

UnboundID[®] Directory Proxy Server

Product Description

Version 3.2

September 21, 2011



UnboundID Corp.
13809 Research Blvd Suite 500
Austin, TX 78750

512-600-7700
www.UnboundID.com

Copyright

Copyright ©2011 UnboundID Corp.
All rights reserved.

This document constitutes an unpublished, copyrighted work and contains valuable trade secrets and other confidential information belonging to UnboundID Corp.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

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

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

UnboundID® Directory Proxy Server

Product Description

Introduction

Today's businesses are driving new revenues with large-scale, Web-based, customer-facing applications. As a result, the directory service is an increasingly important component of identity management. Data centers are under increased pressure to manage rising technology costs, while maximizing processing efficiencies and meeting aggressive performance objectives. Many corporations are leveraging virtualization technologies, adopting energy-saving green strategies, and automating security, risk, and compliance processes as user activity and enterprise mobility systems increase. Your identity management system must also expand to meet these demanding requirements.

To meet these challenges, UnboundID® Corp. has developed a suite of directory services products. This document focuses on the UnboundID® Directory Proxy Server, an LDAP proxy server that provides high availability, fault tolerance, load distribution and additional security for the UnboundID® Directory Server or other compliant LDAPv3 servers, all while remaining invisible to client applications. The directory proxy server includes a rich set of features to meet the needs of your unique deployment, from transforming data, to fine-tuned failover and load-balancing policies. And since the UnboundID Directory Server and Directory Proxy Server share a common code base, they work together seamlessly.

This document presents the features and benefits of the UnboundID Directory Proxy Server.

Corporations are consolidating their data center operations through reductions or mergers yet must plan for user account growth and rising web application usage. Your directory services must be able to meet these demands.

The staff at UnboundID Corp. has a long track record of successful LDAP technology deployments.

The UnboundID Product Family

The UnboundID product family integrates disparate systems to provide a robust and high performance directory services solution. The **UnboundID Directory Server** offers all of the standard LDAPv3 server functions, plus powerful relational database features, such as transactions, joins, and event notifications. The **UnboundID Directory Proxy Server** is a fast, scalable, load-balancing server that seamlessly distributes client requests to the backend servers. It provides high availability and added security for the UnboundID Directory Server while remaining invisible to client applications, helping your data center run smoothly. The **UnboundID® Synchronization Server** provides fast, low-latency data synchronization between directory systems and platforms. The **UnboundID® LDAP SDK for Java** is a feature-rich API for client communication with LDAPv3 directory servers.

The UnboundID products are deployed in production systems around the world.

This document describes in detail the UnboundID Directory Proxy Server and how it interacts with other products, whether in the UnboundID product family or third-party products. **Figure 1** illustrates the following key benefits of the directory proxy server:

- Works seamlessly with third-party directory servers.
- Provides transparent failover and load balancing between directory servers.
- Provides entry balancing to automatically distribute large datasets across multiple systems without requiring any changes to your data or the way that clients access it.

For more information about the UnboundID offerings, go to the UnboundID web site at:

www.unboundid.com

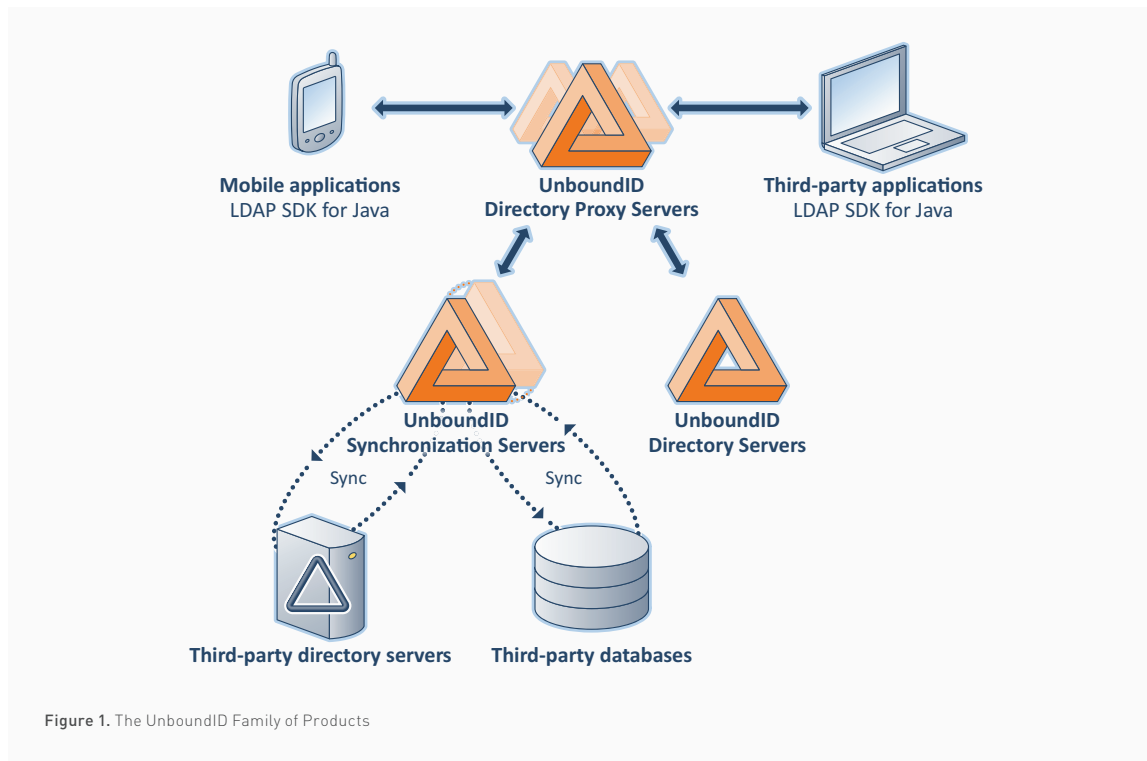


Figure 1. The UnboundID Family of Products

What is the UnboundID Directory Proxy Server?

While network load balancers are popular choices for providing fault tolerance and rough load distribution across a topology of directory servers, they are not adept at providing advanced request routing based on anything beyond the individual TCP connections.

