

UnboundID[®] Synchronization 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This document constitutes an unpublished, copyrighted work and contains valuable trade secrets and other confidential information belonging to UnboundID Corp.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® Synchronization Server

Product Description

Introduction

Today, data centers must contend with the rapidly increasing data demands resulting from the explosive growth of mobile smartphones, electronic records conversion in the health industry, and merger consolidations in the financial industry. While data centers integrate different hardware systems and OS configurations, data synchronization among these different systems is crucial to successfully manage the growth of new accounts and web applications.

To meet these challenges, UnboundID Corp. has developed a feature-rich, fast synchronization server that provides excellent performance (high throughput and low latency) while providing a robust, scalable, and extensible solution. This document presents the features and benefits of the UnboundID Synchronization Server.

Corporations must meet the challenge of successfully synchronizing data across disparate repositories in near real time. The UnboundID Synchronization Server meets these needs with a high-performance, highly reliable, yet light-weight approach.

UnboundID Corp. has many years of extensive synchronization experience with a proven track record of successful 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**, together with its performance and scalability, offers all of the standard LDAPv3 server functions, plus powerful Relational Database Management System (RDBMS) features, such as transactions, joins, and event notifications. The **UnboundID® Directory Proxy Server** is a fast, scalable load-balancing LDAP proxy server that seamlessly distributes client requests to the backend servers. The **UnboundID® Synchronization Server** provides a point-to-point, directory-centric solution for fast, low-latency data synchronization between directory systems. The **UnboundID® LDAP SDK for Java** is a feature-rich, fast, and user-friendly API for client communications with LDAPv3 directory servers.

For more information about the UnboundID Directory Server, UnboundID Directory Proxy Server, and the UnboundID LDAP SDK for Java, go to the UnboundID web site at:

www.unboundid.com

The UnboundID Synchronization Server is a key component of the UnboundID family of Identify Management products that includes the UnboundID Directory Proxy Server, the UnboundID Directory Server, and the UnboundID LDAP SDK for Java.

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

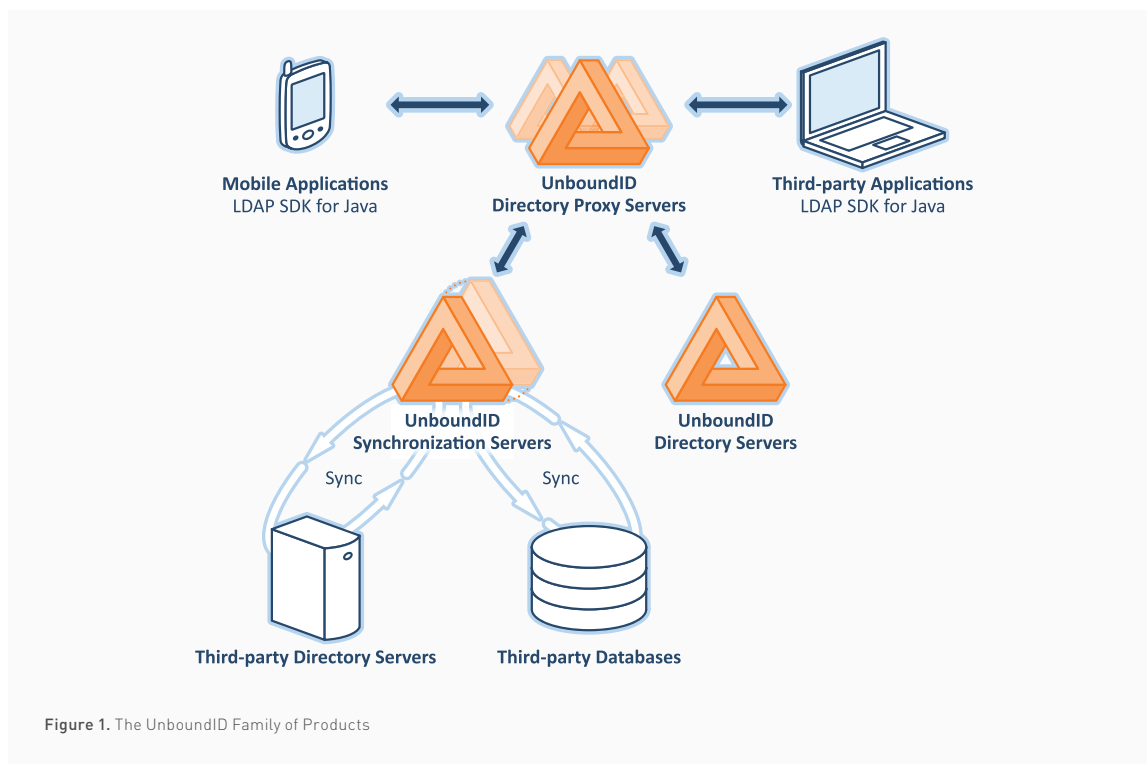


Figure 1. The UnboundID Family of Products

What is the UnboundID Synchronization Server?

