the machine where you installed NetIQ LDAP Proxy.

- 2 Look for the following trace configuration in the file:

```
<config client-addr="false" inline="false" session="true" severity="false" time="true" trace-file-name="nlptrace.log" trace-file-size-kb="1024">
  <module log-level="Debug">LDAP</module>
  <module log-level="Info">BACKEND</module>
</config>
```

- 3 To enable the proxy trace, remove the comments (`<!-- !-->`) in the configuration.
- 4 To add or remove tracing of modules and parameter information, change the configuration according to your requirements.

For example, to enable tracing of the socket module, add the `<module log-level="Info">SOCKET</module>` element to the configuration.

- 5 Save the `nlptraceconf.xml` file in the `/etc/opt/novell/ldaproxy/conf` directory.
- 6 To commit the changes to the LDAP Proxy, run the following command:
 - ♦ `/etc/init.d/nlpd refresh` (On SLES 11 and RHEL 6)
 - ♦ `systemctl reload nlpd.service` (On SLES 12 and RHEL 7)

The trace log files are located in the `/var/opt/novell/ldaproxy` directory.

The following sample shows the trace message format:

```
[Time] SessionID:ThreadID TAGS: LEVEL: [Client Address] Message String
[2009/06/04 16:15:17.981] 1:3067648928 LDAP: INFO: [192.168.1.1:50167] OUT-CONN
sending request to backend service ldap://192.168.1.3:1389.
```

NOTE: The log-level value in the `nlptrace.conf` file is case-sensitive. It does not return the desired results if log-level value is specified entirely in lowercase or uppercase. For example, if you specify "debug" or "DEBUG" instead of "Debug", it does not work. The following example has the correct format:

```
<xsd:enumeration value="Critical"/>
<xsd:enumeration value="Error"/>
<xsd:enumeration value="Warning"/>
<xsd:enumeration value="Info"/>
<xsd:enumeration value="Debug"/>
```

4 Configuring Monitoring and Trending Activities

You can enable the live monitoring and trending feature of NetIQ LDAP Proxy by configuring the events to be monitored through NLPManager. The NLPManager also allows you to manage the trend analysis for LDAP events.

- ◆ [“Configuring Monitoring Activities” on page 85](#)
- ◆ [“Managing Trend Analysis” on page 88](#)
- ◆ [“Enabling Monitoring and Trending” on page 92](#)
- ◆ [“Restoring Monitoring and Trending Configuration” on page 95](#)

Configuring Monitoring Activities

You can configure live monitoring of ongoing activities on the LDAP proxy and back-end directory servers. You can configure multiple proxy servers for live monitoring.

To use the live monitoring functionality, ensure that you have fulfilled the following prerequisites:

- ◆ Ensure that the proxy server you want to monitor is running.
- ◆ Ensure that a dedicated listener is configured to support monitoring. This listener must include a Connection Route policy that will reference a Search Request policy. To enable live monitoring, you must define the `<do-send-monitor-response>` element in this Search Request policy. For more information about the `<do-send-monitor-response>` element, refer to [“Example 3” on page 60](#) in [“Evaluating Incoming Search Request \(Search Request Policy\)” on page 58](#).

1 Run the `./NLPManager` command to start NLPManager.

The NetIQ LDAP Proxy Manager window is displayed.

2 Do one of the following:

- ◆ Click the  icon.
- ◆ In the **Live Monitor** menu, click **Live Monitor**.

The **Monitor Manager Configuration** tab is displayed in the Editor pane.

3 Configure the proxy server to be monitored:

3a Click the  icon in the **Proxy Monitor Tree**.

The Proxy Server Details page is displayed.

Add Proxy Server

Proxy Server Details
Specify the details of your LDAP Proxy Server. Required fields are denoted by "**".

Generic Properties

IP Address : *

Port : * 389

Use Anonymous Login

Login Properties

User DN : *
(Example: cn=admin,o=company)

Password : *

Enable TLS Connection

TLS Certificate

Certificate File : *
Browse File...

Test Connection Restore Defaults

< Back Next > Finish Cancel

3b On the Proxy Server Details page, provide the following information:

- ◆ **DNS or IP Address:** The IP address of the listener that is configured to enable monitoring.
- ◆ **Port:** The port number of the listener that is configured for monitoring.
- ◆ **Use Anonymous Login:** Select this option to get the monitoring statistics anonymously. If you select this option, you do not need to specify the User DN and Password.
- ◆ **User DN:** The user credential of the back-end server configured on the specified listener.
- ◆ **Password:** The password to be used for authenticating the specified user credentials.

