NetIQ® LDAP Proxy 1.5

Administration Guide

June 2014

Legal Notice

Contents

About This Guide 7
About NetIQ Corporation 9

1 What's New 11

1.1 Persistent Modify DN Cache . . . 11
1.2 Using Modify DN Cache for Bind Requests . . . 11
1.3 Hash-Based Routing . . . 11
1.4 IPv6 Support . . . 12
1.5 Licence Activation Mechanism . . . 12
1.6 Command Line Options for Activating and Deactivating LDAP Proxy . . . 12
1.7 Support for Active Directory as Back-end . . . 12
1.8 Support for XDAS Standards-Based Auditing . . . 13

2 Overview of NetIQ LDAP Proxy 15

2.1 Introduction to NetIQ LDAP Proxy . . . 15
2.2 Benefits of Using LDAP Proxy . . . 16
2.3 Features of LDAP Proxy . . . 16

3 How LDAP Proxy Works 19

3.1 Architecture . . . 19
3.2 Key Concepts . . . 21
3.2.1 Listener . . . 21
3.2.2 Back-End Server . . . 21
3.2.3 Back-End Server Group . . . 22
3.2.4 Policy . . . 22

4 Manually Configuring NetIQ LDAP Proxy 27

4.1 Understanding the LDAP Proxy Configuration . . . 28
4.2 Sample XML Files and XML Formatting . . . 28
4.3 Supported Directory Servers . . . 28
4.4 Basic Configuration . . . 29
4.5 Configuring Additional Listeners . . . 30
4.5.1 Configuration Parameters . . . 31
4.5.2 Examples . . . 32
4.6 Configuring Additional Back-End Servers . . . 33
4.6.1 Configuration Parameters . . . 34
4.6.2 Examples . . . 36
4.7 Configuring Additional Server Groups . . . 37
4.7.1 Configuration Parameters . . . 38
4.7.2 Example . . . 39
4.8 Configuring Additional Policies . . . 39
4.8.1 Client Network Policy . . . 39
4.8.2 Operation Restriction Policy . . . 43
4.8.3 Map Schema Policy . . . 45
4.8.4 Search Request Policy . . . 47
4.8.5 Connection Route Policy . . . 52

4.8.6 Replace String Policy . . . 57
4.9 Handling Attribute OIDs in Policies . . . 59
4.10 Configuring Proxy Paths . . . 59
4.11 Configuring Audit Events Using XDAS . . . 60
4.11.1 Configuring the XDAS Audit Events . . . 63
4.11.2 Configuring the XDASv2 Property File . . . 64
4.11.3 Enabling XDAS Event Caching . . . 68
4.12 Configuring Audit Events . . . 68
4.13 Configuring the Stat Log . . . 71
4.14 Exporting Certificate Information . . . 71
4.15 Signing the Certificate by 3rd Party CA . . . 72
4.16 Setting the User DN Password . . . 73
4.17 Configuring the Redis Server . . . 73

5 Using the NLPManager to Configure NetIQ LDAP Proxy 75

5.1 Using NLPManager . . . 75
5.1.1 System Requirements . . . 75
5.1.2 Downloading and Starting NLPManager . . . 76
5.2 Basic Configuration . . . 76
5.3 Configuring Additional Listeners . . . 78
5.4 Configuring Additional Back-End Servers . . . 79
5.5 Configuring Additional Server Groups . . . 80
5.6 Creating a New Configuration File . . . 81

6 Managing NetIQ LDAP Proxy 85

6.1 Starting LDAP Proxy . . . 85
6.2 Stopping LDAP Proxy . . . 85
6.3 Restarting LDAP Proxy . . . 85
6.4 Checking the Status of LDAP Proxy . . . 85
6.5 Backing Up the LDAP Proxy . . . 85

7 Configuring Monitoring and Trending Activities 87

7.1 Configuring Monitoring Activities . . . 87
7.2 Managing Trend Analysis . . . 90
7.3 Enabling Monitoring and Trending . . . 93

8 Enabling an LDAP Proxy Trace 97

A Configuring a Linux High Availability Cluster for NetIQ Ldap Proxy 99

A.1 Software Requirements . . . 99
A.2 Hardware Requirements . . . 99
A.3 Installation iSCSI Target . . . 100
A.4 Configuring a NetIQ Ldap Proxy Setup for HA . . . 101
A.4.1 Configuring Node 1 . . . 101
A.4.2 Configuring Node 2 . . . 102
A.4.3 Configuring the Constraints . . . 105

B Sample Configurations 107

B.1 Sample Entries . . . 107
B.2 Using the Proxy as a Directory Firewall . . . 108

B.3 Mapping the Schema Based on the Network and Users 108
B.4 Setting a Search Base for User Identities 108
B.5 Preventing Wild Card Search Filters 108
B.6 Configuring Access Control Based on Users 108

C Error Codes 109

About This Guide

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.

- Chapter 2, “Overview of NetIQ LDAP Proxy,” on page 15
- Chapter 3, “How LDAP Proxy Works,” on page 19
- Chapter 4, “Manually Configuring NetIQ LDAP Proxy,” on page 27
- Chapter 5, “Using the NLPManager to Configure NetIQ LDAP Proxy,” on page 75
- Chapter 6, “Managing NetIQ LDAP Proxy,” on page 85
- Chapter 7, “Configuring Monitoring and Trending Activities,” on page 87
- Chapter 8, “Enabling an LDAP Proxy Trace,” on page 97
- Appendix A, “Configuring a Linux High Availability Cluster for NetIQ Ldap Proxy,” on page 99
- Appendix B, “Sample Configurations,” on page 107
- Appendix C, “Error Codes,” on page 109

Audience

This guide is intended for network administrators.

Feedback

We want to hear your comments and suggestions about this manual and the other documentation included with this product. Please use the User Comments feature at the bottom of each page of the online documentation.

Documentation Updates

For the most recent version of the *NetIQ LDAP Proxy Administration Guide*, visit the NetIQ Documentation Web site (https://www.netiq.com/documentation/).

Additional Documentation

For additional NetIQ LDAP Proxy documentation, refer to the NetIQ LDAP Proxy 1.5 documentation Web site (https://www.netiq.com/documentation/ldapproxy/).

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 What's New

LDAP Proxy 1.5 now supports the following features:

- Section 1.1, “Persistent Modify DN Cache,” on page 11
- Section 1.2, “Using Modify DN Cache for Bind Requests,” on page 11
- Section 1.3, “Hash-Based Routing,” on page 11
- Section 1.4, “IPv6 Support,” on page 12
- Section 1.5, “Licence Activation Mechanism,” on page 12
- Section 1.6, “Command Line Options for Activating and Deactivating LDAP Proxy,” on page 12
- Section 1.7, “Support for Active Directory as Back-end,” on page 12
- Section 1.8, “Support for XDAS Standards-Based Auditing,” on page 13

1.1 Persistent Modify DN Cache

Whenever LDAP Proxy server is restarted, the in-memory ModDNCache is wiped off. This causes the subsequent modify requests to be routed to any of the back-end servers depending on the configuration of the load balancer.

LDAP Proxy now supports storing ModDNCache in a persistent storage to preserve routing information even if the LDAP Proxy server is restarted.

For more information, see “request-route-dit:” on page 53.

1.2 Using Modify DN Cache for Bind Requests

LDAP Proxy is now enhanced to use ModDNCache for routing user bind requests also. For more information, see “moddncache-enable-for-bind:” on page 53.

1.3 Hash-Based Routing

In addition to supporting connection-based load balancing and dynamic load balancing, LDAP Proxy now supports hash-based routing.

For more information, see “Configuring a Hash-Based Route Policy” on page 55.

1.4 IPv6 Support

LDAP Proxy supports both IPv4 and IPv6 networks. You can configure both IPv4 and IPv6 addresses simultaneously. LDAP Proxy can now handle LDAP clients and backend LDAP servers on both IPv4 and IPv6. This capability allows LDAP Clients on IPv6 to work with LDAP servers on IPv4 and vice versa

It supports the following IPv6 modes:

- Dual stack
- Tunneling
- Pure IPv6

LDAP Proxy does not support the following IPv6 address types:

- Link local addresses
- IPv4-mapped IPv6 addresses
- IPv4-compatible IPv6 addresses

LDAP Proxy supports the following addressing formats:

- [::]
- [2015::12]

1.5 Licence Activation Mechanism

In the previous releases, every time the LDAP Proxy server is started, it looks for the license file even if the product is activated using a valid license file.

In this release, after you have activated the product using a valid licence file, the proxy server does not look for the licence key every time it is started.

For more information, see "Activating LDAP Proxy" in the *NetIQ LDAP Proxy 1.5 Installation Guide*.

1.6 Command Line Options for Activating and Deactivating LDAP Proxy

LDAP Proxy now includes the `status`, `activate`, `deactivate` options to check the status, activate and deactivate the product. For more information about these options, see "Activating LDAP Proxy" in the *NetIQ LDAP Proxy 1.5 Installation Guide*.

1.7 Support for Active Directory as Back-end

LDAP Proxy now supports Active Directory (AD) as back-end servers. For more information, see Section 4.3, "Supported Directory Servers," on page 28.

1.8 Support for XDAS Standards-Based Auditing

In addition to the traditional way of auditing, LDAP Proxy now supports XDAS standards-based auditing. For more information, see Section 4.11, “Configuring Audit Events Using XDAS,” on page 60.

2 Overview of NetIQ LDAP Proxy

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.

- Section 2.1, “Introduction to NetIQ LDAP Proxy,” on page 15
- Section 2.2, “Benefits of Using LDAP Proxy,” on page 16
- Section 2.3, “Features of LDAP Proxy,” on page 16

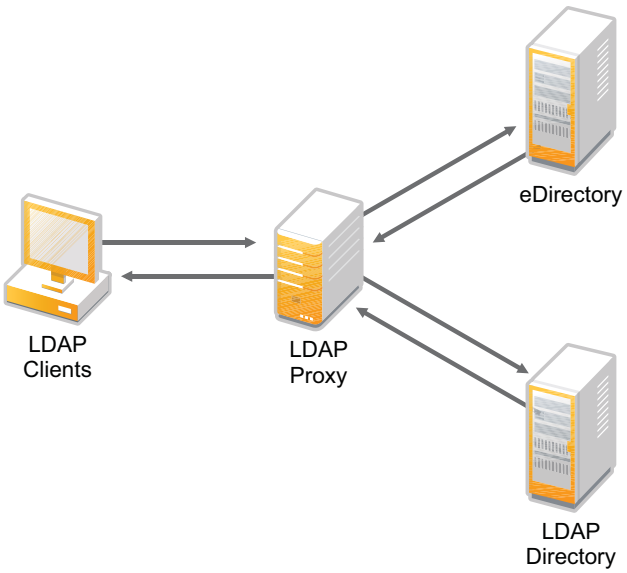
2.1 Introduction to NetIQ LDAP Proxy

NetIQ LDAP Proxy acts as a middleware layer between LDAP clients and LDAP directory servers, and provides support to the LDAP protocol for regulating requests and responses between client applications and directory servers. It provides features such as load balancing, failover, query filtering, data hiding, request denial, centralized auditing and monitoring, and graphical trending of LDAP activities.

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.

Figure 2-1 is a graphical representation of the NetIQ LDAP Proxy environment.

Figure 2-1 *NetIQ LDAP Proxy Environment*



2.2 Benefits of Using LDAP Proxy

The LDAP Proxy server provides significant benefits 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 53.

2.3 Features of LDAP Proxy

LDAP Proxy has many features that help you to efficiently manage directory servers and LDAP traffic:

- **Load Balancing:** LDAP Proxy uses dynamic load balancing algorithms to distribute the load across various servers. The load balancing algorithms use different parameters such as active connections, server response time, and capability. Balance is achieved by grouping at least two back-end servers with the same tree structure into a back-end server group.
- **Failover Mechanism:** LDAP Proxy performs periodic health checks to detect unavailable or slow back-end servers.