The UnboundID Synchronization Server is an efficient, pure Java background server that supports high-scale, highly available data synchronization between the following directory servers:

- UnboundID Directory Server or UnboundID Directory Proxy Server
- Alcatel- Lucent 8661 Directory Server or Alcatel-Lucent 8661 Directory Proxy Server
- Sun™ Directory Server Enterprise Edition (DSEE 6.x, 7.x)
- Sun™ Directory Server (5.2 p3 or higher)
- Microsoft Active Directory

The UnboundID Synchronization Server also supports synchronizing data between one of the endpoints listed above with another endpoint consisting of a relational database management system (RDBMS). Official support is provided for synchronizing with the following relational databases:

- Oracle Database 10g and 11g
- Microsoft SQL Server 2005 and 2008

The architecture, however, does not make any assumptions about the type of database or schema being managed; any database with a Type 4 JDBC driver can be synchronized. Lastly, the UnboundID Synchronization Server, in conjunction with the Server SDK, provides a set of extension points that make it possible to synchronize with other endpoints, such as web services.

Designed to run reliably on inexpensive hardware with little administrative maintenance (for example, backups are not required), the UnboundID Synchronization Server provides an effective cost-per-performance solution for synchronizing identity data.

The synchronization server includes the following key features:

- High performance and availability with built-in redundancy to help ensure no downtime
- Dataless, virtual architecture for a small-memory footprint and ease of maintenance
- Hassle-free setup that allows you to transform and map attribute names, values, and DNs, allowing you to make schema and directory information tree changes without the added costs of custom coding and scripting
- Data flexibility and security, allowing you to replicate data and use advanced replication capabilities to provide filtered, fractional, or sub-tree replication scenarios
- A full-fledged notification service that allows applications to subscribe to receive messages based upon changes made to monitored data

UnboundID Synchronization Server Capabilities At-A-Glance

BIDIRECTIONAL SYNCHRONIZATION	
Flexibility and Reliability <ul style="list-style-type: none">• Provides a choice of real-time or scheduled bulk synchronization.• Supports bidirectional synchronization with complete, independent control over synchronization in both directions.• Support synchronizing in one direction only, for example, to keep a testing environment up-to-date with a scrambled version of production data.• Provides immediate failover in the event of a failure on the destination server, source server, or synchronization server itself.	Benefits <ul style="list-style-type: none">• Enables different architectures to operate in parallel.• Mitigates risks and enables more realistic test environments.• Synchronization is not interrupted by changes to backend server or synchronization server availability
DATA TRANSFORMATIONS	
DN and Attribute Mapping <ul style="list-style-type: none">• Allows you to define DN and attribute mapping, letting legacy applications 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.	Benefits <ul style="list-style-type: none">• Does not require complex, costly coding and scripting.
ADVANCED REPLICATION	
Flexibility and Security <ul style="list-style-type: none">• Allows replica servers to store a subset of the data or extra data.• Supports filtered, fractional, local data, and sub-tree replication scenarios. You can also combine these different types of replication to meet the needs of your deployment.• Provides the ability to comply with privacy rules and regulations, for example, allowing you to restrict what data crosses country borders.	Benefits <ul style="list-style-type: none">• Provides maximum flexibility for data storage.• Application specific directory instances require less hardware.• Only need to access the data required to meet the needs of the application.
ACTIVE DIRECTORY SYNCHRONIZATION	
Seamless Integration <ul style="list-style-type: none">• Synchronizes Active Directory with other corporate directories in real-time.• Supports secure real-time password synchronization.	Benefits <ul style="list-style-type: none">• Enables all directories to operate in parallel.• Adheres to the Microsoft® guidelines for password synchronization.
DATABASE SYNCHRONIZATION	
Seamless Integration <ul style="list-style-type: none">• Synchronizes relation database management systems (RDBMS) with directories in real-time.• Provides Server SDK extension points for defining the logic necessary to map and transform data between the relational and directory environments.	Benefits <ul style="list-style-type: none">• Enables the unification of identity data from a multitude of applications that rely on an RDBMS as their data store.• Provides unlimited flexibility in how data is mapped and transformed between relational and directory environments.
NOTIFICATION MODE	
Real-time Notifications <ul style="list-style-type: none">• Allows you to define arbitrary notification messages that can be sent to third-party applications based on changes made to monitored data.• Messages can be sent to any type of end-point.• Complies with 3GPP UDC Notifications specification.	Benefits <ul style="list-style-type: none">• Supports asynchronous integration with any third-party application.• Server SDK extensions provide maximum integration flexibility.• Provides a standards-based method for integrating data with other applications.