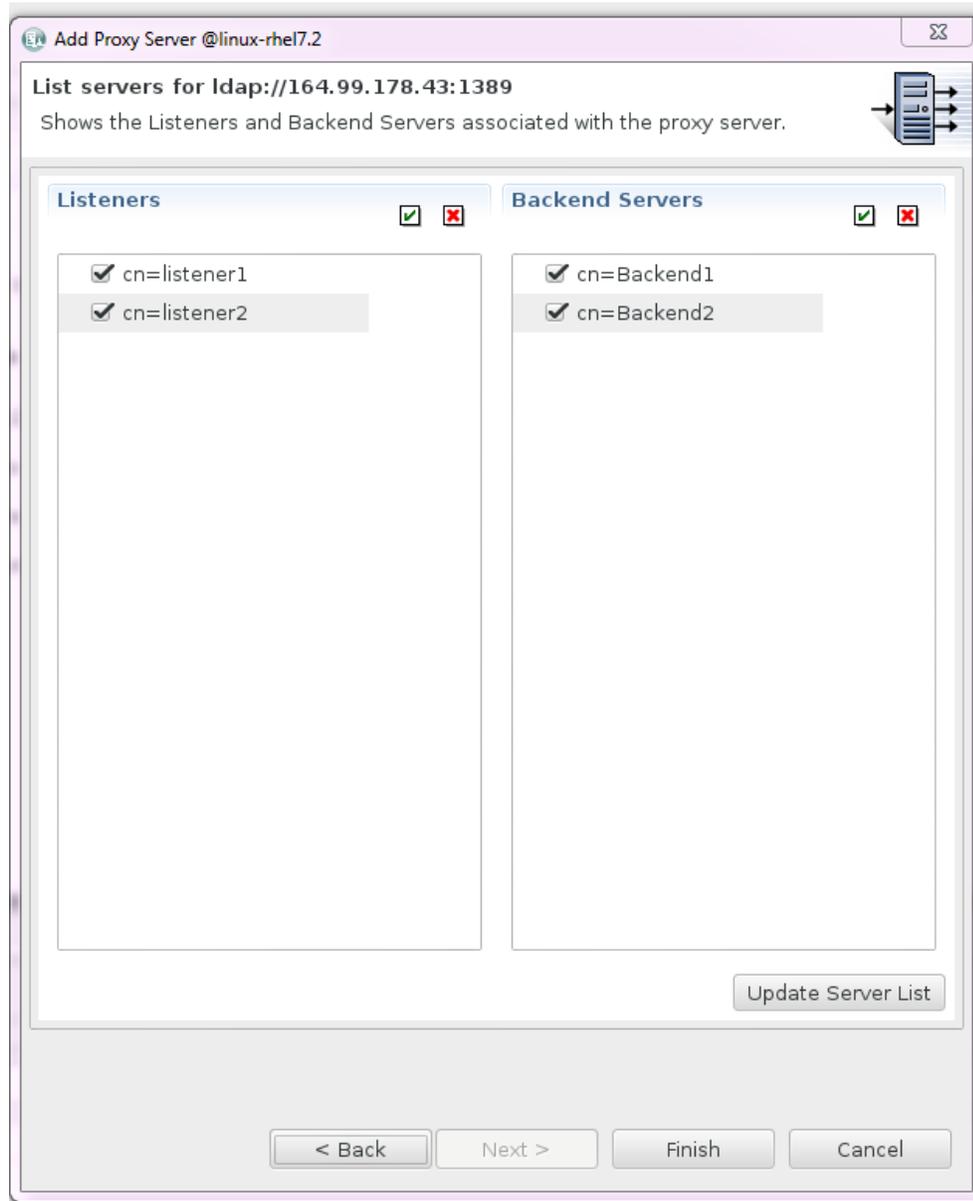
Currently, the TLS Certificate section is disabled.

- 3c** (Optional) To validate the connection and test whether the specified proxy server supports monitoring, click **Test Connection**.

If the validation fails, an error message appears at the top of the Proxy Server Details page.

- 3d** Click **Next**. A validation is performed to authenticate the connection and confirm that the specified proxy server supports monitoring.

The List servers for ldap page is displayed. This page lists all the listeners and back-end servers that are configured to the specified proxy server.



- 3e** Select the listeners and back-end servers that you want to monitor and click **Finish**.

By default, all the listeners and back-end servers are selected.

- 3f** Click **Next**.

An hierarchical representation of the configuration is displayed in the Proxy Monitor Tree section.

4 Start live monitoring:

4a Right-click the listener or back-end server that you want to monitor and click **Start Monitor**.

The Showing Monitor Data page is displayed. This page displays an empty graph with Time Interval (In Seconds) and Number of Operation(s) as its X and Y axes.

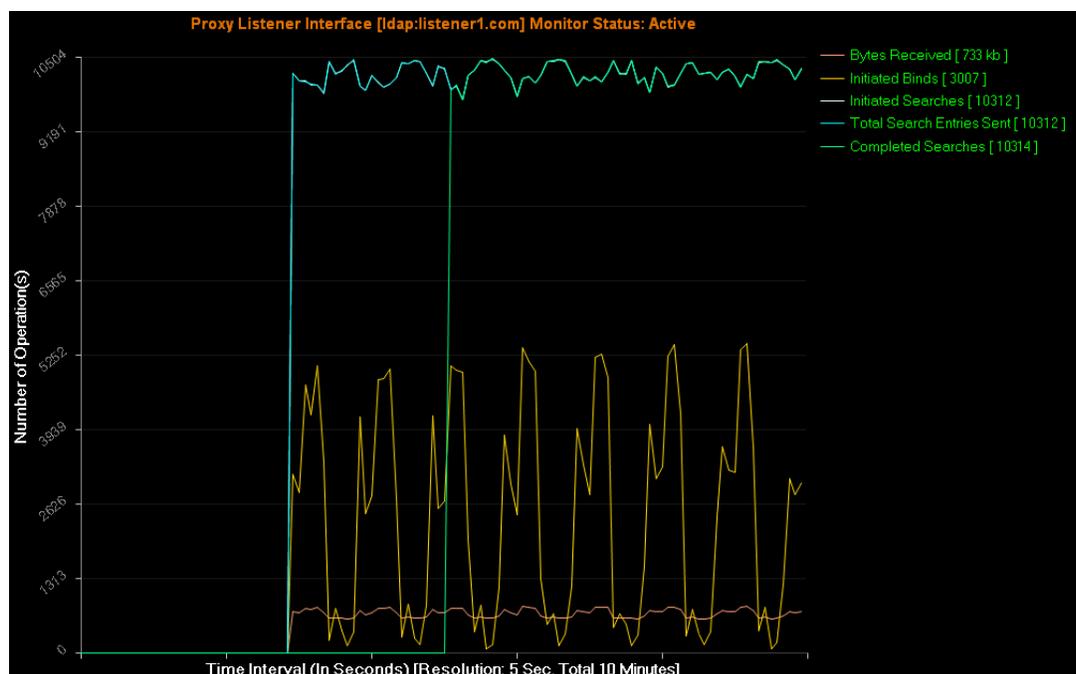
4b To define the events that must to be monitored, right-click the graph and click **Monitor Options**.

The Events Monitor Wizard is displayed. It displays a list of events that can be monitored, grouped into categories.

4c Select the events by clicking each category and selecting the check box corresponding to each event.

For more information about the listener and back-end events that can be monitored, refer to [Table 4-1](#).

The graph displays the live monitoring statistics of the selected events for every five seconds.



4d To stop monitoring on a listener or back-end server, right-click the listener or back-end server and click **Stop Monitor**. Alternatively, right-click the graph and click **Stop Monitor**.

5 (Optional) To remove a proxy server from the Monitor Tree Home, right-click the server and click **Remove Proxy Server**.

6 (Optional) To save the monitoring configuration, click the  icon at the top.

Managing Trend Analysis

The LDAP Trend Manager is an advanced LDAP event trend analysis tool that allows you to analyze the trend log files through a graphical representation. You can generate a graphical view of each server activity based on different time interval and LDAP events.

1 Run the `./NLPManger` command to start the NLPManger.

The LDAP Proxy window is displayed.