  A server is marked unavailable or slow based on any of the following conditions:
  - The connection attempt returns an error
  - The connection has timed out

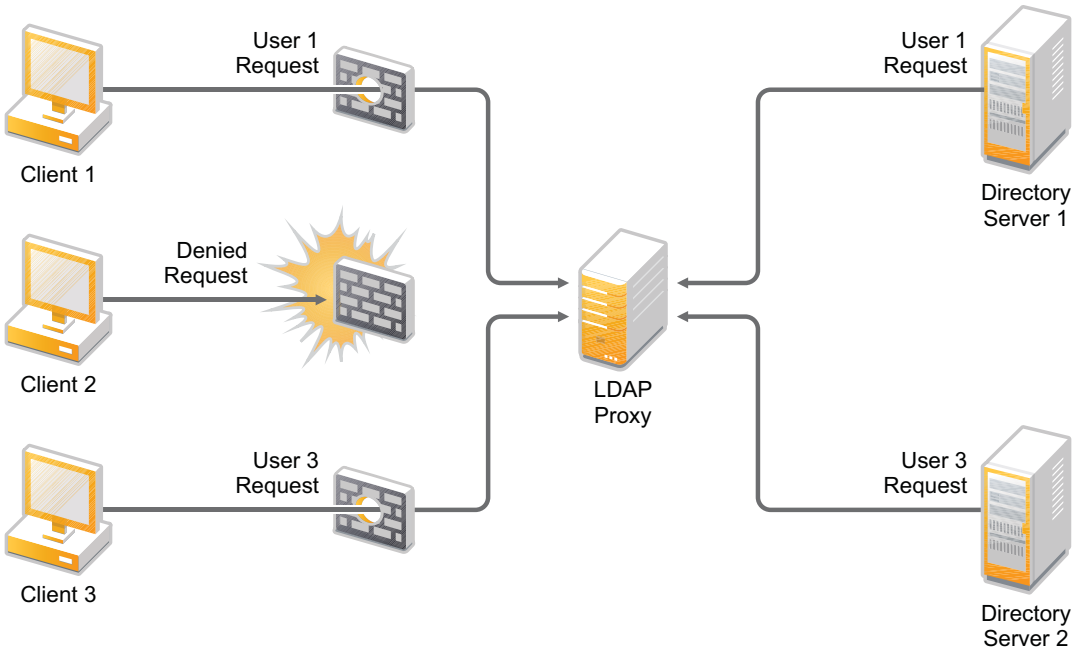
- The directory server is unhealthy
- The proxy receives a connection error on an active connection while sending a request

When a back-end server is unavailable, LDAP Proxy switches active connections to an available back-end server in the server group. Requests that are partially serviced are also routed to a new back-end server, and an LDAP busy result code (51) is sent for the partially serviced requests.

Backing up of LDAP Proxy is achieved by configuring high availability for the LDAP Proxy. For more information, see Appendix A, "Configuring a Linux High Availability Cluster for NetIQ Ldap Proxy," on page 99.

- **Identity-based Policies:** LDAP Proxy provides a simple but powerful set of policies that allows you to implement a greater level of access control over incoming LDAP requests.
  - The Network Restriction policy allows you to configure the proxy server as a firewall. You can use this policy to restrict requests based on clients' network parameters, such as IP address and network address.

Figure 2-2 *LDAP Proxy as a Directory Firewall*



  - The Connection Route policy enables you to route an incoming connection to an appropriate back-end server group. It also determines the identity of an incoming connection and applies required policies before forwarding the processed connection to the associated server group.
  - The Search Restriction policy facilitates re-encoding of incoming search requests. This helps to implement actions such as hiding containers, restricting search attributes, and restricting the search filter (such as CN=*).
  - The Operation Restriction policy allows you to restrict LDAP operations such as Bind, Search, Add, Modify, Delete, Modify DN, and Compare. This restriction helps to achieve read-only and search-only functionality for a server group.
  - The Map Schema policy enables schema compatibility. This helps an application to work with any LDAP directory and allows you to obtain multiple views of the same Directory Information Tree, based on identity.

For more information about each of these policies, refer to Section 3.2, "Key Concepts," on page 21.

- **Live Monitoring:** LDAP Proxy uses an Eclipse-based client tool to provide a graphical view of the activities on the proxy server and back-end directory servers. This helps you to monitor the live LDAP traffic, load, and performance of different LDAP operations.

  For more information about configuring the events to be monitored by using the NetIQ LDAP Proxy GUI, refer to Chapter 7, "Configuring Monitoring and Trending Activities," on page 87.

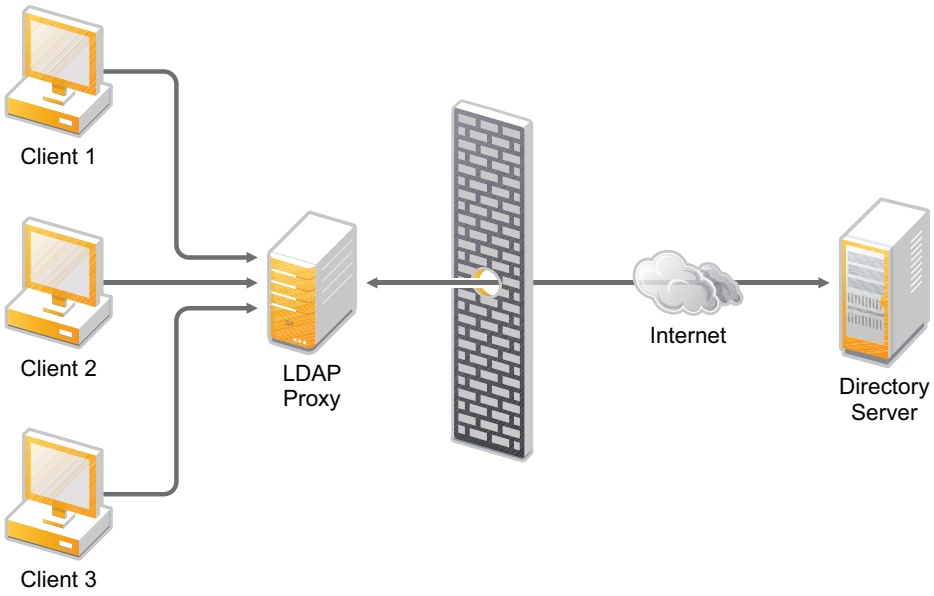
- **Trending:** LDAP Proxy uses an Eclipse-based client tool to analyze and view the trends of LDAP traffic. It also helps you to analyze the load and performance of the proxy server and back-end directory servers. You can analyze this historical trend data for any given time duration for different LDAP operations. The analyzed data is generated and displayed in an informative and customizable graph.

  For more information about configuring the log files for trending, refer to Chapter 7, "Configuring Monitoring and Trending Activities," on page 87.

- **Auditing:** LDAP Proxy allows you to audit the activities on the proxy and back-end directory servers. This helps you to track session details, LDAP policies, and back-end activities. It supports the traditional method of auditing as well as the XDAS-standards based auditing.

- **Forward Proxy:** LDAP Proxy allows you to configure the proxy server as a forward proxy. For instance, there might be a legacy LDAP application that communicates directly to the back-end LDAP server over a clear text channel, compromising security. LDAP Proxy overcomes this limitation by securing the connection between the proxy server and back-end directory servers.

Figure 2-3 *LDAP Proxy as a Forward Proxy*



- **Chaining:** LDAP Proxy provides a chaining feature that can be leveraged by a back-end server or LDAP client that does not support chaining. This feature also ensures the security of back-end server information.

- **Request Routing:** LDAP Proxy provides you with the latest data for any directory server. In a distributed directory environment, all servers might not have the latest copy of the data because of a network failure or synchronization delay. LDAP Proxy overcomes this limitation by tracking data modifications across different servers.

3 How LDAP Proxy Works

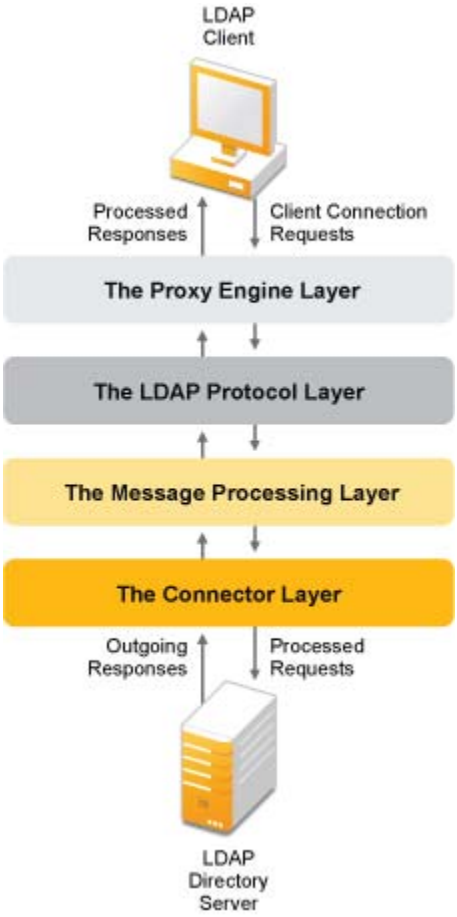
NetIQ LDAP Proxy is designed to analyze the network traffic from various interfaces and regulate requests and responses among LDAP server directories, based on policies.

- Section 3.1, “Architecture,” on page 19
- Section 3.2, “Key Concepts,” on page 21

3.1 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 3-1 *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.

3.2 Key Concepts

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

- Section 3.2.1, “Listener,” on page 21
- Section 3.2.2, “Back-End Server,” on page 21
- Section 3.2.3, “Back-End Server Group,” on page 22
- Section 3.2.4, “Policy,” on page 22

3.2.1 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.

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. For more information about how to configure listeners for LDAP Proxy, refer to Section 4.5, “Configuring Additional Listeners,” on page 30.

3.2.2 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.

To facilitate the load balancing and fault tolerance feature of NetIQ 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 any number of back-end servers for the proxy.

3.2.3 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.

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.

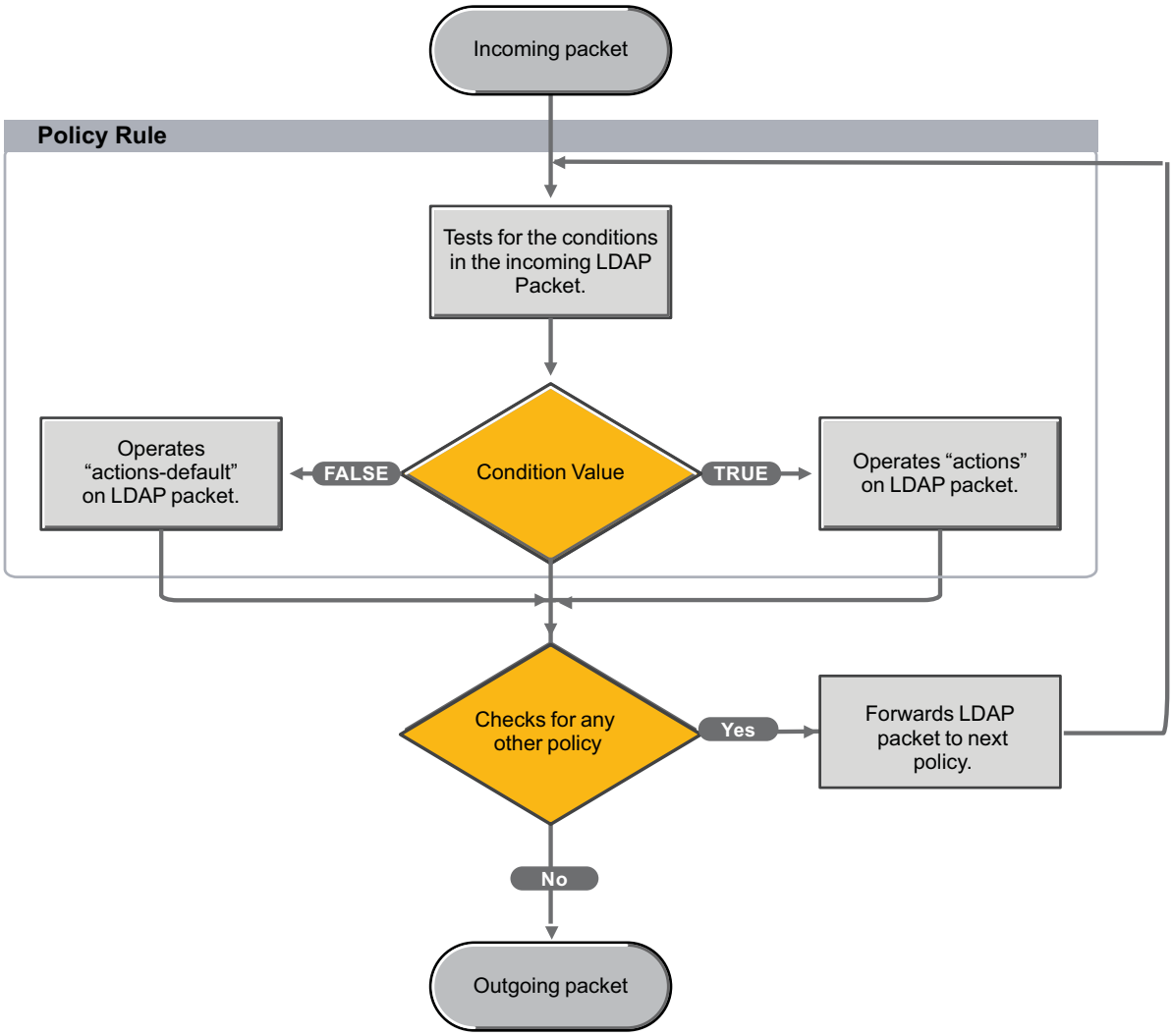
3.2.4 Policy

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 3-2 illustrates how a request is processed by policies.