UnboundID Synchronization Server Capabilities At-A-Glance (cont.)

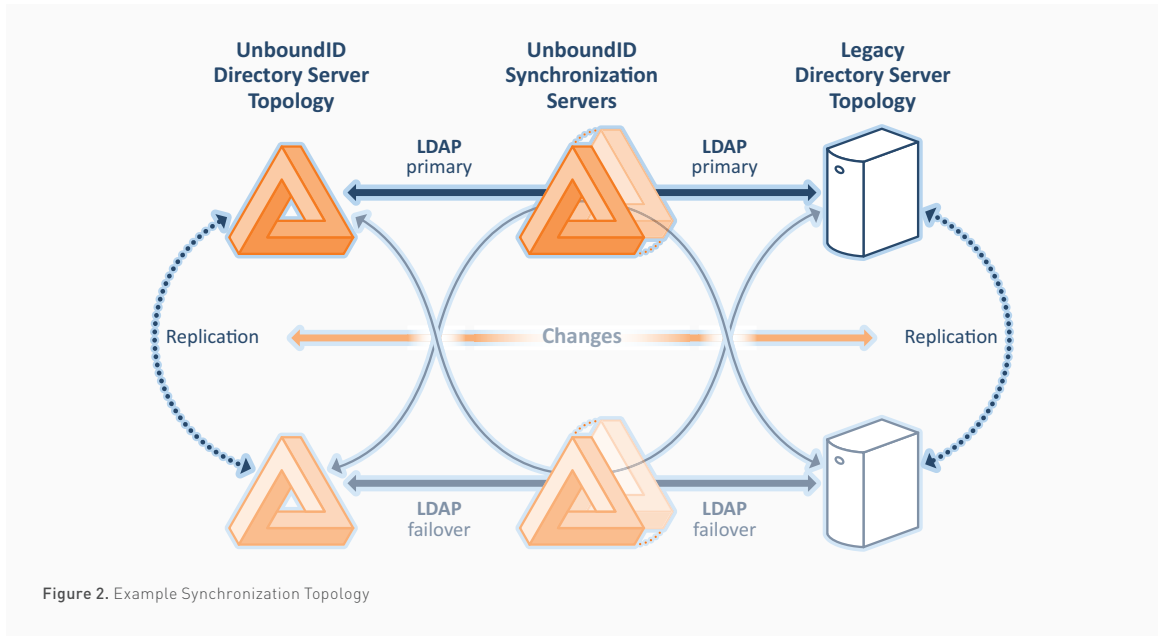
HIGH PERFORMANCE AND AVAILABILITY	
<p>Performance</p> <ul style="list-style-type: none"> • Handles the most demanding, high transaction application environments. • Processes changes in parallel to ensure high-throughput, even over a WAN. 	<p>Benefits</p> <ul style="list-style-type: none"> • Delivers maximum performance so that the synchronization server is not the bottleneck.
<p>Architecture</p> <ul style="list-style-type: none"> • Multiple synchronization servers provide built-in redundancy. • Provides failover between synchronization servers, source servers, and destination servers. 	<p>Benefits</p> <ul style="list-style-type: none"> • Ensures zero downtime
MONITORING AND MANAGEMENT	
<p>Server Configuration</p> <ul style="list-style-type: none"> • Supports command-line tools and a web console. • Records every server configuration change, user who made the change, and how to reverse it. • Allows the configuration of multiple synchronization servers at once. 	<p>Benefits</p> <ul style="list-style-type: none"> • Ensures a simple way to access the configuration. • Ensures security and accountability with the ability to playback or undo configurations.
<p>Logging</p> <ul style="list-style-type: none"> • Employs an extensive set of logging capabilities for sync auditing, error, and debug, including admin alerts and customized logging support. • Provides a detailed audit of every change that was synchronized. • Can separately log only those changes that failed and require administrative attention. 	<p>Benefits</p> <ul style="list-style-type: none"> • Ensures fast and effective troubleshooting to meet the needs of any data center.
<p>Administration Alerts</p> <ul style="list-style-type: none"> • Supports an alert framework to notify administrators about significant events via JMX™, SNMP or SMTP. • Provides an API to define custom alert handlers. 	<p>Benefits</p> <ul style="list-style-type: none"> • Ensures fast and effective management if a problem occurs. • Custom alerts can be tailored to specific production environments for more efficient management.
<p>Real-Time Monitoring</p> <ul style="list-style-type: none"> • Exposes all monitor information via JMX, LDAP, or the console. 	<p>Benefits</p> <ul style="list-style-type: none"> • Provides detailed information about all of the changes that have been synchronized, including how many changes were detected, how many were applied, and how many are outstanding.
LOW TOTAL COST OF OWNERSHIP	
<p>Platform Support</p> <ul style="list-style-type: none"> • Runs on the Java 6 Virtual Machine (VM). • Dataless architecture allows you to synchronize data without an investment in expensive hardware. 	<p>Benefits</p> <ul style="list-style-type: none"> • Enables maximum utilization of existing hardware. • Server can be moved between system architectures with no modification. Reduces equipment requirements, lowering total cost of ownership.