2 Do one of the following:

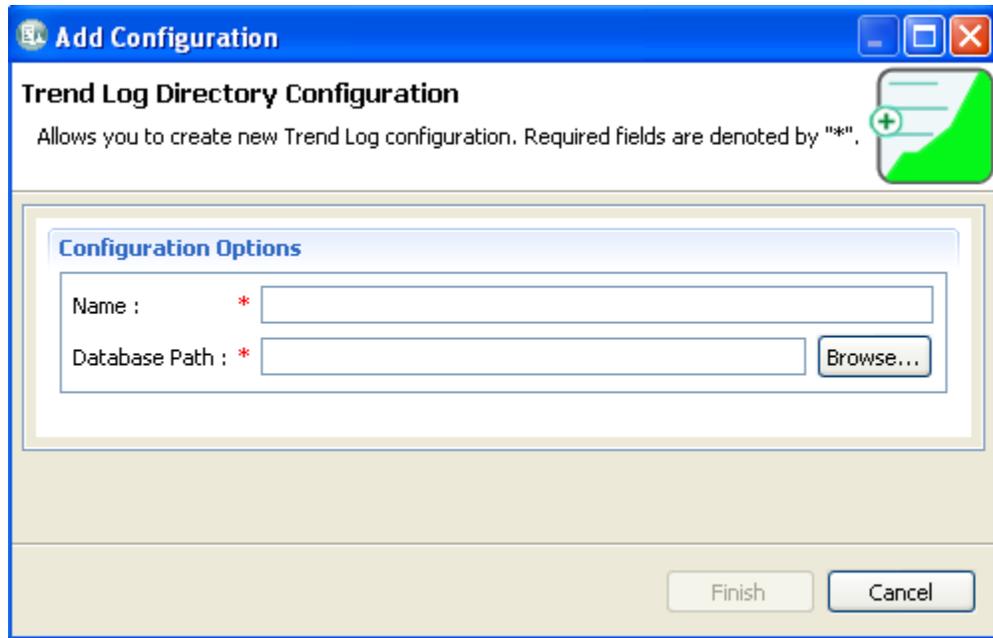
- ◆ Click the  icon.
- ◆ In the **Trend Manager** menu, click **Trend Manager**.

The **Trend Manager Configuration** tab is displayed.

3 Create a trend configuration:

3a Click the  icon.

The Add Configuration dialog is displayed.



Add Configuration

Trend Log Directory Configuration

Allows you to create new Trend Log configuration. Required fields are denoted by "*".

Configuration Options

Name : *

Database Path : *

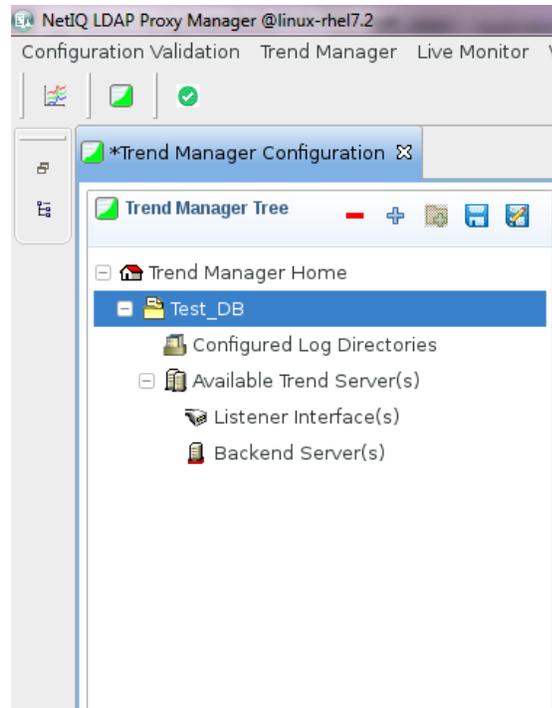
3b In the Add Configuration window, provide the following information:

- ◆ **Name:** A name for the trend configuration. This name is used as a reference for this configuration.
- ◆ **Database Path:** The directory where the analyzed trend data must be stored. The same directory location must be used in future to analyze the trend or to add more trend data in the new or updated proxy server log file.

If you have two instances of the Trend Manager tool running, the same database path cannot be used in both instances.

3c Click **Finish**:

An hierarchical representation of the configuration is displayed in the Trend Manager Tree.



4 Specify the log files to be analyzed:

4a Right-click **Configured Log Directories** and click **Add Log Directory**.

The Select Trend Server Log Directory window is displayed.

4b Specify the directory that contains the proxy server trend log files.

If the Trend Manager tool and proxy server are running on different systems, you can manually copy the log files from the system where you have configured the proxy server (`/var/opt/novell/ldapproxy/log`) to your local machine. This location must be defined in the LDAP proxy configuration.

The specified log directory is added to the Configured Log Directories tree hierarchy. You can specify multiple log files directories.

5 To analyze the server trend data, right-click the directory to be analyzed and click **Open Log Analyzer**.

The **Log Directory** tab is displayed. This tab displays the details of all the listener and back-end server log files available in the specified directory. The following details are specified:

- ◆ **Normalize Status:** Whether the log file was processed by the Trend Manager tool. The status can be Not Done, Partial, or Done.
The first time you configure a log directory, the status shows as Not Done. When the log file is processed completely, the status changes to Done, or changes to Partial.
- ◆ **Server Type:** Whether the log file is for a listener or back-end server.
- ◆ **Start Date/Time:** The date and time when the trend data was first logged into the file.
- ◆ **End Date/Time:** The date and time when the last trend data was logged into the file. This field is empty if the proxy server is still logging data into the file.
- ◆ **Log File Status:** Whether the proxy server trend data is still being logged into the file. The status can be Complete or On Going.

If you copied the log file from the server while information was being logged into it, this status shows as On Going. If the log file was complete when you copied it from the server, the status shows as Complete.

- ◆ **Log File Size:** The size of the log file.
- ◆ **File Name:** The name of the log file.

6 Select the file that you want to analyze and click **Process Selected Log(s)**.

The data in the log file is processed and the available servers are added to the Available Trend Server node of the Trend Manager Tree. For instance, if you process a back-end server log file, the individual back-end server details are added as child entry to the back-end server node of the tree.

IMPORTANT: Do not exit the Trend Manager tool during the log file processing because all trend information will be lost. If it is necessary to cancel processing, click **Cancel** in the **Data Processing Information** section.