Figure 3-2 *Applying Policies to Requests and Responses*



Currently, NetIQ LDAP Proxy supports the following policies:

- "Client Network Policy" on page 24
- "Connection Route Policy" on page 24
- "Operation Restriction Policy" on page 24
- "Map Schema Policy" on page 24
- "Search Request Policy" on page 25
- "Replace String Policy" on page 25

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. For example, you can use this policy to configure the proxy to:

- Reject the requests that are coming from a specific IP address in a particular location subnet
- Allow clients that reside in a particular internal network

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.

For example, you can use this policy to configure the proxy to route the requests to `SUN_DIRECTORY_GROUP` if the incoming DN is from `dc=sunone,dc=com`.

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.

When the identity of a request is established, this policy checks the message type of the incoming request. If the message type matches any of the operation types specified in this policy, the policy denies or allows such requests. For instance, you can use this policy to configure the proxy to:

- Deny all Modifies
- Deny a Bind
- Deny object moves

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.

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

- Map UID to CN
- Map a `CommonName` attribute in the application schema to a CN attribute in the back-end directory server 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. For example, you can use this policy to configure the proxy to:

- Hide the ACL, title, and description attributes in the search results
- Use a filter to restrict CN=*/Objectclass=* to a one-level search
- Display only the CN and title attributes in the search results

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.

When the identity of the request is established, the policy checks the DN and the attributes specified in the policy and replaces the string sequences with the string sequences that match the sequences specified in the policy.

For example, you can use this policy to replace `ou=dept, o=company` with `o=company` or `o=subsidary` with `ou=dept, o=company`.

4 Manually Configuring NetIQ LDAP Proxy

Before starting NetIQ LDAP Proxy, you must configure it. The proxy configuration is an XML file where you can define the configuration parameters. To customize NetIQ LDAP Proxy according to your requirements, you can either manually edit this XML file or configure it through the NetIQ LDAP Proxy Manager (NLPManager) graphical utility. This section helps you understand how to manually configure LDAP Proxy. For more information about NLPManager, refer to Chapter 5, "Using the NLPManager to Configure NetIQ LDAP Proxy," on page 75.

- Section 4.1, "Understanding the LDAP Proxy Configuration," on page 28
- Section 4.2, "Sample XML Files and XML Formatting," on page 28
- Section 4.3, "Supported Directory Servers," on page 28
- Section 4.4, "Basic Configuration," on page 29
- Section 4.5, "Configuring Additional Listeners," on page 30
- Section 4.6, "Configuring Additional Back-End Servers," on page 33
- Section 4.7, "Configuring Additional Server Groups," on page 37
- Section 4.8, "Configuring Additional Policies," on page 39
- Section 4.9, "Handling Attribute OIDs in Policies," on page 59
- Section 4.10, "Configuring Proxy Paths," on page 59
- Section 4.11, "Configuring Audit Events Using XDAS," on page 60
- Section 4.12, "Configuring Audit Events," on page 68
- Section 4.13, "Configuring the Stat Log," on page 71
- Section 4.14, "Exporting Certificate Information," on page 71
- Section 4.15, "Signing the Certificate by 3rd Party CA," on page 72
- Section 4.16, "Setting the User DN Password," on page 73
- Section 4.17, "Configuring the Redis Server," on page 73

4.1 Understanding the LDAP Proxy Configuration

When LDAP Proxy is installed, the `nlpconf.xml` configuration file is automatically saved in the `/etc/opt/novell/ldapproxy/conf` directory.

To start LDAP Proxy, you can use either of the following configurations:

- Define the basic configuration required to start LDAP Proxy. This configuration must have at least one listener, a back-end server, and a Connection Route policy. For more information about the basic configuration, refer to Section 4.4, "Basic Configuration," on page 29.
- Customize the LDAP Proxy 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, and so on. Optionally, you can define the proxy paths and monitoring events.

4.2 Sample XML Files and XML Formatting

Some sample XML files are available in the `/etc/opt/novell/ldapproxy/conf-sample/` directory. For more information about the sample use case scenarios for deploying LDAP Proxy, refer to Appendix B, "Sample Configurations," on page 107.

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>
```

4.3 Supported Directory Servers

LDAP Proxy supports eDirectory and Active Directory as back-end servers.

eDirectory

All the supported version of eDirectory can be used as back-end servers.

NOTE: In this release, eDirectory 8.8 SP8 Patch 1 or later versions are supported to co-exist with LDAP Proxy on the same server.

Active Directory

From this release onwards, LDAP Proxy support Active Directory.

Note that communication with AD servers over SSL may fail if the CRL information from the Certificate Authority (CA) is not accessible anonymously. You can make the CRL information accessible, by installing the IIS Web server and then getting the CRLs published from the CA. The CA can then be configured to mint certificates to the AD servers with this URL. If there are any LDAP URLs present in the Cisco Discovery Protocol (CDP), you must remove it because it cannot be accessed anonymously.

NOTE: Cross-domain operations are not supported.

4.4 Basic Configuration

The basic configuration define the following:

- **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, but you can customize it to listen only on specific interfaces.
- **Back-end server group:** 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. For more information about configuring additional servers, refer to Section 4.6, "Configuring Additional Back-End Servers," on page 33.
- **Connection route policy:** A minimum of one Connection Route policy that specifies where the connections are to be routed.

  The `<list-policy>` node in the `nlpconf.xml` file contains a simple Connection Route policy that defines where the 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.

To define the basic configuration for LDAP Proxy:

1. Open the `nlpconf.xml` file from the `/etc/opt/novell/ldapproxy/conf` directory in a text editor.
2. (Optional) If you want to configure LDAP Proxy to listen on a specific interface, provide the appropriate values in the `<addr-ipv4>` or `<addr-ipv6>` attribute, and the `<port>` attribute. Note that you must specify the IPv6 address within square braces, as shown in the following example.

```
<list-listener>
         <listener id-listener="listener1">
                <service protocol="ldap">
                       <addr-ipv6>[2015::37]</addr-ipv6>
                       <port>4489</port>
                </service>
         <ref-policy-client-network>allow_my_company_network</ref-policy-
client-network>
         <ref-policy-connection-route>default_user_route</ref-policy-
connection-route>
         </listener>
         <listener id-listener="listener2">
                <service protocol="ldap">
                       <addr-ipv4>192.168.1.2</addr-ipv4>
                       <port>3389</port>
                </service>
         <ref-policy-client-network>allow_my_company_network</ref-policy-
client-network>
         <ref-policy-connection-route>default_user_route</ref-policy-
connection-route>
         </listener>
<list-listener>
```

3. Configure back-end servers:

```
<list-backend-server health-check-interval-secs="60">
       <backend-server id-backend-server="Backend1">
              <service protocol="ldap">
                     <addr-ipv6>[2015::37]</addr-ipv6>
                     <port>389</port>
              </service>
       </backend-server>
       <backend-server id-backend-server="Backend2">
              <service protocol="ldap">
                     <addr-ipv4>0.0.0.0</addr-ipv4>
                     <port>1389</port>
              </service>
       </backend-server>
       <backend-server id-backend-server="Backend3">
              <service protocol="ldap">
                     <addr-ipv4>1.2.3.4</addr-ipv4>
                     <port>389</port>
              </service>
       </backend-server>
</list-backend-server>
```

The `<backend-server id-backend-server="Backend1">` node specifies the configuration for the first back-end server (the server ID is Backend1). The `<backendserver id-backend-server="Backend2">` node specifies the configuration for the second back-end server (the server ID is Backend2).

3a In the `<backend-server id-backend-server="Backend1">`, `<backendserver id-backend-server="Backend2">`, and `<backendserver id-backend-server="Backend3">` nodes, change the `<addr-ipv4>` or the `<addr-ipv6>` attribute value to the IP address of the system where the LDAP server that you want to use as one of the back-end servers is running. Change the `<port>` to the port number through which you want the LDAP server to receive requests from the LDAP Proxy.

4 Save the `nlpconf.xml` file in the `/etc/opt/novell/ldapproxy/conf` directory.

You can now start NetIQ LDAP Proxy. For more information, refer to Chapter 6, "Managing NetIQ LDAP Proxy," on page 85.

NOTE: Some sample XML configuration files are available in the `/etc/opt/novell/ldapproxy/conf-sample` directory. You can use these samples to test your configuration setup.

4.5 Configuring Additional Listeners

You can configure additional listeners for the proxy configuration. 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, assume that 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>
```

To add 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" on page 31.
4. To add more listeners, repeat Step 2 to Step 3.
5. Save the `nlpconf.xml` file.

For additional information about configuring listeners, see the following sections:

- Section 4.5.1, "Configuration Parameters," on page 31
- Section 4.5.2, "Examples," on page 32

4.5.1 Configuration Parameters

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 attribute:

- **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`.

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`.

- **<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 Section 4.8.1, "Client Network Policy," on page 39.
- **<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 Section 4.8.5, "Connection Route Policy," on page 52.

  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/ldapproxy/conf/ssl/private` directory. The certificate should be in the `pem` format.

For more information about how to export certificate file information, refer to Section 4.14, "Exporting Certificate Information," on page 71.

4.5.2 Examples

- "Example 1" on page 32
- "Example 2" on page 33

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`.

4.6 Configuring Additional Back-End Servers

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, assume that 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-ipv6>[2015::37]</addr-ipv6>
   <port>389</port>
  </service>
 </backend-server>
</list-backend-server>
```

To add a back-end server:

1. Open the `nlpconf.xml` file from the `/etc/opt/novell/ldapproxy/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" on page 34.
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.

For additional information about configuring back-end servers, see the following sections:

- Section 4.6.1, "Configuration Parameters," on page 34
- Section 4.6.2, "Examples," on page 36

4.6.1 Configuration Parameters

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 32, 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/ldapproxy/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.
- **<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.

    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.

4.6.2 Examples

- "Example 1" on page 36
- "Example 2" on page 37

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`.

4.7 Configuring Additional Server Groups

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, assume that 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>
```

To add a back-end server group:

1. Open the `nlpconf.xml` file from the `/etc/opt/novell/ldapproxy/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" on page 38.

5 To add more back-end server groups, repeat Step 2 to Step 4.

6 Save the `nlpconf.xml` file.

For additional information about configuring back-end server groups, see the following sections:

- Section 4.7.1, "Configuration Parameters," on page 38
- Section 4.7.2, "Example," on page 39

4.7.1 Configuration Parameters

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.

4.7.2 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 `dynld`. `Connld` indicates that a connection-based load balancing is performed between `Backend1` and `Backend3`. `Dynld` indicates that dynamic load balancing is performed between `Backend2` and `Backend4`.

4.8 Configuring Additional Policies

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.

You can configure the following policies for LDAP Proxy:

- Section 4.8.1, “Client Network Policy,” on page 39
- Section 4.8.2, “Operation Restriction Policy,” on page 43
- Section 4.8.3, “Map Schema Policy,” on page 45
- Section 4.8.4, “Search Request Policy,” on page 47
- Section 4.8.5, “Connection Route Policy,” on page 52
- Section 4.8.6, “Replace String Policy,” on page 57

4.8.1 Client Network Policy

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

For instance, assume that 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>192.168.5.0</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>
   </conditions>
   <actions>
     <do-allow/>
   </actions>
   <actions-default>
    <do-deny/>
   </actions-default>
  </rule>
</policy-client-network>
<list-policy>
```

To add a Client Network policy:

1. Open the `nlpconf.xml` file from the `/etc/opt/novell/ldapproxy/conf` directory in any XML editor.
2. To add a Client Network 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.

   For more information about the elements and attributes that are used to define a Client Network policy, refer to "Configuring a Client Network Policy" on page 40.
4. To add more Client Network policies, repeat Step 2 to Step 3.
5. Save the `nlpconf.xml` file.

Configuring a Client Network Policy

The following elements and parameters are used to configure Client Network 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.

The `<policy-client-network>` element 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 Client Network policy that is configured. Every policy has a rule.

  This element can have the following child elements:

  - **<conditions>:** The condition to be evaluated.

This element can have the following child elements:

- **<and>:** The AND logical operator.
- **<or>:** The OR logical operator.
- **<not>:** The NOT logical operator.
- **<if-network-addr>:** A network address to be evaluated. This element can have the following child elements specified by using the equal or not-equal values:
  - **<network-addr>:** The network address.
  - **<subnet-mask>:** The subnet mask.
  - **<subnet-bits>:** The subnet bits. This element must have a value in the range 0-32.

  NOTE: If the `<network-addr>` element is defined, you must also define either the `<subnet-mask>` or the `<subnet-bits>` element.
- **<if-ip-addr>:** A network IP address specified by using the equal or not-equal values.
- **<if-port>:** A network port number in the range 1-65536 specified by using the equal, not-equal, less-or-equal, or greater-or-equal values.

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

  In the sample configuration, the action to be performed is specified as `<do-allow />`, which means that all incoming requests satisfying the condition are allowed to establish a connection.
- **<actions-default>:** The default action to be performed if the condition evaluates to false.

  In the sample configuration, the action to be performed is specified as `<do-deny />`, which means that all incoming requests that do not satisfy the specified condition cannot establish a connection.

Depending on how you want to configure the policy, the `<actions>` and `<actions-default>` elements can have the following child elements:

- **<do-allow />:** Accepts the connection.
- **<do-deny />:** Denies the connection.

Examples

- "Example 1" on page 41
- "Example 2" on page 42
- "Example 3" on page 42

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`

4.8.2 Operation Restriction Policy

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

For instance, assume that 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>
```

To add an Operation Restriction policy:

1. Open the `nlpconf.xml` file from the `/etc/opt/novell/ldapproxy/conf` directory in any XML editor.
2. To add an Operation Restriction policy, create an instance similar to the sample configuration within the `<list-policy>` node. You must define the policy after the Client Network 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.