Bidirectional Synchronization

The UnboundID Synchronization Server performs point-to-point synchronization from a source endpoint to a destination endpoint. An endpoint is either a source or destination data store supported by the synchronization server. Please see the Platform Support section for a full listing of supported endpoints. The UnboundID Synchronization Server synchronizes changes directly from the data sources in the background without storing any data from the endpoints themselves. This dataless approach allows applications to update their data sources directly and reduces hardware and administration costs.

Data is synchronized from a source to a destination endpoint without any data being stored locally by the synchronization server.

You can synchronize data in one direction or bidirectionally. For example, in a migration phase from Sun Directory Server to UnboundID Directory Server, you might want to sync in one direction for testing purposes. Bidirectional sync allows parallel, active installations. You can also make sensitive attributes anonymous, such as SSN and phone numbers, making it possible to use production data in test systems.



By definition, the source server is an authoritative endpoint for attributes being synchronized, because it generates all of the changes. You could set up a bidirectional synchronization configuration, so that both sets of endpoints (i.e., the source and the destination) are authoritative for the same set of attributes, or for different sets of data.

Architecture

Similar to the UnboundID Directory Server, the UnboundID Synchronization Server provides several administrative protocols:

- **LDAP:** Used for monitoring, configuration, server state, and tasks.
- **JMX:** Used for monitoring and alerts.
- **SMTP:** Used for email alert notifications.
- **SNMP:** Used for monitoring SNMP notifications.

The synchronization engine detects any changes between directory server topologies over LDAP using the UnboundID LDAP changelog, the Sun Directory Server Retro Changelog, or the Active Directory DirSync control. The UnboundID Synchronization Server provides full control over the synchronization process by determining which entries are synchronized, how they are correlated to the entries at the destination endpoint, and how they are transformed into the destination schema.

Immediate Failover

The UnboundID Synchronization Server provides immediate failover capabilities. The synchronization server is connected to a single source and a single destination server at a time. If any of these go down, then the synchronization server can connect to the other source or destination server. For example, if there is a hardware problem or a machine is down for maintenance, the failover server instance can assume its active role and start synchronizing wherever it left off. The servers can pick up the first missed change and apply it quickly.

Modes of Operation

The UnboundID Synchronization Server runs as a standalone Java process with two complementary synchronization modes of operation: real-time and scheduled bulk synchronization. The real-time and bulk modes can be used together. For example, a bulk resynchronization operation guarantees that all entries are in-sync at initialization, and afterwards the UnboundID Synchronization Server can continue to synchronize new changes in real time.

Real-time Mode

In real-time mode, the synchronization server polls the source server for changes and synchronizes the destination entries immediately. The UnboundID Synchronization Server uses the mechanism that is most efficient for each platform. For UnboundID Directory Server topologies, the UnboundID Synchronization Server uses the server's LDAP Change Log for change detection. For DSEE or Sun Directory Server topologies, the UnboundID Synchronization Server uses the server's Retro Change Log, which provides a detailed summary of each change applied to the directory. For Active Directory, the UnboundID Synchronization Server uses the DirSync control.

Once the UnboundID Synchronization Server determines that a detected change should not be excluded from synchronization, it fetches the full entry from the source. Then, it finds the corresponding entry in the destination endpoint using flexible correlation rules and applies the minimal set of changes to bring the attributes that were modified into sync. The reason the server fetches and compares the full entries is to make sure it does not synchronize any stale data from the changelog.

Bulk Synchronization Mode

In bulk synchronization mode, also known as resynchronization, the synchronization server streams all entries or those entries matching certain criteria from the source servers and updates the corresponding destination entries. The server does a bulk comparison of source entries with destination entries and, depending on the configuration, updates or reports on what is out of sync. You can operate in this mode at the same time as real-time synchronization is running. It accounts for synchronization delays by rechecking out-of-sync entries before updating them, which takes into account short transient delays for entries that are modified by real-time synchronization. Resynchronization gives you lots of control over what entries and attributes get synchronized.

Resynchronization is designed to accomplish the following tasks:

- Initially populate a target directory over LDAP.
- Verify that the two endpoints are in sync.
- Perform scheduled (such as nightly) synchronization in place of real-time synchronization.
- Recover from a failure by resynchronizing entries that were modified since the last backup was taken.
- Preview what is out of sync to validate that the server is behaving as expected. You can use the **--dry-run** option on active systems to validate that sync is operating properly, without updating any entries.