The UnboundID Directory Proxy Server is a fully LDAPv3-compliant proxy server designed to allow the finest-grained control over load distribution for LDAP clients. The directory proxy server provides an intelligent gateway to your directory environment and includes the following key features:

- High-performance, topology-aware load balancing and failover.
- Attribute and DN mapping, which allows you to control what the directory looks like on a per-application basis.
- Entry balancing, which allows you to seamlessly split large data sets among smaller, commodity systems without altering the data or clients in any way.
- Client access limits on the backend directories, including what part of the DIT is exposed, what operations are allowed, the rate at which clients are allowed to issue requests, and so on.
- Control over different types of connections, including who, what, when, where, and how the client is connecting (bind DN, IP address, time of day, whether the connection is secure or not, and so on).
- Simple configuration and management, so that managing a large topology is as easy as managing a single instance.
- Extensive logging, monitoring and alerting capabilities.
- Advanced health checking, to identify potential problems with backend servers as quickly as possible, including problems that may not be immediately visible but could still cause problems for clients (for example, if a server falls behind in replication so that it is serving stale data).

Many of these capabilities can be configured in a fine-grained manner so that different clients experience the directory environment in different ways. For example, you may want to restrict a particular client so that it can only access certain portions of the DIT or request certain types of operations.

The UnboundID Directory Proxy Server is a pure Java proxy server that provides speed, scalability, and ease of use, plus additional features you won't find anywhere else.

UnboundID Directory Proxy Server Capabilities At-A-Glance

RELIABILITY	
<p>Automatic Failover</p> <ul style="list-style-type: none"> • Forwards client requests away from backend servers that become unavailable or operate in a degraded capacity • Forwards client requests in a location-aware manner so that backend servers in the local data center are preferred over those in remote locations • Uses both proactive and reactive mechanisms to monitor the health of backend servers so that it can quickly detect and respond to problems before they are visible to clients 	<p>Benefits</p> <ul style="list-style-type: none"> • Client requests are serviced as usual despite changes to backend server availability and dataset structure. From a client perspective, the directory proxy server looks like a directory server, even though it does not contain the data
<p>Safe Integration of New and Legacy Systems</p> <ul style="list-style-type: none"> • Allows you to gradually phase in new backend servers by controlling how much client load they receive until you are comfortable that they can meet the demands of the environment • Allows you to gradually phase in new client applications so that they are only allowed to access a subset of the directory servers until you are comfortable that they are well-behaved 	<p>Benefits</p> <ul style="list-style-type: none"> • Allows you to safely introduce changes in your directory environment in a way that minimizes the potential impact if the changes produce undesirable results
FLEXIBILITY	
<p>Data Transformations and Topology Awareness</p> <ul style="list-style-type: none"> • Maps DNs and attributes, allowing legacy clients to interact with the server using older names for directory content • Unifies the contents of backend servers into a common namespace, for example, in the case of a merger 	<p>Benefits</p> <ul style="list-style-type: none"> • Can continue to use legacy client applications despite the new DIT and schema served by the backend directory servers or inherent design limitations in the clients
<p>Synchronization Through the Proxy Server</p> <ul style="list-style-type: none"> • Allows the configuration of the proxy server as an external server for a synch source or synch destination • Provides the ability to configure a single synch pipe for an entire entry-balanced sub-tree 	<p>Benefits</p> <ul style="list-style-type: none"> • No need for the synch server to know if the external servers are directory or proxy servers • Simplifies the configuration and on-going maintenance of the synch server in an entry-balanced environment
ADDITIONAL SECURITY	
<p>Security and Access Control</p> <ul style="list-style-type: none"> • Allows you to prevent direct access to the backend servers and restrict access to only clients that meet certain criteria, such as IP ranges, authentication state, or communication security level • Introduces constraints and filtering to help protect the directory server from attacks • Can be placed in a DMZ, allowing you to avoid giving clients direct access to the backend servers in the internal network or providing the data in the DMZ 	<p>Benefits</p> <ul style="list-style-type: none"> • Ensures that customer, employee, and company information is secure • Provides secure access to the data, while you define what actions clients are allowed to perform
PERFORMANCE	
<p>Horizontal Scalability</p> <ul style="list-style-type: none"> • Horizontally scales operations using load balancing. Directory proxy server can make the directory server layer appear as an even higher-performance directory 	<p>Benefits</p> <ul style="list-style-type: none"> • Spreads the work load across multiple servers, or even multiple data centers, to effectively utilize available resources • Ensures high availability by avoiding servers that are unavailable or operating in a degraded state
<p>Better Throughput for Very Large Datasets</p> <ul style="list-style-type: none"> • Uses entry balancing to divide large datasets among multiple sets of directory servers • Provides an in-memory global index that can quickly determine which server(s) should be used to process requests, avoiding the need to split the data algorithmically and minimizing the need to broadcast requests to multiple servers 	<p>Benefits</p> <ul style="list-style-type: none"> • Performance improves because more servers can handle client requests • Fits more entries in memory for better performance, while allowing you to use a lower class of hardware • Improves write throughput. Though searches can scale up without entry balancing, writes cannot because the writes are replicated

UnboundID Directory Proxy Server Capabilities At-A-Glance (cont.)