   For more information about the elements and attributes that are used to define an Operation Restriction policy, refer to "Configuring an Operation Restriction Policy" on page 44.

   You must provide the `ldap-bind-request` with any other message type you provide. For example, if want to specify the message type `ldap-search-request`, it must be combined with the `ldap-bind-request`, as shown in the "Example" on page 45.
4. To add more Operation Restriction policies, repeat Step 2 to Step 3.
5. Save the `nlpconf.xml` file.

Configuring an Operation Restriction Policy

The following elements and attributes are used to configure Operation Restriction policies:

<policy-operation-restriction>: Specifies that the element configured is an Operation Restriction 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. It can be any alphanumeric value and must be a unique value. This is a mandatory attribute.

The `<policy-operation-restriction>` element 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 Operation Restriction policy that is configured. Every policy has a rule.

  This element has the following child elements:

  - **<conditions>:** The condition to be evaluated. This is an optional element defined by using the `<or>`, `<not>`, and `<if-message-type>` elements.

    The `<if-message-type>` element is defined by using the equal or not-equal attributes, and the following operation types:

    - `<ldap-bind-request>`
    - `<ldap-search-request>`
    - `<ldap-modify-request>`
    - `<ldap-add-request>`
    - `<ldap-delete-request>`
    - `<ldap-moddn-request>`
    - `<ldap-compare-request>`
    - `<ldap-extended-request>`
  - **<actions>:** The action to be performed if the condition evaluates to true.

In the "Example" on page 45, the action to be performed is specified as `<do-deny/>`, which means that all incoming requests that do not satisfy the specified condition are restricted from performing any action on the directory structure.

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

  In the "Example" on page 45, the action to be performed is specified as `<do-allow/>`, which means that all incoming requests satisfying the condition are allowed to perform an action on the directory structure.

Depending on how you want to configure the policy, the `<actions>` and `<actions-default>` elements can have the following child elements:

- **<do-allow />:** Allows the client operations to proceed.
- **<do-deny />:** Denies the client requests.

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.

4.8.3 Map Schema Policy

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

For instance, assume that 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 atrribute-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>
```

To add a Map Schema policy:

1. Open the `nlpconf.xml` file from the `/etc/opt/novell/ldapproxy/conf` directory in any XML editor.
2. To add a Map Schema policy, create an instance similar to the sample configuration within the `<list-policy>` node. You must define the policy after the Operation Restriction policy in the node.
3. Define a name to identify the policy, an action, and a default action for the policy.

   For more information about the elements and attributes that are used to define a Map Schema policy, refer to "Configuring a Map Schema Policy" on page 46.
4. Save the `nlpconf.xml` file.

Configuring a Map Schema Policy

The following elements and attributes are used to configure Map Schema policies:

<policy-map-schema>: Specifies that the element configured is a Map Schema 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.

  The `<policy-map-schema>` element 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 Map Schema policy that is configured. Every policy has a rule. This element can have the following child elements:
  - **<actions>:** The action to be performed.

    In the sample configuration, the action to be performed is specified as `<do-map-schema>`, which means that all incoming requests and outgoing responses satisfying the condition must be allowed to perform schema mapping on the directory structure. It specifies the following mapping:
    - The `CommonName` attribute maps to the `cn` attribute
    - The `country` attribute maps to the `c` attribute
    - The `organization` attribute maps to the `o` attribute

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

  In the sample configuration, the default action is specified as `<do-nothing />`, which means that all incoming requests and outgoing responses are not allowed to perform schema mapping.

Depending on how you want to configure the policy, the `<actions>` and `<actions-default>` elements can have the following child elements:

- **<do-map-schema>:** Map the schema based on attributes.
- **<do-nothing>:** Do not perform any action.

4.8.4 Search Request Policy

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

For instance, assume you want to define a Search Request policy, 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 evaluates to 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>
```

To add a Search Request policy:

1. Open the `nlpconf.xml` file from the `/etc/opt/novell/ldapproxy/conf` directory in any XML text editor.
2. To add a Search Request policy, create an instance similar to the sample configuration within the `<list-policy>` node.
3. Define a name to identify the policy, a set of conditions, an action, and a default action for the policy.

   For more information about the elements and attributes that are used to define a Search Request policy, refer to "Configuring a Search Request Policy" on page 48.
4. To add more Search Request policies, repeat Step 2 to Step 3.
5. Save the `nlpconf.xml` file.

Configuring a Search Request Policy

The following elements and attributes are used to configure Search Request policies:

<policy-search-request>: Specifies that the element configured is a Search Request 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.

The `<policy-search-request>` element 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 Search Request policy that is configured. The element can have the following child elements:
  - **<conditions>:** The search-related conditions to be evaluated. This element can have the following child elements:
    - **<or>:** The OR logical operator.
    - **<and>:** The AND logical operator.
    - **<not>:** The NOT logical operator.
    - **<if-srch-selection-attr>:** This type is specified by using the op attribute and defined either as equal or not-equal to some string values. The match value is defined either as `case-exact` or `case-ignore`.
    - **<if-srch-base>:** Specifies to evaluate for the specified base container. This type is specified by using the `op` attribute and defined either as equal or not-equal to some string values. The match value is defined either as case-exact or case-ignore.
    - **<if-srch-filter>:** The search filter to be evaluated for. This element has the following attributes:
      - **<filter-attribute>:** The attributes to filter. The match value is defined as `case-ignore`.
      - **<filter-type>:** The filter type to use for filtering. This element is defined by using the `op` attribute and values such as substrings, greater-or-equal, less-or-equal, present, or approx-match.
    - **<if-srch-scope>:** The scope of the search operation to be evaluated. The three types of search scope are `base-object`, `one-level`, and `sub-tree`. The match is controlled by using the `op` attribute, as specified in the sample configuration.

  Depending on how you want to configure the policy, the `<actions>` and `<actions-default>` elements can have the following child elements:

  - **<do-deny>:** Denies the search request.
  - **<do-allow>:** Performs the search request.
  - **<do-modify-search>:** Modifies a search request. This element uses the following child elements:
    - **<description>:** An explanation about the modify search request.
    - **<base>:** The modifications to be done on the search base. The modification is specified by using the `op` attribute. The two types of actions that can be specified are `replace` and `append`.
    - **<scope>:** The scope of the modify request. The three types of search scope are `base-object`, `one-level`, and `sub-tree`. If this element is not defined, the scope of the search is used to progress with the modify action.

- **<time-limit>:** The time limit within which the modify request must be performed.
- **<size-limit>:** The size limit of the number of values to be returned after evaluating the modify request.
- **<selection-attributes>:** The attributes that must be returned after evaluating the modify search request.

- **<do-restrict-view>:** Restricts some part/information of the directory tree. This element is specified by using the `op` attribute and the `show-only` value. It can use the following child elements:
  - **<attributes>:** Shows the attributes of the directory tree.
  - **<objectclasses>:** Shows the objectclass of the directory tree.
  - **<containers>:** Shows the containers of the directory tree.
- **<do-send-monitor-response>:** Sends a response for the monitor request and enables live monitoring. "Example 2" on page 50 describes the configuration to enable live monitoring.

- **<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.

Examples

- "Example 1" on page 49
- "Example 2" on page 50
- "Example 3" on page 50

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/ldapproxy/conf</dir-config>
    <dir-log>/var/opt/novell/ldapproxy/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-grp1</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-grp1</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>`.

4.8.5 Connection Route Policy

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, assume that 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>connld</ref-load-balance>
    </do-use-route>
   </actions>
   <actions-default>
    <do-nothing/>
   </actions-default>
  </rule>
 </policy-connection-route>
</list-policy>
```

To add a Connection Route policy:

1. Open the `nlpconf.xml` file from the `/etc/opt/novell/ldapproxy/conf` directory in any XML editor.
2. To add a Connection Route policy, create an instance similar to the sample configuration within the `<list-policy>` node.
3. Define a name to identify the policy, a set of conditions, an action, and a default action for the policy.

   For more information about the elements and attributes that are used to define a Connection Route policy, refer to "Configuring a Connection Route Policy" on page 53.
4. To add more back-end servers, repeat Step 2 to Step 4.
5. Save the `nlpconf.xml` file.

Configuring a Connection Route Policy

The following elements and attributes are used 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.
- **moddncache-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.
- **moddncache-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 moddncache-expiry-time attribute:

```
<policy-connection-route id-policy="conn-route-policy" request-route-
dit="ldap_proxy_tree" moddncache-enabled-for-bind="true" moddncache-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 moddncache-enable-for-bind attributes are enabled. An expiry time of 1800 seconds for the cache is also defined.

The `<policy-connection-route>` element 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 Connection Route policy that is configured. The element can have the following child elements:
  - **<conditions>:** The client related identity to be evaluated. In the sample configuration, the condition specifies to allows any incoming client requests either from a network having IP address equal to 192.168.1.1 having `24` subnet bits or from a base `ou=dept1,o=novell`. This element can have the following child elements:
    - **<and>:** The AND logical operator.
    - **<or>:** The OR logical operator.
    - **<not>:** The NOT logical operator.
    - **<if-network-addr>:** A network address to be evaluated. This element can have the following child elements specified by using the `equal` or `not-equal` attribute:
      - **<network-addr>:** The network address.
      - **<subnet-mask>:** The subnet mask.
      - **<subnet-bits>:** The subnet bits. This element must have a value in the range 0-32.

      NOTE: If the `<network-addr>` element is defined, you must also define either the `<subnet-mask>` or `<subnet-bits>` element.

    - **<if-ip-addr>:** A network IP address. It is specified by using the equal or not-equal attribute.
    - **<if-port>:** A network port number in the range 1-65536. It is specified by using the `equal`, `not-equal`, `less-or-equal`, or `greater-or-equal` attribute.
    - **<if-bind-dn-container>:** The container value, which is specified by using the `equal` or `not-equal` attribute.
    - **<if-bind-dn>:** The container type, which is specified by using the equal or not-equal attribute.
  - **<actions>:** The action to be performed if the condition evaluates to true.

    This element can have the following child elements:
    - **<do-deny>:** Denies the action.
    - **<do-use-route>:** The route for all incoming requests. This element can have the following child elements:
      - **<ref-policy>:** The term `ref` in this element indicates that this element is a pointer to a policy of type policy-client-network policy.
      - **<ref-load-balancer>:** This element is used to route the incoming request to a back-end server group. The term `ref` in this element indicates that this element is actually a pointer to a defined back-end server group.

        In the above sample configuration, this `<ref-load-balancer>` element is defined within the `<actions>` element. It directs the incoming request to back-end server group `connld`. This is a mandatory attribute.

        When you define the `<do-use-route>` element, you must ensure that:

Any policy that is referred in the `<do-use-route>` element must be defined above this location where it is referred.

More than one Map Schema policy cannot be referred within one Connection Route policy.

<ref-hash-balancers> : This element is used to configure a hash-based route policy with a group of load balancers, as shown in the sample configuration in "Configuring a Hash-Based Route Policy" on page 55.

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.

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

  This element is specified by using the `<do-nothing/>` element.

- **<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 Section 4.17, "Configuring the Redis Server," on page 73.

  - **redis-auth-username:** This is an optional attribute and 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

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-grp
                          </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.