You have fine control over what is included in the bulk mode resynchronization. For example, you can control the following:

- Include or exclude source and destination attributes.
- Apply an LDAP filter to only sync entries created since the last time you ran the tool (`createTimeStamp=>=200503311200-0600`).
- Synchronize only creates or only modifications.
- Change the logging verbosity.
- Set a limit on how fast the resynchronization runs (only 2000/second) to limit impact on endpoint directory servers.

Realistic Test Environments

In an effort to secure sensitive user data, many companies have their own test environments that use synthetic data that does not closely match actual production data. This discrepancy could introduce problems when moving application from the test environment to the production environment. The UnboundID Synchronization Server is capable of fully synchronizing test or stage servers with production servers while also obfuscating sensitive customer information. The synchronization server can sync in real-time or on a nightly basis with little additional performance load on the production servers.

Data obfuscation can be easily configured using direct, constructed (destination attributes are constructed from source), or DN mapping techniques.

Data Transformations

When the UnboundID Synchronization Server synchronizes entries between a source and destination server, it can be configured to transparently change the contents of these entries, so that neither server needs to be aware of the transformation. Available transformations include:

- **Attribute mapping:** The synchronization server can transparently rename any attributes in the entry. This mapping makes it possible to synchronize information stored in one attribute in one directory server topology to an attribute with a different name in another directory server topology, or to construct an attribute using portions of source attribute values.
- **DN mapping:** The synchronization server can transparently alter any DNs referenced in the entries. This mapping makes it possible to synchronize data from a topology that uses one DIT structure to a system that uses a different DIT structure.
- **Server SDK Extension Points:** When synchronizing data with relational databases or web services, extension points are available to act as an adapter between the UnboundID Synchronization Server and your database or web service environment. This allows you to define the exact mapping semantics between the data schemas (LDAP, relational, REST, etc...) utilized in the various environments.

Data transformations alter the contents of entries that are synchronized between source and destination directory server topologies.

Advanced Replication

The UnboundID Synchronization Server provides advanced features that extend the replication capability of directory servers. Traditionally, replication creates exact replicas of servers, including the same DIT structure, entries and attributes. However, in many cases, replica servers need to store a subset of entries, a subset of attributes, or even extra attributes, as compared to the primary master replicas. The UnboundID Synchronization Server provides this flexibility.

The server provides the following types of partial replication that you can combine to meet the needs of your topology:

- **Fractional replication:** Instead of synchronizing the all attributes on an entry, only a subset of the attributes are synchronized.
- **Local data replication:** You can choose to exclude application specific data that is not replicated back to the authoritative server, such as large XML blobs that are needed by a single application.
- **Filtered replication:** Only certain types of entries are replicated as determined by matching an LDAP filter. For example, all entries except subscriber entries may be included.
- **Subtree replication:** Only parts of the DIT are replicated, depending on the inclusion or exclusion of specific base DNs.

This section describes the replication methods in detail and illustrates them using a sample use case. In this scenario, a large telecom company is managing data replication between three divisions: billing, web, and network. The directory server for each division contains the same subscriber information. However, each division has a unique directory structure. The billing division is the authority for subscriber information.

Each division replicates between its own local servers using replication agreements that accommodate the division's unique DIT and schema. The synchronization server replicates data across division boundaries using the different replication methods to address the needs of the telecom.

A directory deployment often requires directory servers to contain less (or sometimes even more) data than the primary master directory servers. The synchronization server solves this use case, improving the overall performance of the directory service and reducing hardware costs, because all servers do not need full copies of the data.

Fractional Replication

Fractional replication is a form of partial replication that allows a subset of attributes to be replicated to another replica. This feature is often used to reduce replication bandwidth by creating directory servers that contain only the data they need. For example, if a replica only performs user authentications, then synchronization can be configured to only propagate the **uid** and **userpassword** attributes.

Fractional replication reduces the database size at the replica and the network traffic needed to keep this server in sync with the masters. Furthermore, changes due to password policy attributes, such as account lockouts, can be synchronized back to the main masters. The UnboundID Synchronization Server supports fractional replication to any type of server. Using our telecom use case example, a user can change their email address in a variety of ways:

- Calling the billing department
- Going into a retail store
- Logging in to the web site
- Using IVR on their telephone

No matter where the email address is changed, it needs to be reflected everywhere. If a subscriber changes their email address by calling the billing department, the synchronization server uses fractional replication to replicate only the updated email attribute of the subscriber's entry across the servers in the other departments.