TOTAL COST OF OWNERSHIP	
<p>Entry Balancing</p> <ul style="list-style-type: none"> Allows you to split large datasets across smaller, less expensive servers rather than requiring systems large enough to hold the entire dataset 	<p>Benefits</p> <ul style="list-style-type: none"> Not necessary to invest in more expensive hardware to support massive datasets
<p>Platform Support</p> <ul style="list-style-type: none"> Runs on any system with a compliant Java SE 6 virtual machine. Designed for platform independence 	<p>Benefits</p> <ul style="list-style-type: none"> Can use existing hardware. Server can be moved unmodified between system architectures, lowering total cost of ownership
MONITORING AND MANAGEMENT	
<p>Operation Tracking</p> <ul style="list-style-type: none"> Includes information in the communication with backend servers that makes it easy to correlate information about an operation in the backend directory server with information about that same operation in the directory proxy server, and vice versa 	<p>Benefits</p> <ul style="list-style-type: none"> Simplifies administration of complex topologies Allows administrators to easily track requests through the directory service
<p>Server Configuration</p> <ul style="list-style-type: none"> Supports scripted and interactive command-line tools and a web console Records every server configuration change, the user who made it, and how to revert it 	<p>Benefits</p> <ul style="list-style-type: none"> Ensures a simple means of accessing the configuration Ensures security and accountability with the ability to playback or revert configuration changes
<p>Topology Awareness</p> <ul style="list-style-type: none"> Allows you to specify the locations of the servers in your topology so that the directory proxy server can prefer to forward requests to backend servers in the local data center over those in remote locations 	<p>Benefits</p> <ul style="list-style-type: none"> Provides an intuitive means for defining connection preferences and failover rules Makes it easier to consider geographic location in the process of selecting backend servers
<p>Logging</p> <ul style="list-style-type: none"> Provides an extensive set of logging capabilities, including access, error, and debug logging Allows multiple loggers of each type, each with its own configuration Supports filtered access logging to provide fine-grained control over which types of messages are logged on a per-logger basis Supports logging to different targets, including files, relational databases, and syslog servers Includes information in log messages that makes it easy to correlate information about an operation in the directory proxy server with information about that operation in backend servers 	<p>Benefits</p> <ul style="list-style-type: none"> Ensures fast and effective troubleshooting and auditing to meet the needs of your data center
<p>Administration Alerts</p> <ul style="list-style-type: none"> Provides an alert framework that can be used to notify administrators about errors, warnings, or significant events that occur in the directory environment Delivers alerts as log messages, e-mail messages, SNMP traps, and/or JMX notifications Allows you to configure the directory proxy server to execute a specified command whenever an alert is generated Allows you to access over LDAP the alert information generated by the server. This information can also be consumed using persistent searches Consumes and reacts to administrative alerts generated by UnboundID Directory Server instances to better detect problems with backend servers 	<p>Benefits</p> <ul style="list-style-type: none"> Ensures fast and effective management in the event of a problem or event Custom alerts can be tailored to specific production environments for more efficient management
<p>Real-Time Monitoring</p> <ul style="list-style-type: none"> Consolidates all monitor entries in a single location Exposes all monitor information via LDAP, JMX, and the console Provides APIs so that you can programmatically access monitor information 	<p>Benefits</p> <ul style="list-style-type: none"> Simplifies the location of monitoring information Allows custom components to expose their own monitoring information

Reliability from Intelligent Load Balancing

Network load balancers use a few basic algorithms to route a broad range of network protocols. If a port on a backend server is not responding to a health check for one or more consecutive attempts, the load balancer marks the server as inactive. If the backend servers are LDAP servers, the sophistication of the health check is, at best, a basic LDAP connect and search for a well-known entry. While this approach is reasonable for a device responsible for balancing traffic across a variety of protocols, it is grossly inadequate when a directory server exposes more relevant information describing its state. Further, network load balancers can only control routing at a per-connection level, so all operations on a given connection will always be sent to the same server, even if that server begins experiencing problems.

The UnboundID Directory Proxy Server provides advanced load-balancing algorithms along with numerous properties that govern connectivity and measure health. This added intelligence results in a more robust environment for reliably managing LDAP traffic. Plus, its load balancing mechanism is operation based rather than connection based, so operations on the same client connection can be forwarded to different backend servers. The directory proxy server can even retry an operation on a different server if it fails on the first server that was selected for it.

Load-Balancing Algorithms

The directory proxy server uses load-balancing algorithms to determine which backend server(s) should be used in the course of processing client requests. The load-balancing algorithm may consider a number of factors in selecting the target servers, including the health and location of each of the backend servers, the type of operation that was requested, the configuration of the load-balancing algorithm, and other activity in progress in the directory proxy server.

The following load-balancing algorithms are available for use:

- **Single server:** All requests will be forwarded to the same backend server.
- **Fewest operations:** Requests will be forwarded to the backend server with the fewest outstanding requests.
- **Round robin:** Requests will be spread evenly across all of the backend servers, in a round-robin manner.
- **Weighted:** Requests will be forwarded to the backend servers in accordance with administratively-defined weights, such that servers with a larger weight receive a correspondingly higher percentage of client requests than servers with a lower weight.
- **Health weighted:** Requests will be forwarded to the backend servers in accordance with the health check score assigned to each server, such that servers with a higher health check score will receive a correspondingly higher percentage of client requests than servers with a lower health check score.
- **Failover:** All requests will be consistently forwarded to the same backend server as long as it remains available. If the server becomes unavailable or degraded, all requests will then be forwarded to the primary alternate server, then to the secondary alternate server, and so on.

With the exception of the single-server algorithm, which only knows about one backend server, all of the load-balancing algorithms can also take into account information about the health and location of the backend servers. Servers that are fully available will be preferred over those that are degraded or unavailable, and servers in the same location as the directory proxy server will be preferred over servers in other data centers.

The UnboundID Directory Proxy Server may also be configured to establish an affinity between clients and the backend servers used to process requests from those clients, allowing subsequent requests to be consistently forwarded to the same backend server used to process earlier requests for that client. This affinity can help avoid problems that arise because of delays in replication. For example, if a client issues a search request to retrieve an entry immediately after adding or modifying that entry, you will want the read to go to the same server as the previous write to ensure that the client retrieves the most up-to-date copy of the entry, in the event that the change has not yet propagated to all other servers.

Load balancing spreads the workload across multiple backend servers for better performance, scalability, and availability.

Load balancing algorithms can take many factors into account when selecting which backend servers to use, including the health and location of the servers. It may also consider the servers that were selected for previous requests from the same client.

Backend Server Health and Automatic Failover

The directory proxy server provides precise controls for monitoring the health of your backend servers. Server health is classified into the following states:

- **Available:** The server is running with no issues.
- **Degraded:** The server should be avoided if possible, but may be used if there are no other options.
- **Unavailable:** The server should not be used.

The state of available and degraded also have a score, a numeric value between ten (the best) and one (the worst). Some load-balancing algorithms, such as the health-weighted algorithm, may use this score to differentiate between servers with the same state. Most of the time, the health check state and score are dynamically determined using health checks, but you can also explicitly configure the state for a server. For example, you may want to configure the state to ensure that no traffic is forwarded to a particular server that is going to be taken offline for maintenance.

The directory proxy server health checking mechanism operates in both a proactive and a reactive manner. Periodically, the directory proxy server evaluates the health of each of the backend servers to ensure that they remain acceptable for use. But, in the event that a forwarded operation fails, it may choose to immediately invoke another health check so that it can quickly react to changes in backend server availability. If the directory proxy server detects a significant change in the health of any of the backend servers, it generates an alert to notify administrators of the problems and immediately adjusts its load-balancing policy to consider this new information when deciding where to forward requests.