7 (Optional) To update the log file information, click **Update Log Files**.

You might want to update the log file when you copy additional proxy server trend log files to the same local directory at a later time. The Trend Manager tool can identify and append the partially processed log file.

8 To analyze the data, right-click the listener or back-end server for which you want to analyze the trend, then click **Analyze Trend**.

The Showing Trend Data page is displayed. This page displays an empty graph with Date/Time and Number/units as its X and Y axes.

9 To define the LDAP operations that must be used for trending, right-click the graph, then click **Trend Options**.

The Events Selection Wizard window is displayed. It displays a list of events that can be used for trending, grouped into categories.

10 Select the events by clicking each category and selecting the check box corresponding to each event.

For more information about the listener and back-end events that can be monitored, see [Table 4-1](#).

11 Specify the time period for trending:

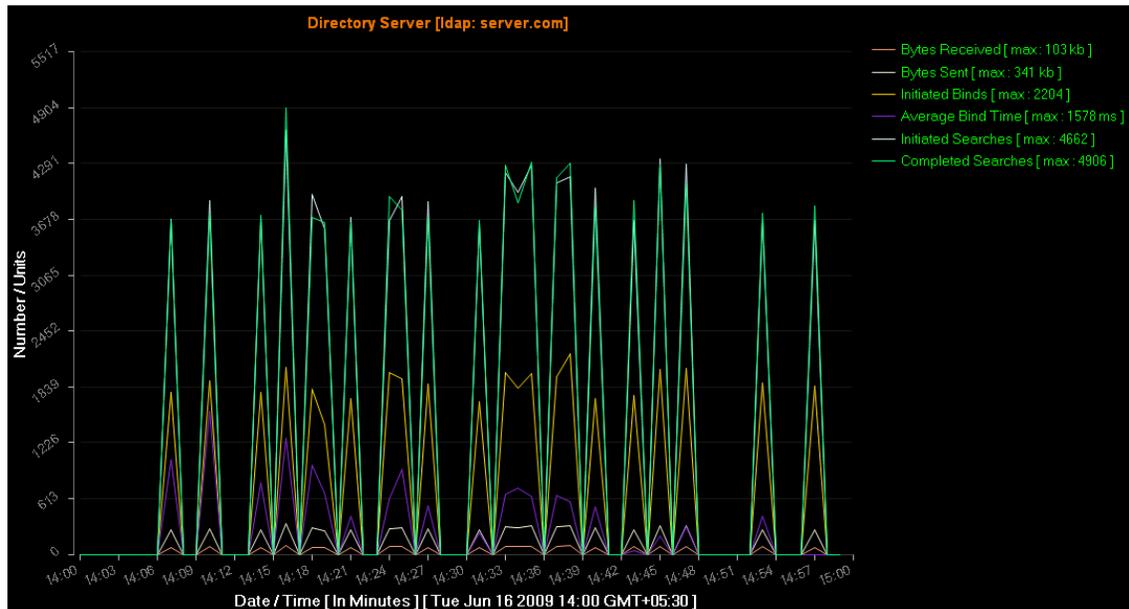
11a Select the time unit in the **Trend Unit** drop-down list. The available options are Hour, Day, Week, Month, Quarter, and Year.

11b In the **Start Date/Time** field, click the  icon to specify the time period.

For example, if you specify the time unit as Week, specify the first day of the week from the calendar and the end date is set automatically.

12 Click **Show Trend**.

The graph displays the trending statistics of the selected LDAP operations for the specified time period. Each event can be easily identified by the color code assigned to it.



13 (Optional) To save the trending configuration, click the  icon at the top.

Enabling Monitoring and Trending

When you enable live monitoring and trending, you must specify the events to be analyzed. [Table 4-1](#) provides the list of event categories and events that can be monitored and analyzed.

Table 4-1 Live Monitoring and Trending Events

Event Category	Event	Description
Connection Options This category is applicable only for the listener interface.	Total Accepted Connections	The number of LDAP connections accepted by the LDAP Proxy server.
	Total Rejected Connections	The number of LDAP connections rejected by the LDAP Proxy server.
	Current Accepted Connections	The number of accepted connections that are active.
TLS Options	Initiated Start TLS	The number of LDAP Start TLS requests initiated by an application or user.
	Completed StartTLS	The number of initiated LDAP Start TLS requests that were successful.
	Failed StartTLS	The number of initiated LDAP Start TLS requests that failed.
Generic Options	Bytes Received	The total amount of data in kilobytes received by server.

Event Category	Event	Description
	Bytes Sent	Total number of data in kilobytes sent by the server.
Bind and Unbind Options	Initiated Binds	The number of LDAP Bind requests initiated by an application/user.
	Failed Binds	The number of LDAP Bind requests that were initiated but failed.
	Completed Binds	The number of LDAP Bind requests that were initiated and successful.
	Average Bind Time	The average time taken by LDAP Bind in milliseconds.
	Anonymous Binds	The number of anonymous LDAP Bind requests initiated by an application or user.
	Initiated Unbinds	The number of LDAP Unbind requests initiated by an application or user.
	Initiated Abandons	The number of LDAP Abandon requests initiated by an application or user.
Add Options	Initiated Adds	The number of LDAP Add requests initiated by an application or user.
	Completed Adds	The number of LDAP Add requests that were initiated and successful.
	Failed Adds	The number of LDAP Add requests that were initiated but failed.
	Average Add Time	The average time taken by LDAP Add requests in milliseconds.
Compare Options	Initiated Compares	The number of LDAP Compare requests initiated by an application or user.
	Completed Compares	The number of LDAP Compare requests that were initiated and successful.
	Failed Compares	The number of LDAP Failed requests that were initiated but failed.
	Average Compare Time	The average time taken by LDAP Compare requests in milliseconds.
Delete Options	Initiated Deletes	The number of LDAP Delete requests initiated by an application or user.
	Completed Deletes	The number of LDAP Delete requests that were initiated and successful.
	Failed Deletes	The number of LDAP Failed requests that were initiated but failed.
	Average Delete Time	The average time taken by LDAP Delete requests in milliseconds.
Extended Operation Options	Initiated Extended Operations	The number of LDAP Extended operations initiated by an application or user.