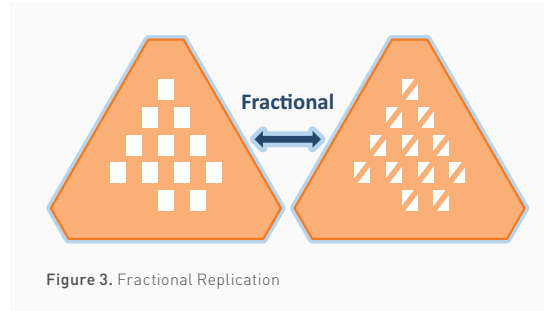


Figure 3. Fractional Replication

Local Data Replication

With fractional replication, you need certain application-specific replicas to contain less data than the primary master servers. With local data replication, replicas need to have more data than the primary master servers. Some applications need to store large amounts of data in a user entry, such as an XML blob of preference information. Though this extra data is required only by one application, the data can impact the server performance for all applications. The synchronization server can keep this data isolated to only a few servers dedicated to this application.

In the large telecom company example, we can imagine that the web department uses a portal server and web applications that are not used by the other departments. The user preference information stored on the user entries in the web department's directory server is not replicated back to the other departments in the telecom's deployment. Instead, this information is replicated only between the servers in the web department.

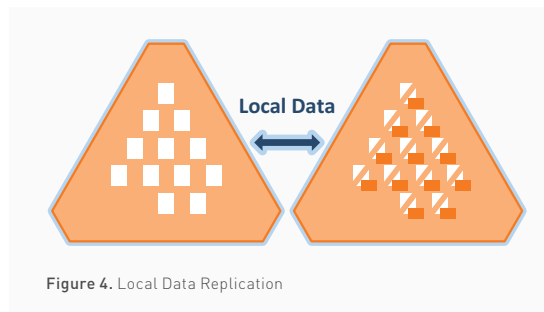


Figure 4. Local Data Replication

Filtered Replication

Filtered replication allows the replication of certain types of entries that match an LDAP filter. For example, applications that create an application-specific entry can be restricted to only those directories used by the application and not to every replica in the topology.

In the large telecom example, the company has an extranet directory server and a corporate directory server. The extranet directory server is used by sales staff that are on the road to authenticate and use their email over the web.

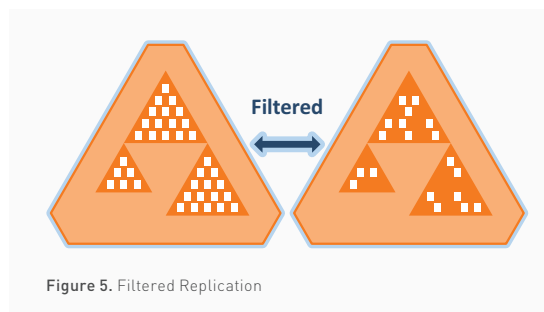


Figure 5. Filtered Replication

To access their email remotely, an employee must have a **web-access-enabled=true** flag set on their entry. Using filtered replication, the synchronization server can look for entries that have this flag set to true and replicate their changes back to the other corporate directories. For example, if an employee changes their password in the extranet directory, this change will be replicated back to the master corporate directory.

Subtree Replication

In subtree replication, a replica contains only selected entries as determined by a directory branch or LDAP filter. This feature allows directory server instances to replicate only specific subtrees of a DIT, which are determined by inclusion or exclusion filtering on the base DN's.

For example, the large telecom company acquires a media company. The telecom contains a subtree of data in its directory server for the media company and the media company itself has an on-premise LDAP directory server. Using a subtree replication agreement, data can be replicated between the media company's directory server and the main telecom directory server.

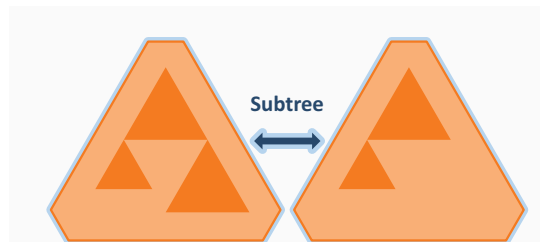


Figure 6. Subtree Replication

Advanced Replication Combinations

The UnboundID Synchronization Server supports various combinations of fractional, subtree, filtered and local data replication for those applications that require it.

For example, imagine that the telecom operates in several countries in the European Union. Europe itself restricts the disclosure of anything considered personal data. Rules inside individual countries can further restrict the type of data that can be transferred between country boundaries.

While this data must remain in the individual country's directory server, the corporate directory still needs to contain as much information as possible. Using fractional replication, only the parts of the entry that can legally cross country borders would be replicated back to the main corporate directory server.