The types of health checks that the UnboundID Directory Proxy Server may perform are described in **Table 1**.

Health Check Type	Description	Server Availability
Admin Alert	Watch for and consume any administrative alert generated by a backend directory server and evaluate whether that alert may indicate any change in availability for the server.	UnboundID Directory Server backend instances only.
Replication Backlog	Monitor the replication state for each of the backend servers to ensure that none of them falls too far behind, either in number of outstanding changes or in the age of the oldest change that has not yet been replicated.	UnboundID Directory Server backend instances only.
Work Queue Busyness	Monitor the state of the work queue for backend servers to ensure that the number of operations waiting to be processed by a worker thread does not grow too large.	UnboundID Directory Server backend instances only.
Search LDAP	Send a search request to each backend server and evaluate the success of the search, the number of entries returned, the contents of those entries, and/or the length of time required to process the search.	Any type of backend server.

Table 1. UnboundID Directory Proxy Server Health Check Types

The UnboundID Directory Proxy Server provides advanced health-checking capabilities.

The UnboundID Directory Proxy Server can automatically detect and avoid backend servers that may be experiencing problems and can notify administrators so that they can investigate the issue.

Safe System Integration

The directory proxy server makes it easy to introduce changes into your directory environment while minimizing the impact of those changes if unforeseen problems occur. For example, when adding new backend servers, or when upgrading existing backend servers, you can configure the directory proxy server to initially send only a small percentage of the overall traffic, or only traffic from a subset of applications, to those servers until you are comfortable that they are configured properly and will hand the needs of your directory environment. If a problem occurs, the affected operations can be automatically retried on a different server so that the failure is not visible to the clients.

Similarly, if you are deploying a new directory-enabled application, you can configure the directory proxy server so that requests submitted by that application are only forwarded to a subset of the backend servers. In such a deployment, if a problem arises with the new application (for example, it requests inefficient search operations that take a long time to complete and consume a lot of available resources in the backend servers), then only a subset of the servers are impacted, so that other servers can handle the load from other applications.

Entry Balancing for Large Datasets

Entry balancing provides an efficient way to support very large datasets by transparently splitting entries across two or more sets of servers. The directory proxy server maintains in-memory indexes to efficiently forward LDAP client requests, including requests that span multiple backend servers, so that they are only sent to the server containing the data needed to fulfill them.

At startup, or at the request of a server administrator, the directory proxy server can be configured to quickly build compact, in-memory indexes of the directory data, either from the backend directory servers or from a peer proxy server, making the first LDAP client request for entry balanced data as efficient as possible. The directory proxy server uses a robust, fault-tolerant indexing model, so that it is able to transparently handle changes to indexed data (including the addition of new entries, or the modification, renaming, or deletion of existing entries), even for changes made to those servers outside of the directory proxy server.

Entry balancing improves scalability and performance by spreading entries below a common parent among multiple sets of directory servers.

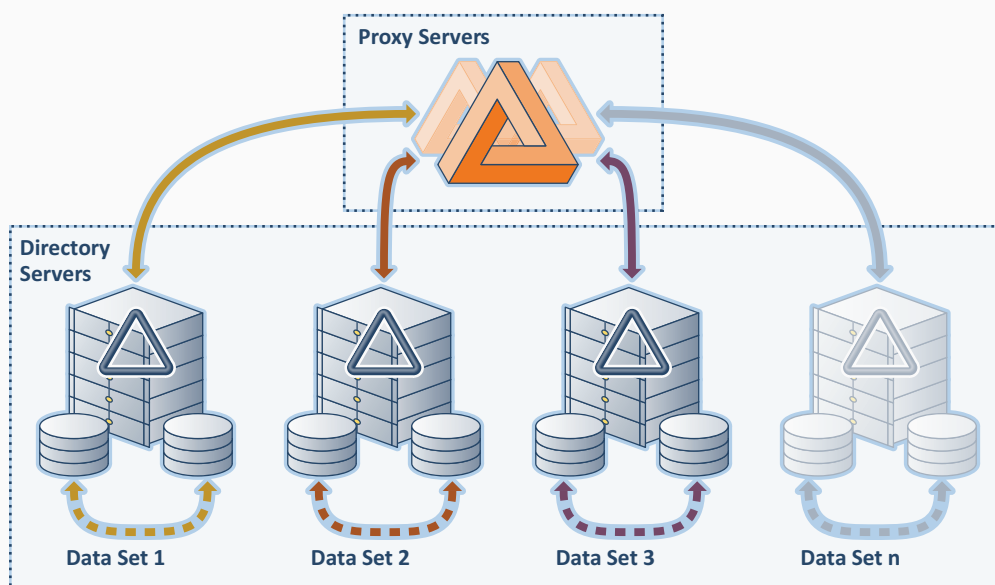


Figure 2. Balancing Entries across Three Sets of Servers

Data Transformations

When the UnboundID Directory Proxy Server receives a request from an LDAP client, it can be configured to apply transformations to the request before it is forwarded on to any backend server. It can also apply transformations to any responses returned for the request.

Proxy data transformations alter the contents of client requests and the server response to these requests.

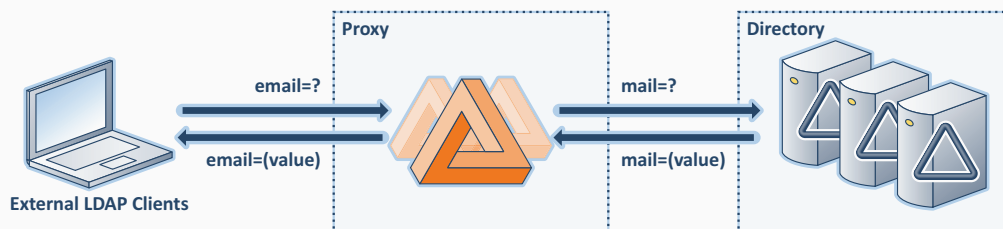


Figure 3. A Proxy Data Transformation

Available transformations include:

- **Attribute mapping:** The directory proxy server can transparently rename any attributes referenced in requests from clients and in responses from the server back to those clients. It performs a thorough mapping, including references to the target attributes in entries, DN's, search filters, referrals, and even control values. This mapping makes it possible to support applications that expect information to be stored in one attribute when it is actually held in an attribute with a different name.
- **DN mapping:** The directory proxy server can transparently alter any DN's referenced in requests from clients and in responses from the server back to these clients. It performs a thorough mapping, including the values of attributes with a DN syntax, search result references, and even control values. This mapping makes it possible to support applications that expect a DIT structure which differs from what the backend servers provide.
- **Default value:** The directory proxy server can automatically insert a specified set of values for a target attribute in add requests before they are sent to backend servers, or in search result entries before they are returned to clients. If the entry already contains one or more values for the target attribute, then the existing values may be retained, replace by, or merged with the default values.
- **Suppress attribute:** The directory proxy server can be configured to prevent clients from accessing a specified attribute. You can merely remove the attribute from search result entries, or configure a transformation that rejects requests attempting to reference the attribute (including add operations that try to provide a value for the attribute, modify or modify DN operations that try to alter the attribute, compare operations that use the specified attribute in the assertion, or search operations that use the specified attribute in the filter).
- **Suppress entry:** The directory proxy server can be configured to prevent entries that match a specified filter from being returned to clients. When using this transformation, search requests will be altered so that the filter includes a NOT clause containing the exclusion filter. This filter ensures that the request is processed by the backend server, which has access to the full entry, rather than performing the filter evaluation in the directory proxy server, which may only have access to a portion of the entry and may not be able to accurately determine whether the entry matches the exclusion filter.

Data transformations can be controlled on a per-client basis using a client connection policy. The client connection policy makes it possible for the directory proxy server to provide access to the same content in very different ways to different clients.

Topology Aware

Directory services are commonly deployed and replicated over wide geographical areas. The UnboundID Directory Proxy Server makes it possible to assign a location to each of the backend servers, and that location can be taken into account when selecting which servers should be used to process requests. When forwarding requests, the load-balancing algorithm will first try to select a server in the same location as the directory proxy server instance to minimize the network latency incurred. If none of the backend servers in the same location as the directory proxy server are available, then it can fail over to servers in an alternate location. Each location can be configured with an ordered list of preferred failover locations, so that you can control the order in which this remote failover occurs. You can also indicate whether to prefer geographic location over server availability (for example, whether to prefer a degraded server in the local data center over an available server in a remote data center).

Topology awareness makes it easy for you to configure the directory proxy servers to consider the location of the backend servers when selecting which one to use for a given request.

The location mechanism of the directory proxy server is specifically designed to simplify the configuration of server instances across data centers. In most deployments, it should be possible for all directory proxy server instances to have exactly the same configuration in all data centers, with the exception of the global configuration property that specifies the location of the directory proxy server instance itself.

Security and Access Control

The UnboundID Directory Proxy Server may be used to enhance the overall security of the directory environment. The directory proxy server honors the access control configuration of the backend servers, but may also be used to enforce additional restrictions on communications, on a server-wide or per-client basis.

The UnboundID Directory Proxy Server can act as a kind of firewall to restrict the types of requests that clients may issue, to limit the data that clients may access, and to minimize the impact that poorly designed or malicious clients may have on the directory environment.

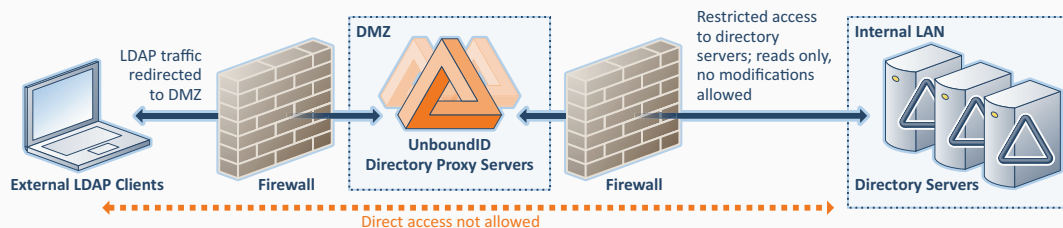


Figure 4. Restricting Client Access Using Directory Proxy Server

SSL and StartTLS Capabilities

The directory proxy server provides full support for secure communication using SSL and StartTLS. It may be used as an SSL or StartTLS termination point for client connections, and it may also use SSL or StartTLS to secure communication with backend servers. The directory proxy server supports the use of SASL EXTERNAL authentication, so that clients may authenticate to the directory environment using digital certificates. It also supports the use of PKCS#11 tokens for potential hardware acceleration and secure key storage.

The UnboundID Directory Proxy Server provides support for secure communication and can differentiate the service it provides to clients based on the type of communication they are using.

The directory proxy server also provides the ability to alter the restrictions that it imposes on clients based on the type of connection-level security in place. For example, a client using an unencrypted communication channel may be constrained so that it can only access certain portions of the DIT or request a limited set of operations. A client using a secure connection could have fewer restrictions, and clients that authenticate using a client certificate via SASL EXTERNAL could have an even greater level of access.

Restricting Client Access

As with the UnboundID Directory Server, the UnboundID Directory Proxy Server can be configured to enforce limits on the resources that clients may consume, including:

- **Restrict the set of allowed clients:** You can configure the directory proxy server to enforce restrictions on the set of clients that are allowed to connect to the server. These restrictions may be based on a number of factors, including the address of the client, the protocol it is using, and the type of communication security. For clients that have authenticated, you can define further restrictions based on the location and content of the authenticated user's entry and the type of authentication performed.
- **Restrict the number of concurrent connections:** You can configure the server to enforce an upper limit on the total number of client connections that can be established at any given time. You can also restrict the number of concurrent connections from the same client system (by IP address), by the same user (by bind DN), or by clients associated with the same client connection policy. Any new connections received that exceed any of these limits will be terminated. You can also define restrictions on the total length of time that client connections may remain established, and the length of time they can remain established with no activity.
- **Restrict the operation rate:** You can configure the server to enforce limits on the number of operations that clients may request in a given time interval, both on a per-client basis as well as for all clients associated with a given client connection policy. If a client attempts to exceed this limit, the directory proxy server may reject any further requests from that client until the end of the associated time interval, or it may immediately terminate the connection. You can define multiple rate limits over different intervals to allow for different long-term and short-term behavior, such as limiting clients to no more than one million operations per day, but allowing for bursts of up to one thousand operations per second.
- **Restrict the type of LDAP operation that a client can perform:** The directory proxy server offers very fine-grained support for restricting the types of operations that clients may request. At a coarse level, you may control the allowed operation types (such as abandon, add, bind, compare, delete, and search), but for certain types of operations you may enforce additional restrictions. For bind operations, you may indicate whether simple and/or SASL authentication should be allowed, and for SASL authentication you may control which mechanism should be permitted. For extended operations, you may define restrictions on which request OIDs may be used. For searches, you may enforce limits on the types of filters that can be used, as well as enforcing upper bounds on the size and time limits and the minimum substring length. For all types of requests, you may enforce restrictions on the types of controls that may be included with those requests.
- **Restrict the portions of the DIT that a client can access:** Each client connection policy includes a list of the base DNs that should be accessible to clients associated with that policy. Each policy can further be configured with different load-balancing algorithms and data transformations to control where the accessible data comes from and how it appears.