4.8.6 Replace String Policy

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 seeAlso 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>
      <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>
```

To add a Replace String policy:

1. Open the `nlpconf.xml` file from the `/etc/opt/novell/ldapproxy/conf` directory in any XML editor.
2. To add a Replace String policy, create an instance similar to the sample configuration within the `<list-policy>` node.
3. Define a name to identify the policy, a set of conditions, an action, and a default action for the policy.

   For more information about the elements and attributes that are used to define a Replace String policy, refer to "Configuring a Replace String Policy" on page 58.
4. To add more Replace String policies, repeat Step 2 to Step 3.
5. Save the `nlpconf.xml` file.

Configuring a Replace String Policy

The following elements and attributes are used to configure Replace String policies:

<policy-replace-string>: Specifies that the element configured is a Replace String policy. This element can have the following child elements:

- **<rule>:** The rule associated with the Replace String policy that is configured. Every policy has a rule. This element can have the following child elements:
  - **<conditions>:** The condition to be evaluated. This is an optional element defined by using the `<or>`, `<not>`, and `<if-message-type>` elements.

    The `<if-message-type>` element is defined by using the equal or not-equal attributes, and the following operation types:
    - `<ldap-bind-request>`
    - `<ldap-search-request>`
    - `<ldap-modify-request>`
    - `<ldap-add-request>`
    - `<ldap-delete-request>`
    - `<ldap-moddn-request>`
    - `<ldap-compare-request>`
    - `<ldap-extended-request>`
  - **<actions>:** The action to be performed if the condition evaluates to true.

    In the sample configuration, the action to be performed is specified as `<do-replace-string>`, which means that all incoming requests and outgoing responses satisfying the condition are allowed to replace strings on the directory structure. It specifies the following replacement:
    - The `o=company` string to `ou=marketing,o=company` in DN and cn attributes.
    - The `o=subsidy` string to `o=company` in DN and cn attributes.
  - **<actions-default>:** The default action to be performed if the condition evaluates to false.

    In the sample configuration, the default action is specified as `<do-nothing />`, which means that all incoming requests and outgoing responses that do not satisfy the specified condition are not allowed to perform string replace.

Depending on how you want to configure the policy, the <actions> and <actions-default> elements can have the following child elements:

- **<do-replace-string>:** Replace the string based on attributes.
- **<do-nothing>:** Does not perform any action.

- **<order>:** Replaces the strings in the order specified in the conf file.

  Within the <order>, when the first replace string request is successful, it stops replacing.

- **<replace>:** Each replace tag has a from, to mapping that is used in the replace requests. For Search responses, the strings are replaced in reverse order.

  For example, for a bind request with o=subsidy changed to o=company, the search response changes o=company to o=subsidy.

- **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.

  The <policy-replace-string> element can have the following child elements:

- **<description>:** An explanation about the policy. This is an optional element used for reference purposes.

4.9 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 nlpschemaconf.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 nlpschemaconf.xml file:

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

4.10 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/ldapproxy/conf`.
- **<dir-log>:** The location of the `log` file. In the sample configuration, the location specified is `/var/opt/novell/ldapproxy/log`.

4.11 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 4-1 *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_CONFIGURATION	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

4.11.1 Configuring the XDAS Audit Events

To configure XDAS audit events:

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-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 4-5.
4. Save the `nlpconf.xml` file.

4.11.2 Configuring the XDASv2 Property File

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

# 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/ldapproxy/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 4-2 *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 4-3 *XDAS Settings*

Setting	Description
log4j.rootLogger=*debug, S, R*	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.
log4j.appender.S=*org.apache.log4j.net.SyslogAppender*	Specifies the appender S to be a Syslog appender.
log4j.appender.S.Host=*localhost*	Specifies the location of the Syslog server where XDAS events are logged. For example, log4j.appender.S.Host=192.168.0.1
log4j.appender.S.Port=*port*	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.
log4j.appender.S.Protocol=UDP	Specifies the protocol to use. For example, UDP, TCP, or SSL.
log4j.appender.S.SSLCertFile=*/etc/opt/novell/mycert.pem*	Specifies the SSL certificate file for the SSL connection. Use double backslashes to specify the path of the file. This is an optional setting.
log4j.appender.S.Threshold=INFO	Specifies the minimum log level allowed in the Syslog appender. Currently, the INFO log level is supported.
log4j.appender.S.Facility=USER	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
log4j.appender.S.layout=*org.apache.log4j.PatternLayout*	Layout setting for Syslog appender.
log4j.appender.S.layout.ConversionPattern=*%c : %p%m%n*	Layout setting for Syslog appender. For information about the conversion patters and their descriptions, see logging.apache.org.
log4j.appender.R=*org.apache.log4j.RollingFileAppender*	Specifies appender R to be a Rolling File appender
log4j.appender.R.File=*/var/opt/novell/ldapproxy/log/xdas-events.log*	The location of the log file for a Rolling File appender.
log4j.appender.R.MaxFileSize=*100MB*	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.
log4j.appender.R.MaxBackupIndex=*10*	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.
log4j.appender.R.layout=*org.apache.log4j.PatternLayout*	Layout setting for Rolling File appender.
log4j.appender.R.layout.ConversionPattern=*%d{MMM dd HH:mm:ss} %c : %p%m%n*	Layout setting for Rolling File appender. See Table 4-4 on page 66 for simple date format patterns. For information about the conversion patters 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 4-4 *Date and Time Pattern Example*

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

4.11.3 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/ldapproxy/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/ldapproxy`. 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.

4.12 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/ldapproxy/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 4-5.
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 4-5 *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.
	1442833	The LDAP Compare responses that are sent.

Category	Event-id	Description
	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.

4.13 Configuring the Stat Log

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>
 <log-filesize-limit>102400</log-filesize-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.

4.14 Exporting Certificate Information

If you specify the protocol as LDAPS when you configure listeners, you must provide the private key certificate file information. You can export the certificate file information from one of the back-end servers by using the `nlpexportcert` utility, which is bundled with the proxy package. This utility is used to export the private key certificate only from NetIQ eDirectory. Currently, the proxy server supports only the PEM certificate file format.

During proxy installation, this utility is installed by default in the `/opt/novell/ldapproxy/bin` directory.

1. Before running the tool, export the library path by using the following command:

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

2. Run the tool by using the following command:

   ```
   nlpexportcert -h <hostName> -p <port> -s <serverDn> -a <userName> -w
   <userPassword> -C <certificateName> -W <certificatePassword> -o <outputFile> -
   F <ouput file format>
   ```

Option	Description
-h <hostName>	The hostname or IP address of the directory server.
-p <port>	(Optional) The NCP port of the directory server. By default, the value is 524.
-s <serverDn>	The directory server DN context.
-a <userName>	The username to log in to the directory server.
-w <userPassword>	The password of the specified user.
-C <certificateName>	The filename of the certificate residing on the server.
-W <certificatePassword>	The password used to encrypt the certificate information while exporting. You can provide any password of your choice.
-o <outputFile>	The name of the file to which the information is to be exported. You must provide this filename while configuring LDAP Proxy to act as a secure interface.
-F <ouput file format>	The certificate file format.

For example, `nlpexportcert -h 192.168.1.1 -p 524 -s server.company -a admin.company -w adminpassword -C "SSL CertificateIP" -W certpassword -o private-cert.pem -F pem`

IMPORTANT: The certificate file you have exported to a machine cannot be used to configure the proxy on another machine because the private key is encrypted by the machine's NICI key.

4.15 Signing the Certificate by 3rd Party CA

To get the server certificate signed by another Certificate Authority (CA), you must perform the following tasks:

1. Create a new server certificate request (CSR) using iManager:
   - **1a** In NetIQ iManager, click the *Roles and Tasks* button.
   - **1b** Click *NetIQ Certificate Access* > *Server Certificates* > *New*.
   - **1c** Specify the nickname, select *Custom*, and click *Next*.
   - **1d** Select *External certificate authority* and click Next.
   - **1e** Retain the default key size, the *Allow private key to be exported* option selected, and click *Next*.
   - **1f** Enter the proxy server name in the *Subject Name* field (for example: o=novell.cn=proxy servername), click *Next* and then click *Finish*.

     NOTE: You may also add the IP address of the proxy server in the *Subject Alternative Name* field.

   A server certificate is created using the parameters you specified.
2. Export the certificate in PEM format:
   - **2a** In NetIQ iManager, click the *Roles and Tasks* button.
   - **2b** Click *NetIQ Certificate Server* > *Configure Certificate Authority*.

2c In the Certificates tab, select the certificate that you created and click *Export*.

2d Enter the password to protect the private key and click *next*.

The certificate is exported.

3 Get the certificate that you exported, signed by a CA.

4 Import the signed certificate along and the CA certificate:

4a In NetIQ iManager, click the *Roles and Tasks* button.

4b Click *NetIQ Certificate Access* > *Server Certificate*s.

4c Click the *Import* option adjacent to the certificate you created.

4d Import the signed certificate and the CA certificate, click *Next*, *Finish*.

You have imported the signed certificate and the CA certificate.

5 Export the new certificate using `nlpexportcert` and save it in the `/etc/opt/novell/ldapproxy/conf/ssl/private` folder.

4.16 Setting the User DN Password

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
```

4.17 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 infirmation about configuring Redis server, refer to the *Redis documentation* (http://redis.io/documentation).

5 Using the NLPManager to Configure NetIQ LDAP Proxy

The NetIQ LDAP Proxy Manager (NLPManager) is a graphical utility that enables you to create and manage configuration files. It also helps you to monitor, analyze, and manage LDAP events.

- Section 5.1, “Using NLPManager,” on page 75
- Section 5.2, “Basic Configuration,” on page 76
- Section 5.3, “Configuring Additional Listeners,” on page 78
- Section 5.4, “Configuring Additional Back-End Servers,” on page 79
- Section 5.5, “Configuring Additional Server Groups,” on page 80
- Section 5.6, “Creating a New Configuration File,” on page 81

5.1 Using NLPManager

You use NLPManager to manage and configure 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 Section 5.6, “Creating a New Configuration File,” on page 81.
- Configure the events to be monitored. For more information, refer to Section 7.1, “Configuring Monitoring Activities,” on page 87.
- Manage the LDAP events for trend analysis. For more information, refer to Section 7.2, “Managing Trend Analysis,” on page 90.

5.1.1 System Requirements

Software Requirements

Before installing NLPManager on SUSE Linux Enterprise Server (SLES) or Red Hat Enterprise Linux (RHEL), you must install the 32-bit RPMs of the following libraries from the platform specific repositories.

- `expat`
- `libXrender`
- `libXfixes`
- `pango`
- `libXinerama`

- freetype
- libXcursor
- libXcomposite
- gtk2

Hardware Requirements

The minimum system requirements for using NLPManager are as follows:

- 1170 MB RAM
- 2 GB hard disk space

5.1.2 Downloading and Starting NLPManager

1. Download the NetIQ NLPManager from the NetIQ Downloads Website (https://dl.netiq.com/index.jsp).
2. Extract the `LdapProxy-linux.gtk.x86.zip`. file.
3. Run the following command:

   ```
   ulimit -n 65536
   ```

4. To launch NLPManager, run the following command:

   ```
   ./NLPManager
   ```

   The NetIQ LDAP Proxy Manager window is displayed. Initially, the Welcome page appears in the NetIQ LDAP Proxy Manager window. To view NLPManager user interface, close the Welcome page.

   The NLPManager UI consists of the following panes:

   - The Project Explorer pane that displays the hierarchal depiction of the configuration you define.
   - The Editor pane that acts as the editor for providing configuration details.

5.2 Basic Configuration

You can use NLPManager to define the following basic configuration:

- **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, but you can customize it to listen only on specific interfaces.
- **Back-end server:** The IP address/domain name and port number of the system where the back-end server is installed. For more information about configuring additional servers, refer to Section 5.4, “Configuring Additional Back-End Servers,” on page 79.
- **Back-end server group:** The back-end servers that form one group. 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.

To define the basic configuration for LDAP Proxy:

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

   The NetIQ LDAP Proxy Manager window is displayed.

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/ldapproxy/conf` directory and click *Open*.

The `conf` directory is 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 the 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 in the Editor pane. By default, one listener is configured to listen on all the interfaces. You need not make any changes to this configuration.

5 Click the *Backend Servers* option in the *Project Explorer* pane.

The *Backend Servers* tab is displayed.

5a Click each back-end server name and specify the following basic configuration details:

- **Name:** The name to identify the back-end server you are configuring. By default, the name of the first server is Backend1 and the name of the second server is Backend2.

  NOTE: All mandatory fields are marked in red.

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

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

  To provide the domain name of the 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*.

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

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

5b Click *Provisioning* > *Save* to save the changes.

NOTE: If LDAP Proxy is configured to use the basic configuration, the Backend1 and Backend2 servers must host the same Directory Information Tree.

6 Save the `nlpconf.xml` file in the `/etc/opt/novell/ldapproxy/conf` directory on the machine where you installed LDAP Proxy.

You can now start NetIQ LDAP Proxy. For more information, refer to Chapter 6, "Managing NetIQ LDAP Proxy," on page 85.