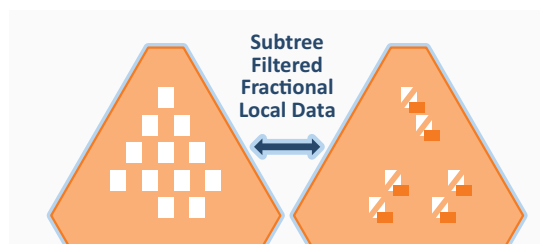


Figure 7. Advanced replication using a combination of methods

Further, imagine that the main directory for the telecom company, **dc=corp,dc=com** contains subdirectories for each of the European directories, such as **ou=France, ou=Germany**, and **ou=Italy**. Locally, each country has its own unique directory data stored in its own DIT structure, such as **dc=corp-fr, dc=com** for France. Using subtree replication in combination with fractional replication, the synchronization server can replicate changes between a country's subtree in the master directory and each country's local directory, while adhering to local laws about the transfer of personal information.

Synchronizing with Active Directory

The UnboundID Synchronization Server supports the synchronization of add, modify, and delete operations of entries within Active Directory. The Synchronization Server uses Active Directory's DirSync control.

If real-time password synchronization is needed, the synchronization server also requires that a dedicated component, the UnboundID Password Sync Agent (PSA), be installed on all Active Directory domain controllers. The agent receives password changes from the Local Security Authority (LSA) and immediately hashes them with a 160-bit salted secure hash. The agent then sends the hashes to each UnboundID Synchronization Server instance in the topology over a secure LDAPS connection. If a synchronization server instance is down, the agent caches the change and retries synchronization until at least one server has received the updates.

The agent is highly optimized with a small memory footprint. It securely handles sensitive data and uses a small, native DLL on the domain controller, which requires a single restart (Microsoft requirement). Subsequent updates to the DLL do not require a restart.

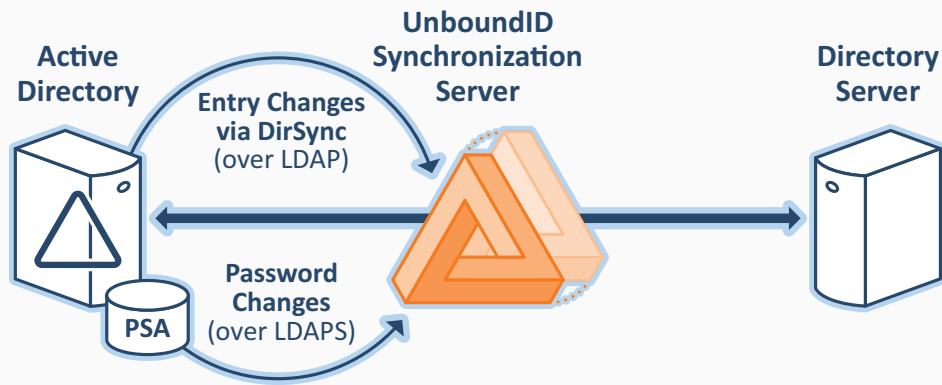


Figure 8. Active Directory Sync

Synchronizing with Databases

The Synchronization Server is designed for high-scale, parallelized, point-to-point data synchronization between a directory server and a relational database management system (RDBMS) system via a Type 4 JDBC driver. It will support the synchronization of any type of data stored in a RDBMS. The Synchronization Server provides multiple configuration options, such as, advanced filtering (fractional and subtree), attribute and DN mappings, transformations, correlations, and configurable logging features for seamless one-way or bidirectional synchronization.

To support synchronizing changes out of a database, the database must be configured with a change tracking mechanism. We recommend a general approach involving triggers (one trigger per table) to record all changes to a changelog table. The database changelog table should record the type of change (INSERT, UPDATE, DELETE) that occurred, the specific table name, the unique identifier for the row that was changed, the database entry type, the changed columns, the modifier's name, and the timestamp of the change.

The Synchronization Server delegates the physical interaction with the database to a scripted layer, which has full control of the SQL queries. The scripted layer provides flexibility in how you define the mapping semantics between your LDAP environment and your relational database environment. The connection management, pooling, retry logic, and other boilerplate code are all handled internally by the Synchronization Server.