Event Category	Event	Description
	Completed Extended Operations	The number of LDAP Extended operations that were initiated and successful.
	Failed Extended Operations	The number of LDAP Extended operations that were initiated but failed.
	Average Extended Operation Time	The average time taken by LDAP Extended operations in milliseconds.
Modify Options	Initiated Modifies	The number of LDAP Modify requests initiated by an application or user.
	Completed Modifies	The number of LDAP Modify requests that were initiated and successful.
	Failed Modifies	The number of LDAP Modify requests that were initiated but failed.
	Average Modify Time	The average time taken by LDAP Modify requests in milliseconds.
ModRDN Options	Initiated Modify DN's	The number of LDAP Modify DN requests initiated by an application or user.
	Completed Modify DN's	The number of LDAP Modify DN requests that were initiated and successful.
	Failed Modify DN's	The number of LDAP Modify DN requests that were initiated but failed.
	Average Modify DN Time	The average time taken by LDAP Modify DN requests in milliseconds.
Search Options	Initiated Searches	The number of LDAP Search requests initiated by an application or user.
	Completed Searches	The number of LDAP Search requests that were initiated and successful.
	Failed Searches	The number of LDAP Search requests that were initiated but failed.
	Average Search Time	The average time taken by LDAP Search requests in milliseconds.
	Total Search Entries Sent	The total number of search entries sent by either LDAP Proxy or LDAP Directory.
	Initiated Base Searches	The number of LDAP Base Search requests initiated by an application or user.
	Completed Base Searches	The number of LDAP Base Search requests that were initiated and successful.
	Failed Base Searches	The number of LDAP Base Search requests that were initiated but failed.
	Average Base Search Time	The average time taken by LDAP Base Search requests in milliseconds.
	Initiated OneLevel Searches	The number of LDAP One Level Search requests initiated by an application or user.

Event Category	Event	Description
	Completed OneLevel Searches	The number of LDAP One Level Search requests that were initiated and successful.
	Failed OneLevel Searches	The number of LDAP One Level Search requests that were initiated but failed.
	Average OneLevel Search Time	The average time taken by LDAP One Level Search requests in milliseconds.
	Initiated Subtree Searches	The number of LDAP Subtree Search requests initiated by an application or user.
	Completed Subtree Searches	The number of LDAP Subtree Search requests that were initiated and successful.
	Failed Subtree Searches	The number of LDAP Subtree Search requests that were initiated but failed.
	Average Subtree Search Time	The average time taken by LDAP Subtree Search requests in milliseconds.
Unknown Options	Unknown Requests	The number of unknown LDAP request initiated by an application or user.
	Unknown Responses	The number of unknown LDAP request initiated by an application or user.

Restoring Monitoring and Trending Configuration

Using the NLPManager utility, you can restore a saved monitoring or trending configuration. This enables you to retrieve a saved monitoring or trending data for analysis in future.

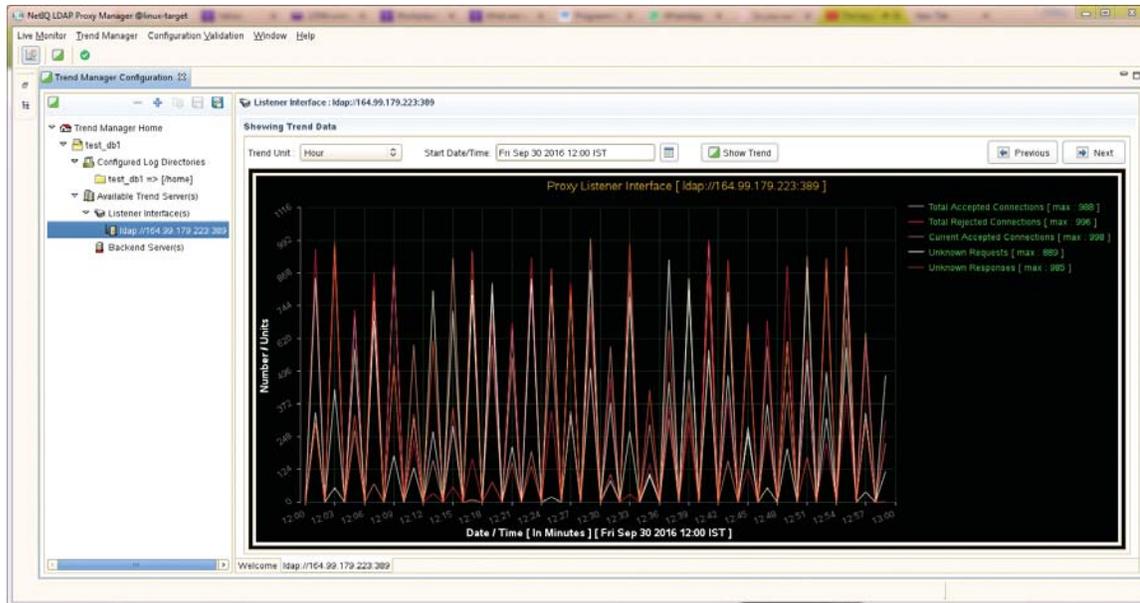
- ◆ [“Restoring Trending Configuration” on page 95](#)
- ◆ [“Restoring Monitoring Configuration” on page 96](#)

Restoring Trending Configuration

To restore a trending configuration, perform the following steps:

- 1 Launch NLPManager and open **Trend Manager Tree**.
- 2 Click the Load Configuration icon  at the top.
- 3 Browse for your saved configuration file (.trend) and click **Finish** to restore the trending configuration.

Figure 4-1 Restoring a Saved Trending Configuration.