5.3 Configuring Additional Listeners

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, 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/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.

5.4 Configuring Additional Back-End Servers

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, 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/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 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.

5.5 Configuring Additional Server Groups

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/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` 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 Section 4.8, "Configuring Additional Policies," on page 39.

5.6 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/ldapproxy/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/ldapproxy/conf` directory on the machine where you installed LDAP Proxy.

The default `nlpconf.xml` file is replaced with the newly created configuration file.

6 Managing NetIQ LDAP Proxy

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

- Section 6.1, “Starting LDAP Proxy,” on page 85
- Section 6.2, “Stopping LDAP Proxy,” on page 85
- Section 6.3, “Restarting LDAP Proxy,” on page 85
- Section 6.4, “Checking the Status of LDAP Proxy,” on page 85
- Section 6.5, “Backing Up the LDAP Proxy,” on page 85

6.1 Starting LDAP Proxy

To start the LDAP proxy server, run the following command:

```
/etc/init.d/nlpd start
```

6.2 Stopping LDAP Proxy

To stop the LDAP proxy server, run the following command:

```
/etc/init.d/nlpd stop
```

6.3 Restarting LDAP Proxy

To restart the LDAP proxy server, run the following command:

```
/etc/init.d/nlpd restart
```

6.4 Checking the Status of LDAP Proxy

To check the status of the LDAP Proxy server, run the following command:

```
/etc/init.d/nlpd status
```

6.5 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/ldapproxy/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.

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

7 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.

- Section 7.1, "Configuring Monitoring Activities," on page 87
- Section 7.2, "Managing Trend Analysis," on page 90
- Section 7.3, "Enabling Monitoring and Trending," on page 93

7.1 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 50 in Section 4.8.4, "Search Request Policy," on page 47.

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

   The NetIQ LDAP Proxy Manager window is displayed.

2. Do one of the following:

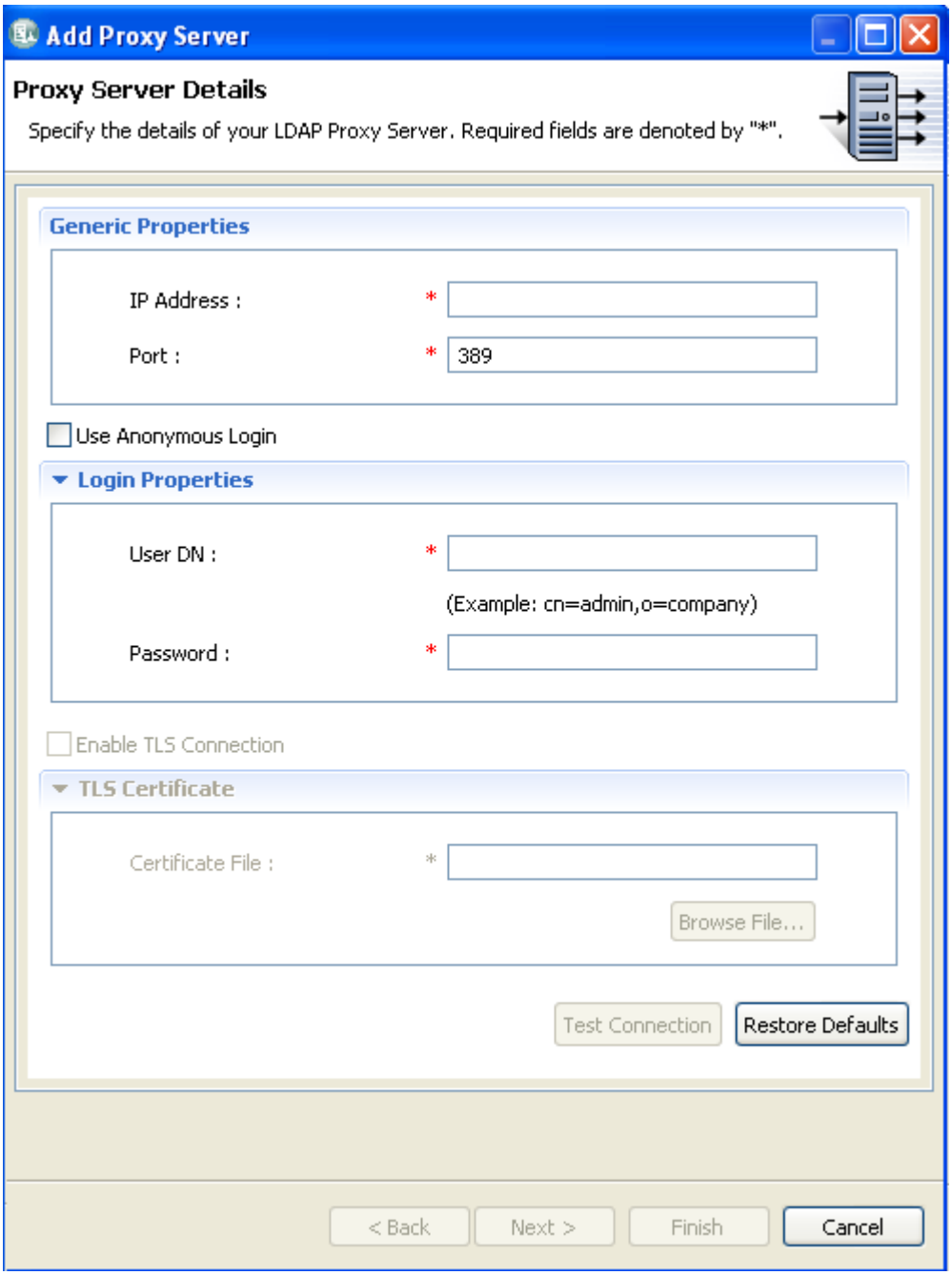
   - Click the  icon.
   - In the *Live Monitor* menu, click *Live Monitor*.

   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.



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.

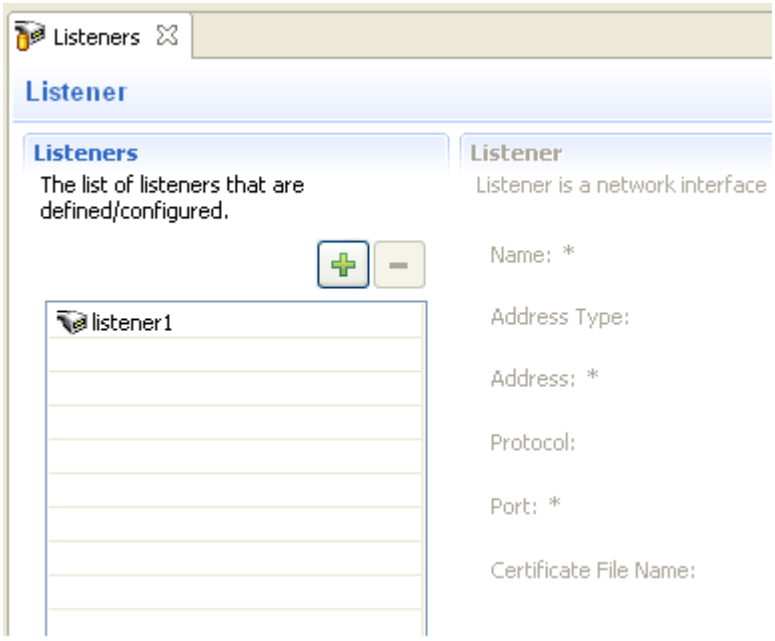
Currently, the TLS Certificate section is disabled.

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.

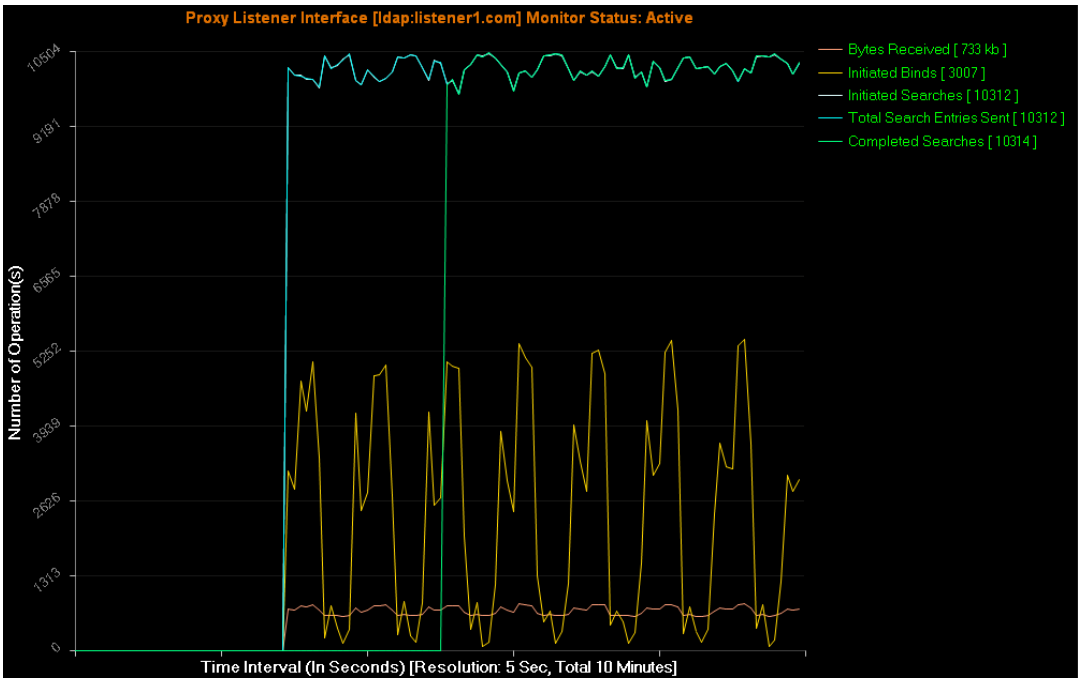
4b To define the events that must to be monitored, right-click the graph and click *Monitor Options*.

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 7-1.

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*.

7.2 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 `./NLPManager` command to start the NLPManager.

The LDAP Proxy window is displayed.

2 Do one of the following:

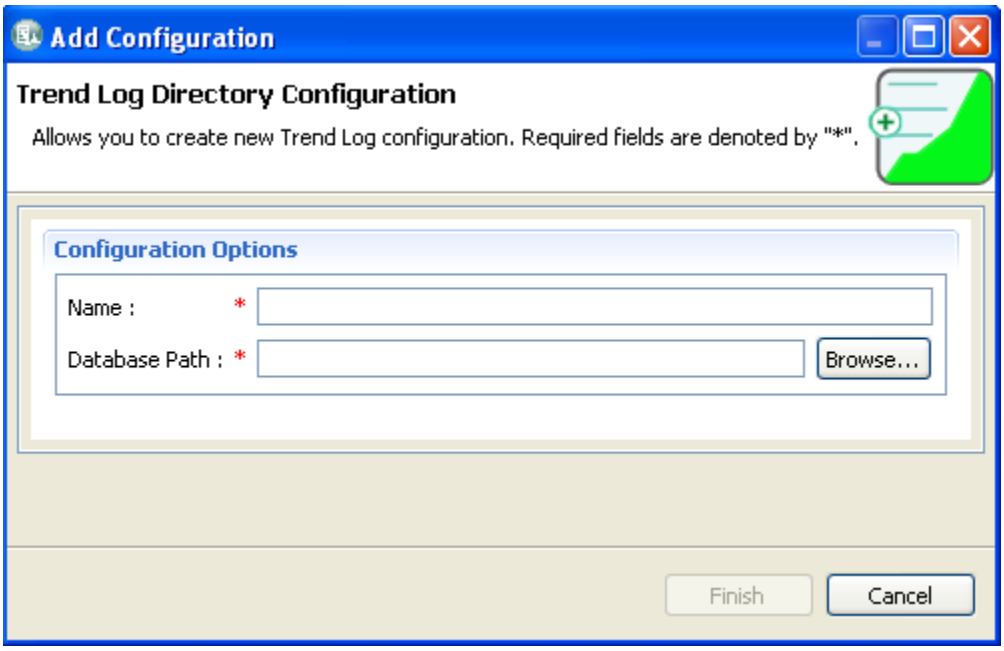
- Click the  icon.
- In the *Trend Manager* menu, click *Trend Manager*.

The *Trend Manager Configuration* tab is displayed.

3 Create a trend configuration:

3a Click the  icon.

The Add Configuration dialog is displayed.