Entry and Attribute Suppression

Once the directory proxy server has forwarded a request to the directory server and the directory server has produced a response, you can configure what parts of the response are passed along to clients. The directory proxy server can be configured to suppress certain attributes or substitute the values of that attribute with a predefined set of default values. You can also configure the directory proxy server to suppress entire entries out of the result set based on filters.

You can use this feature when you do not want to modify the access control rules for the entire directory environment, but still want to limit what is available to certain clients, such as those accessing a directory proxy server instance in a DMZ. This capability is also very useful for limiting access to data from clients that might not otherwise be restricted, such as root users that are not subject to access control evaluation.

You can limit the data available to clients without modifying access control rules using attribute and entry suppression.

Performance

The UnboundID Directory Proxy Server is designed to be the most efficient way to intelligently forward LDAP requests, while at the same time meeting your existing performance service-level agreements. The directory proxy server does not require significant amounts of disk space or memory, and is well-suited for high-density servers like blades. You can scale smaller hardware horizontally to meet the demands of your environment without having to invest in more expensive, larger systems.

For very large datasets, the directory proxy server's entry balancing capabilities may further improve performance by allowing the data to be fully cached across a set of servers when it would not otherwise be possible for cost-effective to purchase systems large enough to cache the entire dataset. The use of entry balancing significantly increases the overall write throughput that a directory environment can sustain, since each portion of the data may be replicated independently.

Total Cost of Ownership

The UnboundID Directory Proxy Server's ability to stabilize the directory environment eases the complicated merging of new and legacy applications, lowering the total cost of ownership (TCO) over time. The entry balancing feature allows you to store very large datasets on many smaller, less expensive machines, making it unnecessary to invest in more expensive hardware to support massive datasets.

Platform Support

The UnboundID Directory Proxy Server is a pure Java application, fully portable to any platform that has a Java 6 runtime, including the following:

- Solaris™ (x86, x64, and SPARC systems)
- Linux® (x86, x64, Itanium, Power, and S/390 systems)
- Microsoft® Windows® (x86, x64, and Itanium systems)
- AIX® (Power systems)
- HP-UX® (PA-RISC and Itanium systems)
- Mac OS X® (x64 systems)
- Azul® Compute Appliance

Because the UnboundID Directory Proxy Server is compiled to Java bytecode rather than machine or OS-specific code, only a single version of the product is necessary to work on all platforms. For products compiled to native binaries, vendors are required to limit which platforms they can support and provide a separate set of binaries for each platform, including separate binaries for 32-bit and 64-bit architectures on the same platform. The UnboundID Directory Proxy Server allows administrators to easily manage a single build in a broad platform environment. In most cases, installing a new directory proxy server instance in an existing environment can be as easy as copying the contents of an already configured instance even if the new installation is on a system with a different OS or CPU architecture.

The UnboundID Directory Proxy Server is distributed in a single zip file for quick and easy installation.

Monitoring and Management

The UnboundID Directory Proxy Server provides tools to help administrators easily manage the directory environment. Administrators can manage the directory proxy server configuration using a set of command-line tools, the web-based Directory Proxy Server Management Console, or both. The directory proxy server also includes a robust operation tracking system to help administrators track client requests as they are forwarded through the system.

Server Configuration

The UnboundID Directory Proxy Server has a full-featured set of administration tools to install and manage the server. It provides a number of administrative interfaces, including a web-based administration console and a **dsconfig** command-line tool. The Directory Proxy Server Management Console is a graphical web application that provides access to the server's configuration. The console provides the same functionality for managing the configuration as the **dsconfig** command-line tool, and also provides easy access to monitor data and server documentation. Both tools allow you to safely manage the configuration of a whole topology of servers as easily as a single server.

The Directory Proxy Server configuration can be accessed using the command-line **dsconfig** tool and the web-based Directory Proxy Server Management Console.

The Directory Proxy Server Management Console

The Directory Proxy Server Management Console is a web application that allows administrators to use a web browser to easily view and manage their topology. The **dsconfig** Command-Line Tool

The Management Console provides an Object Type menu, which allows you to display the properties relevant to the administrator's experience level.