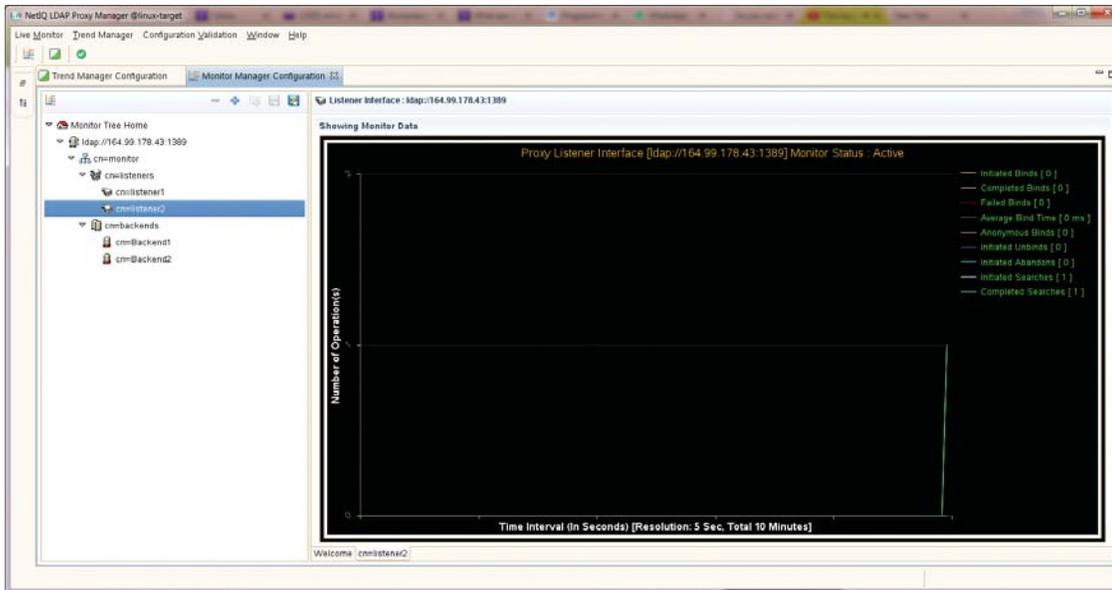
NOTE: If multiple **Trend Manager** tabs are open in the NLPManager utility, while saving, all running trending configurations will be saved in a single file. You can restore all these configuration at one shot using this single configuration file. You can view the list of all trending configurations under the **Trend Manager Tree** panel.

Restoring Monitoring Configuration

To restore a monitoring configuration, perform the following steps:

- 1 Launch NLPManager and open **Proxy Monitor Tree**.
- 2 Click the Load Configuration icon  at the top.
- 3 Browse for your saved configuration file (.mon) and click **Finish** to restore the monitoring configuration.

Figure 4-2 Restoring a Saved Monitoring Configuration.



NOTE: If multiple **Live Monitor** tabs are open in the NLPManager utility, while saving, all running monitoring configurations will be saved in a single file. You can restore all these configuration at one shot using this single configuration file. You can view the list of all the monitoring configurations under the **Proxy Monitor Tree** panel once restored.

5 Maintaining LDAP Proxy

After configuring NetIQ LDAP Proxy, you manage it by running certain commands at the command line.

- ♦ [“Starting LDAP Proxy” on page 99](#)
- ♦ [“Stopping LDAP Proxy” on page 99](#)
- ♦ [“Restarting LDAP Proxy” on page 99](#)
- ♦ [“Checking the Status of LDAP Proxy” on page 99](#)
- ♦ [“Backing Up the LDAP Proxy” on page 100](#)

Starting LDAP Proxy

On SLES 11 and RHEL 6:

- ♦ Start LDAP Proxy by using the `/etc/init.d/nlpd start` command.

On SLES 12 and RHEL 7:

- ♦ Start LDAP Proxy by using the `systemctl start nlpd.service` command.

Stopping LDAP Proxy

On SLES 11 and RHEL 6:

- ♦ Stop LDAP Proxy by using the `/etc/init.d/nlpd stop` command.

On SLES 12 and RHEL 7:

- ♦ Stop LDAP Proxy by using the `systemctl stop nlpd.service` command.

Restarting LDAP Proxy

On SLES 11 and RHEL 6:

- ♦ Restart LDAP Proxy by using the `/etc/init.d/nlpd restart` command.

On SLES 12 and RHEL 7:

- ♦ Restart LDAP Proxy by using the `systemctl restart nlpd.service` command.

Checking the Status of LDAP Proxy

On SLES 11 and RHEL 6:

- ♦ Check the status of LDAP Proxy by using the `/etc/init.d/nlpd status` command.

On SLES 12 and RHEL 7:

- ♦ Check the status of LDAP Proxy by using the `systemctl status nlpd.service` command.

Backing Up the LDAP Proxy

After you configure and start the LDAP Proxy server, you must back up the following directories:

- ♦ The `/etc/opt/novell/ldaproxy/conf` directory, which contains the configuration files, private certificate file, and public certificate files.
- ♦ The `/var/opt/novell/nici/0` directory, which contains the NICI machine key and local password store.
- ♦ The `/etc/opt/novell/nici*.cfg` files, which contain the NICI configuration.

NOTE: You should stop the LDAP Proxy server before backing up the NICI directories. To stop LDAP Proxy, refer to [“Stopping LDAP Proxy” on page 99](#).

You can restore the proxy server by placing the files in the relevant directories.

A

Resolving Error

The following table contains the error codes for LDAP Proxy:

Error Code	Remedy
Proxy Initialization Errors	
0x81510002	Cause: Insufficient memory. Action: Increase the free memory by shutting down services or increase the RAM.
0x8151000D	Cause: The evaluation copy has expired. Action: Register the copy and insert the license key as specified in the documentation. See “Activating LDAP Proxy” in the <i>NetIQ LDAP Proxy 1.5 Installation Guide</i> .
0x8151000E	Cause: The proxy system was not properly initialized. Action: Internal error.
0x8151000F	Cause: The proxy server could not initialize itself because it was running as non-root, or had low system privileges. Action: Take appropriate action.
0x81510100	Cause: The proxy server could not open the log file. Action: Check for the required permissions in the directory specified for logging.
0x81510101	Cause: TLS initialization failed either because of the back-end server or the client. Action: Check for correct TLS/SSL certificates and their permissions and if they are present in the locations specified in the configuration.
0x81510102	Cause: The TLS handshake failed. Action: Check for correct TLS/SSL certificates and their permissions and ensure that they are present in the locations specified in the configuration.
0x81510103	Cause: Verification of the TLS certificate with the back-end server failed. Action: Check for correct TLS/SSL certificates and their permissions and ensure that they are present in the locations specified in the configuration. Also ensure that the back-end server is listening on TLS/SSL with the expected certificates.
0x81510104	Cause: Another instance of proxy is already running. Action: If multiple proxy servers are configured to run on a single machine, ensure that their listening ports/interfaces and log directories are not the same.
Thread Pool Errors	