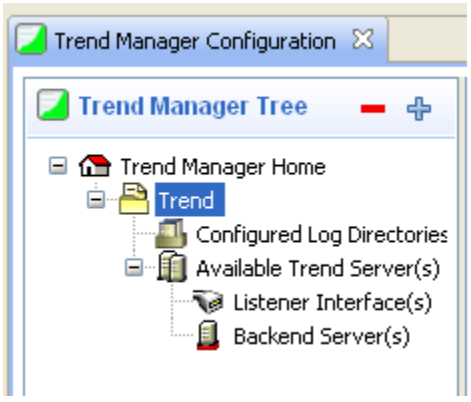
3b In the Add Configuration window, provide the following information:

- **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. This 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 7-1.

11 Specify the time period for trending:

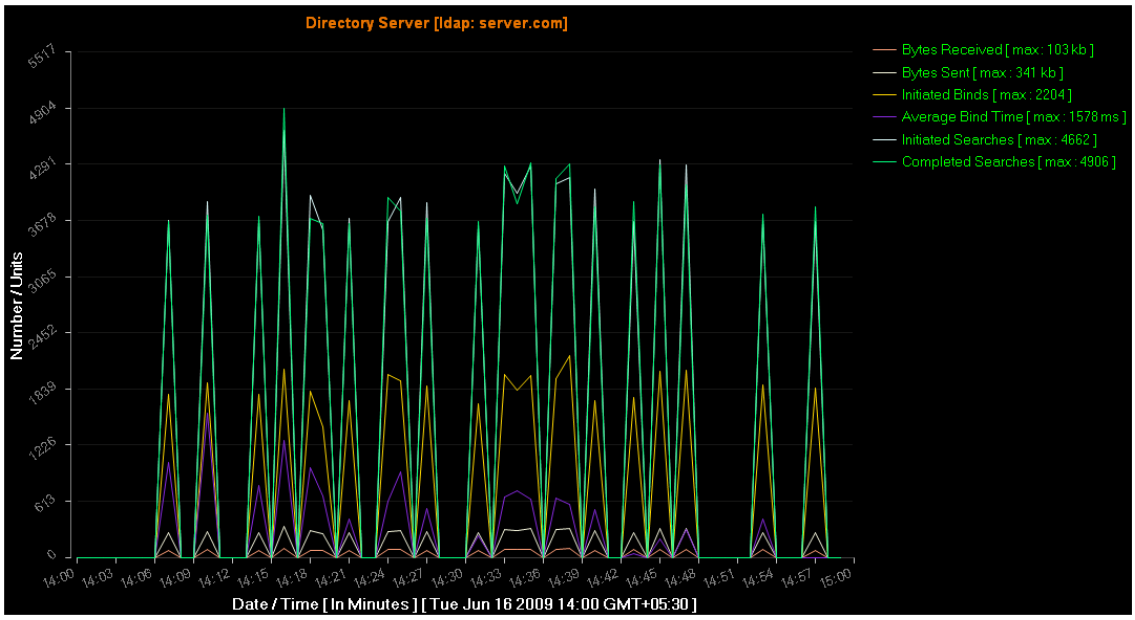
11a Select the time unit in the *Trend Unit* drop- down list. The available options are Hour, Day, Week, Month, Quarter, and Year.

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.



7.3 Enabling Monitoring and Trending

When you enable live monitoring and trending, you must specify the events to be analyzed. Table 7-1 provides the list of event categories and events that can be monitored and analyzed.

Table 7-1 *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.

Event Category	Event	Description
	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.
	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.

Event Category	Event	Description
	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.
	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 DNs	The number of LDAP Modify DN requests initiated by an application or user.
	Completed Modify DNs	The number of LDAP Modify DN requests that were initiated and successful.
	Failed Modify DNs	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.

Event Category	Event	Description
	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.
	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.

8 Enabling an LDAP Proxy Trace

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/ldapproxy/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-filesize-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/ldapproxy/conf` directory.
6. To commit the changes to the LDAP Proxy, run the following command:

   ```
   /etc/init.d/nlpd refresh
   ```

The trace log files are located in the `/var/opt/novell/ldapproxy` 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"/>
```

A Configuring a Linux High Availability Cluster for NetIQ Ldap Proxy

This chapter describes how to configure a high availability cluster for LDAP Proxy in Linux.

It includes the following sections:

- Section A.1, “Software Requirements,” on page 99
- Section A.2, “Hardware Requirements,” on page 99
- Section A.3, “Installation iSCSI Target,” on page 100
- Section A.4, “Configuring a NetIQ Ldap Proxy Setup for HA,” on page 101

A.1 Software Requirements

iSCSI Target

- SLES 11
- iSCSI Target
- YaST2-iSCSI-server

HA Cluster Nodes

- SLES 11
- open-iSCSI
- YaST2-iSCSI-client
- HA Pattern
- NetIQ LDAP Proxy

A.2 Hardware Requirements

iSCSI Target

- A minimum of 1 network card
- Enough disk space to share as an iSCSI partition
- YaST2-iSCSI-server

HA Cluster Nodes

- 2 network interface cards per node (NIC). One for external access, the other for Heartbeat private connection.
- Crossover network cable (for private HA connection).

A.3 Installation iSCSI Target

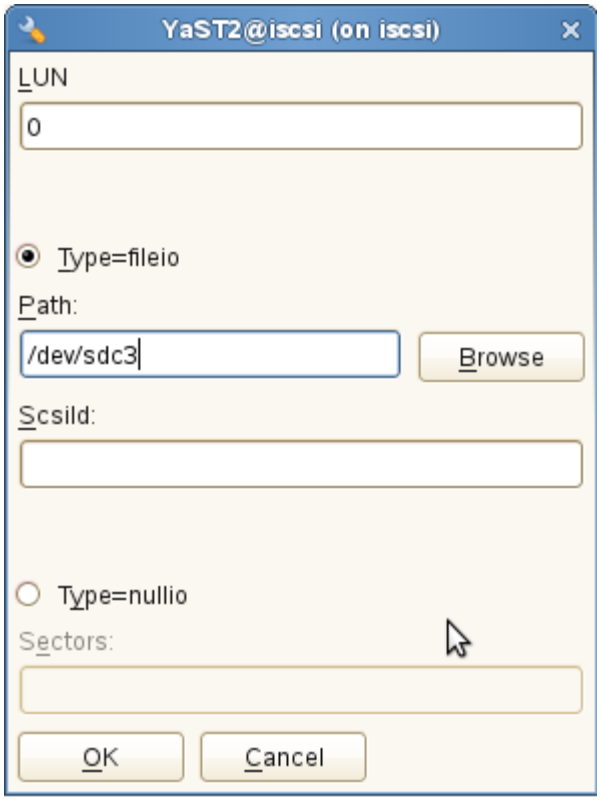
Before installing iSCSI Target, you must install SLES 11. While installing SLES 11, ensure that you perform the following tasks:

- Create a separate partition (for example, /dev/sdc3) for the iSCSI shared storage partition.

  At th NetIQ Test Labs, ReiserFS showed more stability than Ext3. However, you may use any supported file system. This partition is mounted by `/etc/fstab` on the local machine as `/shared`.
- Include the YaST2-iSCSI-server and the iSCSI target packages.

When the SLES 11 install is completed, set it up as the iSCSI server as follows:

1. Launch YaST.
2. Click *Network Services* > *iSCSI Target*.
3. In the *Service* tab, set *Service Start* to *When Booting*.
4. In *Global*, specify the required authentication details. (No authentication is used in this example.)
5. In the *Targets* tab, click *Add* twice, and specify the partition (`/dev/sdc3`) in *Path* field, as shown in the following figure. Retain the default values for *Target*, *Identifier*, and *LUN*.



6 Click *Finish* to restart iSCSI services.

iSCSI shared storage is now be available to the HA nodes.

A.4 Configuring a NetIQ Ldap Proxy Setup for HA

This section includes the following topics:

- Section A.4.1, "Configuring Node 1," on page 101
- Section A.4.2, "Configuring Node 2," on page 102
- Section A.4.3, "Configuring the Constraints," on page 105

A.4.1 Configuring Node 1

Install SUSE Linux Enterprise Server 11.

While configuring Node 1, set up one network interface card for the externally facing IP address, and the another NIC for an internal address that will be used by HA. In this example, the hostname is `node1` and the NICs are: `eth0` is 192.0.0.11 (external), and `eth1` is 10.0.0.1 (private HA).

Configuring an iSCSI Setup for Node 1

1 Execute the `mkdir /shared` command.

2 Launch YaST2.

3 Click *Network Services* > *iSCSI Initiator*.

4 In the *Service* tab, set *Service Start* to *When Booting* and leave *Connected Targets* empty.

5 In *Discovered Targets*, click *Discovery*. This locates the iSCSI target server's partition an populates it.

6 Enter the iSCSI target server's IP address (you can also retain the default port).

7 Click *Log in (No Authentication). Discovered Targets*. The Connected field is automatically populated with the value `true`.

8 Go to *Connected Targets* and set *Start-Up* to `automatic`.

9 Click *Finish*.

10 Execute the `dmesg` command to make the SCSI device `/dev/sdb` available.

11 Mount the iSCSI target `/dev/sdb` as `/shared`, by executing the `mount -t reiserfs /dev/sdb /shared` command.

You have configured an iSCSI setup.

Installing NetIQ LDAP Proxy for Node 1

1 Before initiating the installaion process, manually create a virtual adapter `ifconfig eth0:0 192.0.0.1`. This is the virtual IP address of your HA cluster.

2 Install NetIQ LDAP Proxy. For more information about how to install LDAP Proxy, see "Installing LDAP Proxy" in the *NetIQ LDAP Proxy 1.5 Installation Guide*.

3 Set the LDAP Proxy path to . `/opt/novell/ldapproxy/bin/nlppath`.

4 Configure the LDAP Proxy instance as follows:
   - The configuration files must be placed on the `/shared, nlp.conf in /root/` folder.
   - The proxy server to listen on the HA virtual IP address 192.0.0.1.
5 Verify whether the LDAP Proxy server is up and running on Node 1.
6 Shut down Proxy server, by executing the `/etc/init.d/nlpd stop` command.
7 Copy the following files to the `/shared` directory:
   - The `conf` folder present in the `/etc/opt/novell/ldapproxy` folder.
   - The `log` folder present in the `/var/opt/novell/ldapproxy` directory.
   - The `nici` folder. Create a symbolic link `/var/opt/novell/nici` in the `/shared/nici` folder.
8 Change the Proxy paths for `config` and `log` directories in the `/shared/conf/nlpconf.xml` file, as follows:

```
<proxy-paths>
  <dir-config>/shared/ldapproxy/conf</dir-config>
  <dir-log>/shared/ldapproxy/log</dir-log>
</proxy-paths>
```

9 Modify the `init` script (`/etc/init.d/nlpd`). The path of `nlpconf.xml` is fixed in the `init` script and you must modify it to a variable default_conf_file. In this example, `default_conf_file=/shared/conf/nlpconf.xml`.

NOTE: If you do not want to modify the init script, you can create a symbolic link `/etc/opt/novell/ldapproxy/conf` in `/shared/conf` and copy the files in shared directory as mentioned in Step 7. You must modify the `nlpconf` file available in the `/shared` location otherwise changes will not take effect.

Disabling nlpd Start at Boot Time

1 In YaST, navigate to *System* > *System Services (Runlevel)*.
2 Disable *nlpd start at boot*. Alternatively, you can edit the appropriate files in the `/etc/rc.d runlevels` file.
3 Click *Finish*.

NLPD is shut down on Node 1.

A.4.2 Configuring Node 2

Before configuring Node 2, perform the following steps in Node 1:

1 Shut down the NLPD process, if running, by executing the `/etc/init.d/nlpd stop` command.
2 Ensure that the NLPD process has stopped and then execute the `umount /shared` command. Else, the `/shared` folder will not unmount as expected.
3 Release the virtual IP address, by executing the `ifconfig eth0:0 down` command.

Perform the following steps in Node 2:

1 Install SLES 11.
2 Set up one NIC for the externally facing IP address, and the another NIC for an internal address that will be used by HA. In this example, the hostname is `node2` and the NICs are: `eth0` is 192.0.0.12 (external), and `eth1` is 10.0.0.2 (private HA).

Configuring an iSCSI Setup for Node 2

1. Execute the `mkdir /shared` command.
2. Launch YaST.
3. Click *Network Services* > *iSCSI Target*.
4. In the *Service* tab, set *Service Start* to *When Booting* and leave *Connected Targets* empty.
5. In *Discovered Targets*, click *Discovery*.
6. Enter the iSCSI target server's IP address (you can also retain the default port).
7. Click *Log in (No Authentication). Discovered Targets*. The Connected field is automatically populated with the value true.
8. Go to *Connected Targets* and set *Start-Up* to `automatic`.
9. Click *Finish*.
10. Execute the `dmesg` command to make the iSCSI device `/dev/sdb` available.
11. Mount the iSCSI target `/dev/sdb` as `/shared`, by executing the `mount -t reiserfs /dev/sdb /shared` command.

You have configured an iSCSI setup for Node 2.

Installing NetIQ LDAP Proxy for Node 2

To install NetIQ LDAP Proxy for Node 2, follow the steps mentioned in "Installing NetIQ LDAP Proxy for Node 1" on page 101.

To maintain consistence, you can switch to Node 1, by performing the following steps:

On Node 2

1. Shut down the NLPD process, if running, by executing the `/etc/init.d/nlpd stop` command.
2. Ensure that the NLPD process has stopped and then execute the `umount /shared` command. Else, the `/shared` folder will not unmount as expected.
3. Release the virtual IP address, by executing the `ifconfig eth0:0 down` command.

On Node 1

1. Manually create a virtual adapter `ifconfig eth0:0 192.0.0.1`, which will be the virtual IP address of the HA cluster.
2. Mount the iSCSI target `/dev/sdb` as `/shared`, by executing the `mount -t reiserfs /dev/sdb /shared` command.
3. Set the eDirectory path as `. /opt/novell/ldapproxy/bin/nlppath`.
4. Start NLPD, by executing the `/etc/init.d/nlpd start` command.

Configuring IP Resource

1. Click the *Resources* tab.
2. On the *Primitive* tab add a new primitive.
3. Create clusterip resource as follows:
   - ID: `clusterip`
   - Class: `ofc`

- Provider: `heartbeat`
- Type: `IPaddr`
- Initial state of resource: Retain the default value `Started` or select `Inherit from its parent`.
- Add Monitor Operation: Select this option.

4 On the *Instance Attribute* tab, add `ip 192.0.0.1` and `nic= eth0:0`.

5 On the *Meta Attribute* tab, add `is-managed = True` and `resource-stickiness = 100`.

6 On the *Operation* Tab, add `Monitor`, `Start` and `Stop` with default values.

Configuring File System Resource

1 Click the *Resources* tab.

2 On the *Primitive* tab add a new primitive.

3 Create clusterip resource as follows:

- ID: `Shared_Resource`
- Class: `ofc`
- Provider: `heartbeat`
- Type: `Filesystem`
- Initial state of resource: Retain the default value `Started` or select `Inherit from its parent`.
- Add Monitor Operation: Select this option.

4 On the *Instance Attribute* tab, add `device = /dev/sdc`, `directory =/shared` and `fstype = reiserfs`.

5 On the *Meta Attribute* tab, add `is-managed = True` and `resource-stickiness = 100`.

6 On the *Operation* Tab, add `Monitor` with default values.

Configuring NetIQ LDAP Proxy (NLPD) Resource

1 Click the *Resources* tab.

2 On the *Primitive* tab add a new primitive

3 Create clusterip resource as follows:

- ID: `NLPD_Process`
- Class: `ofc`
- Provider: `heartbeat`
- Type: `NetIQLDAPProxy`
- Initial state of resource: Retain the default value `Started` or select `Inherit from its parent`.
- Add Monitor Operation: Select this option.

4 On the *Instance Attribute* tab, add `device = /dev/sdc`, `directory =/shared`, and `fstype = reiserfs`.

5 On the *Meta Attribute* tab, add `is-managed = True` and `resource-stickiness = 100`.

6 On the *Operation* Tab, add `Monitor`, `Start` and `Stop` with default values.

A.4.3 Configuring the Constraints

Resource Colocation

Create colocation constraint, by specifying the following values:

- ID: `NLPD_Process`
- Resource: `clusterip`
- With Resource: `NetIQLDAPProxy`
- Score: `Infinity`
- Resource Role: `Started`
- With Resource Role: `Started`

Resource Order

Add IP and NLPD process order, by specifying the following values:

Resource Order

- ID: `IP_NLPD`
- Resource: `clusterip`
- With Resource: `NetIQLDAPProxy`

Resource Colocation

- ID: `IP-Shared_Resource`
- Resource: `Shared_Resource`
- With Resource: `clusterip`

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.

- Section B.1, “Sample Entries,” on page 107
- Section B.2, “Using the Proxy as a Directory Firewall,” on page 108
- Section B.3, “Mapping the Schema Based on the Network and Users,” on page 108
- Section B.4, “Setting a Search Base for User Identities,” on page 108
- Section B.5, “Preventing Wild Card Search Filters,” on page 108
- Section B.6, “Configuring Access Control Based on Users,” on page 108

B.1 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:

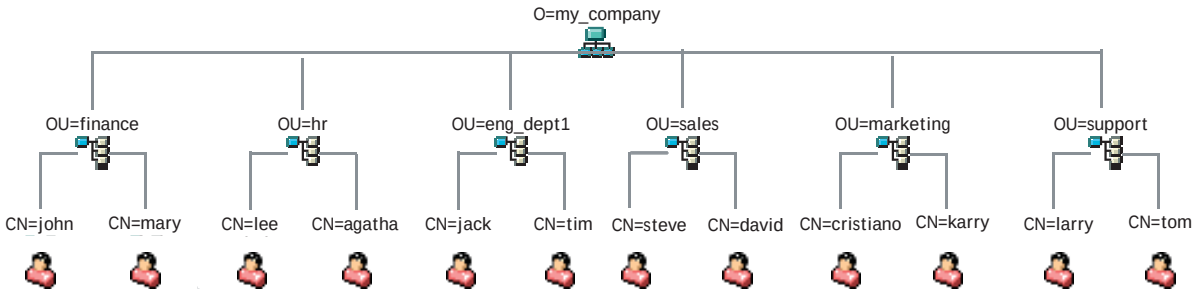
- 192.168.5.3:389 for backend1
- 192.168.5.4:2389 for backend2
- 192.168.5.5:1389 for backend3

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*



B.2 Using the Proxy as a Directory Firewall

The `nlpconf_NetworkRestriction.xml` sample configuration file in the `/etc/opt/novell/ldapproxy/conf-sample` directory shows how to configure LDAP Proxy to act as a directory firewall to allow only LDAP requests from particular networks. In the sample, a Network Restriction policy is used to control the incoming LDAP requests.

B.3 Mapping the Schema Based on the Network and Users

The `nlpconf_MappingSchemaBasedOnUsers.xml` sample configuration file in the `/etc/opt/novell/ldapproxy/conf-sample` directory helps you to understand how LDAP Proxy can be configured to map an application schema with the back-end server schema based on the 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

B.4 Setting a Search Base for User Identities

The `nlpconf_SearchBasedOnUserIdentity.xml sample configuration file` available in the `/etc/opt/novell/ldapproxy/conf-sample` directory enables a proxy server to automatically append a search base to default containers when a base is not supplied.

B.5 Preventing Wild Card Search Filters

The `nlpconf_PreventWildSearch.xml sample configuration file` in the `/etc/opt/novell/ldapproxy/conf-sample` directory defines enables a proxy server to prevent users from sending the cn=* type of LDAP searches.

B.6 Configuring Access Control Based on Users

The `nlpconf_AccessBasedOnUsers.xml` sample configuration file in the `/etc/opt/novell/ldapproxy/conf-sample` directory configures a particular set of servers to behave as read-only for all users except users from ou=support,o=my_company.

C Error Codes

The following table contains the error codes for NetIQ 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 *NetIQ LDAP Proxy 1.5 Installation Guide*.
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.

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

Error Code	Remedy
0x81590005	**Cause:** An invalid parameter was found in the LDAP request/response.
0x81590006	**Cause:** An invalid LDAP request was received.
0x8159000A	**Cause:** The specified LDAP object was not found.
0x81590100	**Cause:** An LDAP protocol error was encountered.
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 Section 4.4, “Basic Configuration,” on page 29.
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 Section 4.4, “Basic Configuration,” on page 29.
0x815B0101	**Cause:** An unsupported policy was specified in the configuration.
	Action: Ensure that the configuration is consistent with the documentation. For more information, see Section 4.4, “Basic Configuration,” on page 29.
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 Section 4.4, “Basic Configuration,” on page 29.
0x815B0104	**Cause:** Unsupported conditional operator was specified.
	Action: Ensure that the configuration is consistent with the documentation. For more information, see Section 4.4, “Basic Configuration,” on page 29.
0x815B0105	**Cause:** An unsupported conditional matching criterion was specified.
	Action: Ensure that the configuration is consistent with the documentation. For more information, see Section 4.4, “Basic Configuration,” on page 29.

Error Code	Remedy
0x815B0106	**Cause:** A request is denied as per policy. **Action:** Ensure that the connection-route policy for that particular connection/request should be denied.
Back-end Errors	
0x815C0002	**Cause:** The back-end subsystem ran out of memory. **Action:** Increase free memory by shutting down services or increase the RAM.
0x815C0100	**Cause:** A particular back-end server is not available. **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.
0x815C0101	**Cause:** The maximum connection limit for a back-end server was exceeded. **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.
XML Errors	
0x815E0002	**Cause:** The XML Logger/XML parser ran out of memory. **Action:** Increase free memory by shutting down services or increase the RAM.
0x815E0005	**Cause:** The XML parser encountered an invalid parameter. **Action:** Check the configuration for XML errors.
0x815E000A	**Cause:** The XML parser could not find an expected XML object. **Action:** Check the configuration file to see if it is consistent with the documentation. For more information, see Section 4.4, “Basic Configuration,” on page 29.
0x815E0100	**Cause:** The XML parser failed to parse. **Action:** Check the configuration file for XML errors.
0x815E0101	**Cause:** The XML parser failed to parse the XML DOM. **Action:** Internal error.
0x815E0102	**Cause:** Could not create the DOM. **Action:** Internal error
0x815E0103	**Cause:** Could not initialize the DOM. **Action:** Internal error.
0x815E0104	**Cause:** Could not serialize XML DOM. **Action:** Internal error.

Error Code	Remedy
0x815E010A	**Cause:** Multiple nodes were found when only one was expected. **Action:** Ensure that the configuration file is consistent with the documentation. For more information, see Section 4.4, “Basic Configuration,” on page 29.
Configuration Errors	
0x815F0001	**Cause:** The configuration directive is not implemented. **Action:** Check the documentation for the supported policy, conditions, or actions. For more information, see Section 4.4, “Basic Configuration,” on page 29.
0x815F0002	**Cause:** The configuration subsystem ran out of memory. **Action:** Increase free memory by shutting down services or increase the RAM.
0x815E010A	**Cause:** The configuration is incomplete. It does not specify a required configuration object. **Action:** Check the configuration file for errors.
0x815E010C	**Cause:** A policy object already exists. **Action:** Multiple policies must have separate IDs.
0x815F0100	**Cause:** Could not read the configuration file or schema configuration file. **Action:** Ensure that the `nlpschemaconf.xml` schema configuration file is in the default location of `/etc/opt/novell/ldapproxy/conf/` and that the permissions are sufficient.
0x815F0101	**Cause:** The Schema configuration file is invalid. **Action:** Check the `nlpschemaconf.xml` schema configuration file for errors.
0x815F0102	**Cause:** A data value specified in the configuration file is invalid. **Action:** Check the configuration file for XML errors or other inconsistencies .
0x815F0103	**Cause:** An XML node specified in the configuration file is invalid. **Action:** Check the configuration file for XML errors or other inconsistencies.
0x815F0104	**Cause:** Two listener nodes were found with the same ID. **Action:** Multiple listener configuration nodes must have different IDs.
0x815F0105	**Cause:** The directory path to the log files specified in the configuration is invalid. **Action:** Check to see if the directory path specified is correct.
0x815F0106	**Cause:** Unable to create the directory path for logging as specified in the configuration. **Action:** Ensure that the appropriate permissions are in place for the directory path specified.

Error Code	Remedy
0x815F0107	**Cause:** The domain name specified in the configuration could not be resolved. **Action:** Check your DNS configuration or specify an IP address instead of DNS name.
0x815F0108	**Cause:** Reading the network interface corresponding to the IP/Domain Name given in the configuration failed. **Action:** Ensure that your network settings for the IP/Domain Name are bound to a network interface as expected.
Audit/Event Subsystem Errors	
0x815D0001	**Cause:** An audit event was received but has not been implemented. **Action:** Check documentation for the supported audit events. For more information, see Section 4.12, “Configuring Audit Events,” on page 68.
0x815D0002	**Cause:** The Audit subsystem ran out of memory. **Action:** Increase free memory by shutting down other services or increase the RAM.
0x815D0005	**Cause:** The ID of the event object is invalid. **Action:** Internal error.
0x815D0006	**Cause:** The audit subsystem received an invalid request. **Action:** Internal error.
0x815D0008	**Cause:** The audit subsystem failed to register a producer event because no event data was found. **Action:** Internal error
0x815D000A	**Cause:** The specified event object was not found. **Action:** Internal error.
0x815D000C	**Cause:** An event object with the same ID is already present. **Action:**
0x815D0100	**Cause:** Invalid event data was received. **Action:** Internal error.