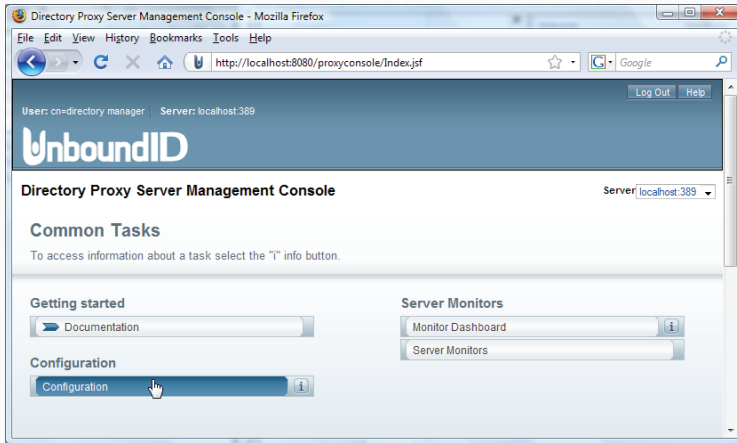


Figure 6. UnboundID Directory Proxy Server Management Console

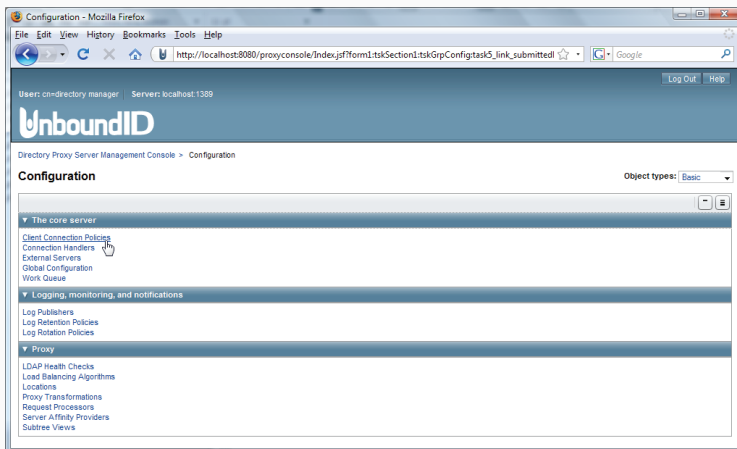


Figure 7. Directory Proxy Server Management Console Configuration Screen

The dsconfig Command-Line Tool

The **dsconfig** tool is a command-line utility with several modes of operation: a menu-driven interactive, a non-interactive mode that can be invoked using command-line arguments, and a batch mode in which configuration changes can be read from a file to make multiple changes in succession. Every configuration change made to a server is recorded in a configuration audit log that can be easily replayed using the **dsconfig** batch mode. This feature allows you to automatically configure other directory proxy servers in your topology using an existing configuration.

The dsconfig tool can be run interactively from the command line or within a script. All changes are recorded within a config-audit log that also records the commands you can use to back out of the change.

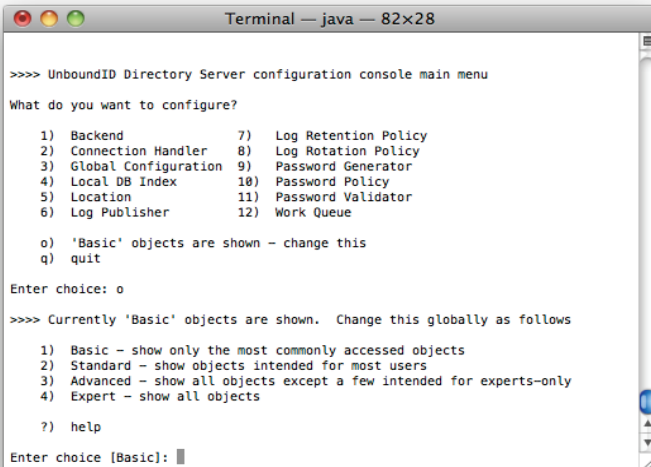


Figure 8. The dsconfig Command-Line Utility with Four Different Menu Levels

Both the web-based administration console and the **dsconfig** utility have many powerful functions for managing a directory proxy server environment. For example, the tools provide the ability to make configuration changes to multiple servers in a server group, rather than requiring administrators to manage each server individually, helping to avoid operation inconsistency. If you want to apply a configuration change to multiple servers, the configuration tool first verifies that all servers will accept the change before attempting to apply it anywhere.

Administrators can use server groups to safely apply a configuration change to multiple servers.

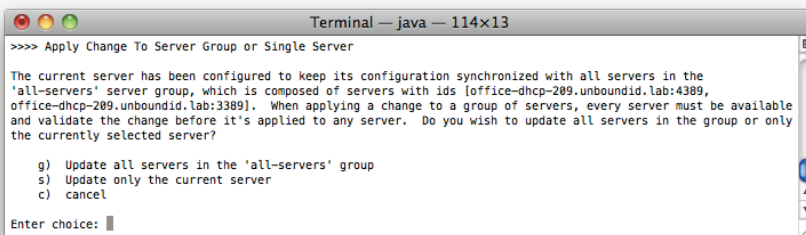


Figure 9. Making Changes to Multiple Servers in a Server Group

Configuration Audit Log

The UnboundID Directory Proxy Server also provides mechanisms for keeping track of configuration changes made over time, regardless of the tool used to make the changes. The configuration audit log provides a record of all configuration changes in a form that is compatible for use with the **dsconfig** batch mode for easy playback. It includes comments that indicate when the change was made, the administrator who requested the change, and a command that may be used to revert the change, if desired. In addition, the UnboundID Directory Proxy Server archives complete copies of each configuration that it has used, so you can see exactly what configuration was in use at any given time in the past.

The configuration audit log records when and by whom each change was made, along with an analogous undo command that can be used to revert the change.

Logging

The UnboundID Directory Proxy Server offers the same rich logging subsystem as the UnboundID Directory Server.

Some of the most significant logging features include:

- **Multiple loggers of any type:** The UnboundID Directory Proxy Server supports any number of active loggers of any type (access, error, or debug) in the server. Each logger has its own configuration and generally has a minimal impact on overall server performance.
- **Filtered access logging:** The UnboundID Directory Proxy Server provides the ability to control which types of access log messages are written on a very fine-grained level. Administrators can configure each access logger so that it will only include messages that meet certain criteria, including information about the client (such as its address, protocol, communication security level, authentication state), the request (such as the type of operation, the location and content of the target entry, any included request controls) and the result (such as the operation result code, the length of time required to process the request and any included response controls). For example, an administrator can easily create an access log that will only record information about operations that did not complete successfully, operations which took longer than some length of time to complete, operations targeting some portion of the DIT, or operations requested by a particular client application.
- **Multiple message consolidation:** The UnboundID Directory Proxy Server provides the ability to consolidate multiple pieces of information into a single access log message. Traditionally, one access log message is generated when the client establishes a connection, one message for each request received by the server, and one message with information about the result of processing each operation. The directory proxy server provides an option to consolidate all information about an operation in a single message for easy viewing. This message can be configured to include details about the request and results, as well as the address and authentication identity of the client that requested it. Often, applications keep connections open for a long time, so details about these connections may be buried in a log file that has been rotated or even deleted. On busy servers, log messages for requests and responses may be separated far apart in the same log file or even in different log files. In addition, providing all information about an operation on a single line is much easier to examine using text processing tools, like the UNIX **grep** command.
- **Centralized logging capabilities:** The UnboundID Directory Proxy Server provides capabilities for centralizing access and error log messages across multiple server instances. In addition to creating log files on the server file system, the directory proxy server can log to a UNIX syslog server and/or to a relational database. Log messages may include an instance name field, which can identify the directory proxy server instance that generated the log message. The directory proxy server can also log messages from the UnboundID Directory Server to the same types of repositories, and these messages may include a product name field to distinguish log messages generated by the directory server from those generated by the directory proxy server. Further, because the UnboundID Directory Server and the UnboundID Directory Proxy Server reset the connection ID counter whenever the server is restarted, log messages may also include a startup ID field to differentiate messages using the same connection ID that was assigned to an earlier connection before the server was restarted.
- **Intermediate Client Control:** The UnboundID Directory Proxy Server and UnboundID Directory Server support the Intermediate Client Control, a custom control to help track operations between multiple systems in the environment. For example, if the backend server is an UnboundID Directory Server instance, then the backend server log information for that request will include information about the address and port of the directory proxy server instance and the connection ID and operation ID used to identify that operation in the directory proxy server. Other applications can be developed to use this control (available in the Commercial Edition of the UnboundID LDAP SDK for Java) to include information about themselves to help better track requests from those applications through the directory environment. The control makes correlation possible, so that inefficient or abusive activity can be tracked to its origin.
- **Access/error log parsing APIs:** The Commercial Edition of the UnboundID LDAP SDK for Java (which is included with the UnboundID Directory Proxy Server) provides APIs for reading and parsing access and error log messages. A **summarize-access-log** tool is also provided with the UnboundID Directory Proxy Server, which uses this API to examine access log messages generated by the server and reports a number of metrics from the data those log files contain. These metrics include information about the number and percentage of each type of operation, the average length of time required to process each type of operation, a breakdown of the processing times into a number of predefined buckets, a breakdown of the result codes returned, and a breakdown of the most common search scopes, filter types, and entry counts.