Error Code	Remedy
0x81530102	<p>Cause: The thread pool ran out of memory.</p> <p>Action: Increase the free memory by shutting down services or increase the RAM.</p>
Session Manager Errors	
0x81550102	<p>Cause: The Session Manager ran out of memory.</p> <p>Action: Increase the free memory by shutting down services or increase the RAM.</p>
Socket Errors	
0x81540102	<p>Cause: Socket logger initialization or SSL/TLS context initialization failed because of insufficient memory.</p> <p>Action: Increase free memory by shutting down services or increase the RAM.</p>
0x81570102	<p>Cause: The socket monitor subsystem ran out of memory.</p> <p>Action: Increase free memory by shutting down services or increase the RAM.</p>
BER Errors	
0x81580002	<p>Cause: Decoding/encoding because of insufficient memory.</p> <p>Action: Increase the free memory by shutting down services or increase the RAM.</p>
0x81580005	<p>Cause: An invalid parameter was found when encoding the BER.</p> <p>Action: Ensure that the LDAP clients are sending proper LDAP requests.</p>
0x81580100	<p>Cause: Decoding error.</p> <p>Action: Ensure that the LDAP clients are sending proper LDAP requests. This error might be transient.</p>
0x81580101	<p>Cause: Encoding error.</p> <p>Action: Internal error.</p>
0x81580103	<p>Cause: Fragment error while decoding.</p> <p>Action: Ensure that the LDAP clients are sending proper LDAP requests.</p>
LDAP Errors	
0x81590001	<p>Cause: This LDAP feature is not implemented.</p>
0x81590002	<p>Cause: The LDAP subsystem ran out of memory.</p> <p>Action: Increase free memory by shutting down services or increase the RAM.</p>
0x81590005	<p>Cause: An invalid parameter was found in the LDAP request/response.</p>
0x81590006	<p>Cause: An invalid LDAP request was received.</p>
0x8159000A	<p>Cause: The specified LDAP object was not found.</p>
0x81590100	<p>Cause: An LDAP protocol error was encountered.</p>

Error Code	Remedy
0x81590101	Cause: An unsupported version of an LDAP request/response was encountered.
0x81590102	Cause: An unsupported LDAP authentication method was encountered.
0x81590103	Cause: The LDAP request received was too big.
0x81590104	Cause: The LDAP response/request received was fragmented.
0x81590107	Cause: No connection route to route this LDAP request was found.
0x81590108	Cause: The LDAP DN received had invalid syntax.
0x81590109	Cause: The LDAP URL received had invalid syntax.
Policy Errors	
0x815B0001	Cause: This policy/action is not implemented. Action: Ensure that the configuration is consistent with the documentation. For more information, see Chapter 2, "Configuring NetIQ LDAP Proxy," on page 17 .
0x815B0002	Cause: The policy subsystem ran out of memory. Action: Increase free memory by shutting down services or increase the RAM.
0x815B0100	Cause: The condition specified in the policy is invalid. Action: Ensure that the configuration is consistent with the documentation. For more information, see Chapter 2, "Configuring NetIQ LDAP Proxy," on page 17 .
0x815B0101	Cause: An unsupported policy was specified in the configuration. Action: Ensure that the configuration is consistent with the documentation. For more information, see Chapter 2, "Configuring NetIQ LDAP Proxy," on page 17 .
0x815B0102	Cause: A request unsupported by an action was received. Action: Internal error.
0x815B0103	Cause: An unsupported action was specified in the policy. Action: Ensure that the configuration is consistent with the documentation. For more information, see Chapter 2, "Configuring NetIQ LDAP Proxy," on page 17 .
0x815B0104	Cause: Unsupported conditional operator was specified. Action: Ensure that the configuration is consistent with the documentation. For more information, see Chapter 2, "Configuring NetIQ LDAP Proxy," on page 17 .
0x815B0105	Cause: An unsupported conditional matching criterion was specified. Action: Ensure that the configuration is consistent with the documentation. For more information, see Chapter 2, "Configuring NetIQ LDAP Proxy," on page 17 .

Error Code	Remedy
0x815B0106	<p>Cause: A request is denied as per policy.</p> <p>Action: Ensure that the connection-route policy for that particular connection/request should be denied.</p>
Back-end Errors	
0x815C0002	<p>Cause: The back-end subsystem ran out of memory.</p> <p>Action: Increase free memory by shutting down services or increase the RAM.</p>
0x815C0100	<p>Cause: A particular back-end server is not available.</p> <p>Action: Ensure that the configured back-end server is running and that the network settings allow the proxy server to communicate with the back-end server.</p>
0x815C0101	<p>Cause: The maximum connection limit for a back-end server was exceeded.</p> <p>Action: The max-connections attribute specified for the particular back-end server has been exceeded. Increase the value for this attribute in the configuration or add more back-end servers.</p>
XML Errors	
0x815E0002	<p>Cause: The XML Logger/XML parser ran out of memory.</p> <p>Action: Increase free memory by shutting down services or increase the RAM.</p>
0x815E0005	<p>Cause: The XML parser encountered an invalid parameter.</p> <p>Action: Check the configuration for XML errors.</p>
0x815E000A	<p>Cause: The XML parser could not find an expected XML object.</p> <p>Action: Check the configuration file to see if it is consistent with the documentation. For more information, see Chapter 2, "Configuring NetIQ LDAP Proxy," on page 17.</p>
0x815E0100	<p>Cause: The XML parser failed to parse.</p> <p>Action: Check the configuration file for XML errors.</p>
0x815E0101	<p>Cause: The XML parser failed to parse the XML DOM.</p> <p>Action: Internal error.</p>
0x815E0102	<p>Cause: Could not create the DOM.</p> <p>Action: Internal error</p>
0x815E0103	<p>Cause: Could not initialize the DOM.</p> <p>Action: Internal error.</p>
0x815E0104	<p>Cause: Could not serialize XML DOM.</p> <p>Action: Internal error.</p>