The UnboundID Directory Proxy Server simplifies operation tracking for UnboundID Directory Server backend server instances by including key information in the access logs of both servers.

Administrative Alerts

The UnboundID Directory Proxy Server provides an administrative alert framework that can be used to notify administrators of any significant warnings, errors, or other noteworthy events that occur in the server. Existing alert-notification handlers can notify administrators through log messages, email, SNMP traps, or JMX notifications. You can configure the directory proxy server to execute a specified command whenever an alert is generated, with information about the alert available as command-line arguments. All administrative alerts are also exposed over LDAP as entries below a base DN of "cn=alerts", and you can use the persistent search operation ensure that you are automatically notified over LDAP of any new alerts generated by the server.

The administrative alert framework allows the UnboundID Directory Proxy Server to generate and handle alerts delivered by different means.

The administrator can select the events that apply for each type of notification based on the severity level or the specific type of alert. For example, it may be desirable to log information about all types of alerts, but only generate e-mail messages or SNMP traps for warnings and errors. Some sample events include the following:

- **Startup/shutdown:** Sends an alert when the directory proxy server completes the startup process or begins the shutdown process.
- **Applied configuration changes:** Sends an alert when a configuration change is applied to the server.
- **Scheduled task errors:** Sends an alert whenever there is a change in the health check state of any of the backend servers.
- **Disk space usage:** Sends an alert if the available disk space drops below a configured threshold.

Real-Time Monitoring

The UnboundID Directory Proxy Server exposes real-time monitoring information in a single branch (cn=monitor) and can be accessed using the Directory Proxy Server Management Console, Java Management Extensions (JMX), or directly over LDAP. The UnboundID Directory Proxy Server's monitoring information includes the following:

The UnboundID Directory Proxy Server's alerting system can be used with the monitoring framework to immediately notify administrators of any problems.

Active Connections	Provides information about all active connections, including when the connection was established, from which source address, which user established the connection, the number of operations completed, and the number of operations in progress.
Active Operations	Provides information about all active operations, including when the operation started, the connection that initiated the operation, the type of operation, and details about the operation itself (such as the search filter and base DN for a search operation).
LDAP Statistics	Provides information about the number of operations of each type that have been processed by the directory proxy server.
LDAP External Server	Provides information about the state of a backend server used by the directory proxy server, including the current health of the server, the number of connections established to the server, and the number of operations forwarded to the server.
Processing Time Histogram	Provides a breakdown per operation type of how long operations have taken in the directory proxy server. A cumulative count is stored for operations that took less than 1ms, 2ms, 3ms, 5ms, 10ms, and so on.
Work Queue Busyness	Provides the percentage of time that worker threads have been busy actively processing operations. Also exposes the recent, average, and maximum busyness percentage since startup.
Disk Space	Provides information about the disk utilization of all file systems used by the directory proxy server, represented as an absolute value and as a percentage of the overall file system size.
JVM Stack Traces	Provides a current stack trace of all threads in the directory proxy server.
System Information	Provides information about the system on which the Directory Proxy Server is running, including details about the hardware, operating system, and JVM being used.

The Commercial Edition of the UnboundID LDAP SDK for Java provides an API for retrieving and parsing various types of monitor entries from the UnboundID Directory Server and Directory Proxy Server.

LDAP SDK for Java

The UnboundID LDAP SDK for Java is a fast, user-friendly, and powerful Java API for client communications with LDAP directory servers. UnboundID developed the LDAP SDK for Java because of the stagnation, limited feature set, usability problems, and performance issues with other Java-based LDAP APIs. The LDAP SDK for Java provides more extensive features, stronger security, and enhanced functionality, and works with any LDAP version 3-compliant directory server.

The LDAP SDK for Java comes in two versions: a Standard Edition that is freely available for use with any type of LDAPv3 directory server, and a Commercial Edition, which is included with the UnboundID Directory Server and UnboundID Directory Proxy Server and provides all of the capabilities of the Standard Edition, as well as additional features specific to the UnboundID Directory Server and Directory Proxy Server. These capabilities include support for additional controls and extended operations and APIs for retrieving and interacting with administrative alerts, monitor entries, scheduled tasks, and log messages. For more information, see the LDAP SDK for Java web site at the following URL:

www.unboundid.com/products/ldapsdk

The UnboundID LDAP SDK offers better performance, better ease of use, and more features than other Java-based LDAP APIs.

About UnboundID Corp.

UnboundID is a leading provider of real-time identity management software for cloud, mobile and social applications. UnboundID is a privately-held company based in Austin, Texas and is funded by Silverton Partners.

For more information, visit www.unboundid.com.



UnboundID and the UnboundID logo are trademarks of UnboundID Corp.
All other product or service names are trademarks of their respective companies.