Error Code	Remedy
0x815E010A	<p>Cause: Multiple nodes were found when only one was expected.</p> <p>Action: Ensure that the configuration file is consistent with the documentation. For more information, see Chapter 2, "Configuring NetIQ LDAP Proxy," on page 17.</p>
Configuration Errors	
0x815F0001	<p>Cause: The configuration directive is not implemented.</p> <p>Action: Check the documentation for the supported policy, conditions, or actions. For more information, see Chapter 2, "Configuring NetIQ LDAP Proxy," on page 17.</p>
0x815F0002	<p>Cause: The configuration subsystem ran out of memory.</p> <p>Action: Increase free memory by shutting down services or increase the RAM.</p>
0x815E010A	<p>Cause: The configuration is incomplete. It does not specify a required configuration object.</p> <p>Action: Check the configuration file for errors.</p>
0x815E010C	<p>Cause: A policy object already exists.</p> <p>Action: Multiple policies must have separate IDs.</p>
0x815F0100	<p>Cause: Could not read the configuration file or schema configuration file.</p> <p>Action: Ensure that the <code>nlp-schemaconf.xml</code> schema configuration file is in the default location of <code>/etc/opt/novell/ldaproxy/conf/</code> and that the permissions are sufficient.</p>
0x815F0101	<p>Cause: The Schema configuration file is invalid.</p> <p>Action: Check the <code>nlp-schemaconf.xml</code> schema configuration file for errors.</p>
0x815F0102	<p>Cause: A data value specified in the configuration file is invalid.</p> <p>Action: Check the configuration file for XML errors or other inconsistencies.</p>
0x815F0103	<p>Cause: An XML node specified in the configuration file is invalid.</p> <p>Action: Check the configuration file for XML errors or other inconsistencies.</p>
0x815F0104	<p>Cause: Two listener nodes were found with the same ID.</p> <p>Action: Multiple listener configuration nodes must have different IDs.</p>
0x815F0105	<p>Cause: The directory path to the log files specified in the configuration is invalid.</p> <p>Action: Check to see if the directory path specified is correct.</p>
0x815F0106	<p>Cause: Unable to create the directory path for logging as specified in the configuration.</p> <p>Action: Ensure that the appropriate permissions are in place for the directory path specified.</p>

Error Code	Remedy
0x815F0107	<p>Cause: The domain name specified in the configuration could not be resolved.</p> <p>Action: Check your DNS configuration or specify an IP address instead of DNS name.</p>
0x815F0108	<p>Cause: Reading the network interface corresponding to the IP/Domain Name given in the configuration failed.</p> <p>Action: Ensure that your network settings for the IP/Domain Name are bound to a network interface as expected.</p>
0xffffe908	<p>Cause: Proxy listener was started with a port which was being used by other services</p> <p>Action: Use a different port to start Proxy listener.</p>
0x815f000a	<p>Cause: The connection route policy in nlpconf.xml referred to a backend group which did not exist.</p> <p>Action: Change the connection route policy to use a backend group which is defined in nlpconf.xml.</p>
Audit/Event Subsystem Errors	
0x815D0001	<p>Cause: An audit event was received but has not been implemented.</p> <p>Action: Check documentation for the supported audit events. For more information, see “Configuring Audit Events” on page 78.</p>
0x815D0002	<p>Cause: The Audit subsystem ran out of memory.</p> <p>Action: Increase free memory by shutting down other services or increase the RAM.</p>
0x815D0005	<p>Cause: The ID of the event object is invalid.</p> <p>Action: Internal error.</p>
0x815D0006	<p>Cause: The audit subsystem received an invalid request.</p> <p>Action: Internal error.</p>
0x815D0008	<p>Cause: The audit subsystem failed to register a producer event because no event data was found.</p> <p>Action: Internal error</p>
0x815D000A	<p>Cause: The specified event object was not found.</p> <p>Action: Internal error.</p>
0x815D000C	<p>Cause: An event object with the same ID is already present.</p> <p>Action:</p>
0x815D0100	<p>Cause: Invalid event data was received.</p> <p>Action: Internal error.</p>

B Sample Configurations

This section lists some sample use case scenarios for deploying NetIQ LDAP Proxy. The sample XML configurations help you to understand the various ways you can configure LDAP Proxy to meet certain requirements. To use the listeners, back-end servers, and network entries provided in these sample configurations, change the configuration to suit your requirements.

The sample XML files are all available in the `/etc/opt/novell/ldapproxy/confsample` directory.

- ♦ [“Sample Entries” on page 107](#)
- ♦ [“Sample XML Files and XML Formatting” on page 108](#)

Sample Entries

The listeners used in the sample configurations are:

- ♦ 192.168.5.2:389 for listener1
- ♦ 192.168.5.2:2389 for listener2

The back-end servers used in the sample configurations are:

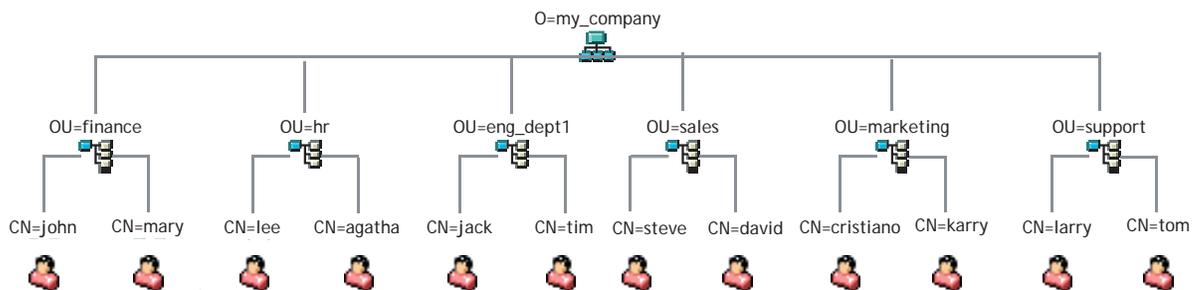
- ♦ 192.168.5.3:389 for backend1
- ♦ 192.168.5.4:2389 for backend2
- ♦ 192.168.5.5:1389 for backend3

The network details used are:

- ♦ 192.168.1.0/24 : ou=finance, ou=hr
- ♦ 192.168.2.0/24 : ou=eng_dept
- ♦ 192.168.3.0/24 : ou=sales
- ♦ 192.168.4.0/24 : ou=marketing
- ♦ 192.168.5.0/24 : ou=support

The Directory Information Tree used in the sample configurations is, as shown in the following figure:

Figure B-1 Sample Directory Information Tree used in the Use Cases



Sample XML Files and XML Formatting

Some sample XML files are available in the `/etc/opt/novell/ldapproxy/conf-sample/` directory.

When you specify the special characters (&, >, <, ;, ", and ,) in your policies to define container names, DN values, and so forth, you must specify the ASCII value for the special character and prefix it with the `\` escape character.

For example, if you want to define a DN, `cn=tes&t,o=novell`, you must specify it as `cn=tes\26t,o=novell`. The relevant XML configuration must be defined as:

```
<if-srch-base op="equal"  
match="case-ignore">cn=tes\26t,o=novell</if-srch-base>
```