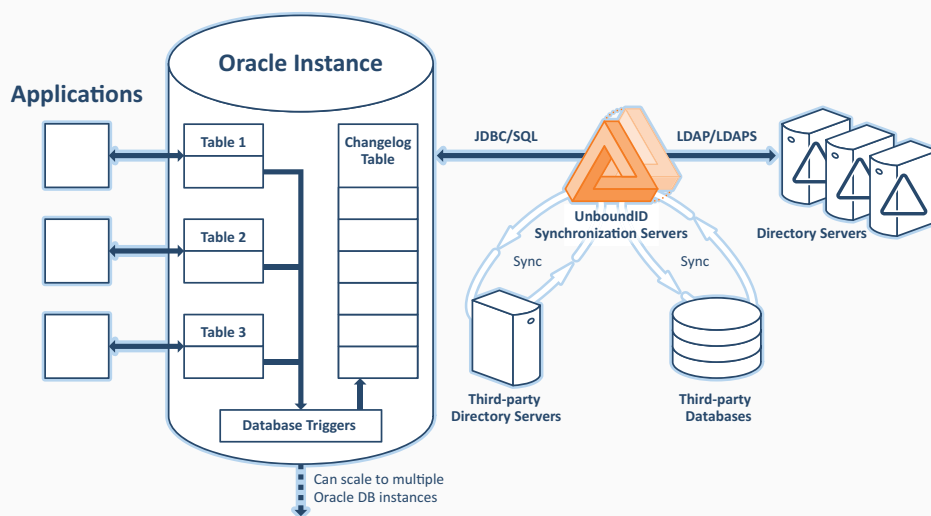


Figure 10. Database Sync

Notification Mode

The UnboundID Synchronization Server supports two general modes of operation: standard and notification. Standard Mode is the default mode used to synchronize data changes between two endpoints. In both modes, the Sync Server polls the changelog of the source endpoint for all create, modify, and delete operations on entries. In standard mode, it fetches the full entries from both the source and destination endpoints and compares them to produce the minimal set of changes needed to bring the destination server in sync with the source server. The Sync Server completes the process by updating the destination endpoint with the necessary changes.

Because the goal with Notification Mode is not to create an exact copy of the data at the source and destination, the Sync Server skips the fetch and compare phases of processing. With Notification Mode, it accesses state information on the changelog to reconstruct the before-and-after values of any modified attribute (for example, for MODIFY operation types). It then passes in the change information to a custom server extension based on the UnboundID Server SDK that is responsible for formatting and sending the notification message. Third-party libraries can be employed to customize the notification messages to an output format required by the client application or service. For example, the server extension can use a third-party XML parsing library to convert the change notifications to a SOAP XML format.

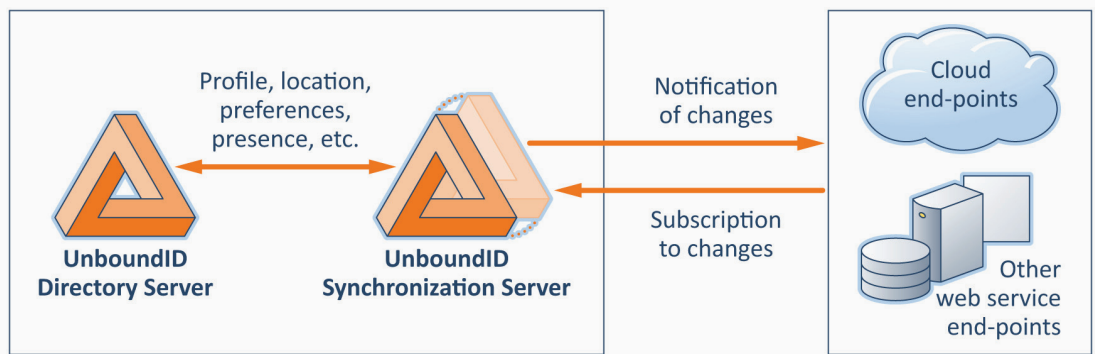


Figure 9. Notification Mode

High Performance and Availability

The UnboundID Synchronization Server provides high performance throughput and low latency processing that exceeds most companies' requirements. The synchronization server is very available and robust. You can deploy multiple synchronization servers for failover purposes.

Changes are processed in parallel whenever possible. A thread pool increases the throughput from the sync source to destination, even if the network has high latency. Table 1 highlights typical performance speeds when synchronizing between UnboundID Directory Server instances.

The sample performance numbers were tested with a 5M entry data set running on a Sun Fire x2270 machine.

Operation	Ops/sec	Total time for 100k
Bulk resync (provision empty destination)	600	167 seconds
Bulk resync (verify directories are in-sync)	6500	15 seconds
Real-time sync (synchronize modify ops)	3000	33 seconds

Table 1. Performance Numbers for Bulk and Real-time Sync Modes

The UnboundID Synchronization Server also contains no single point of failure, either for detecting changes or applying changes. The synchronization server fails over to multiple source and destination server instances, picking up with the first missed change quickly. The synchronization server instances themselves are redundant. Each instance has a priority, and the highest priority live instance synchronizes changes.

The standby instances periodically poll the live instance to update their persistent state. This state contains the minimum amount of information needed to begin synchronizing where the primary server left off. Logically, it is the last processed change number for the source server. In the case of a network partition, multiple synchronization servers can synchronize simultaneously without causing problems because they each verify the full entry before making any changes.

Monitoring and Management

The UnboundID Synchronization Server is streamlined for multiple server installations. Installing the second and subsequent servers is simple, because you merely point the new synchronization server at an existing server and the configuration of the new server is automatically set to match the first server, and kept in sync thereafter. Once you have installed several synchronization servers, you can manage them as a single server using the administration tools, which include command-line tools and the web-based Synchronization Server Management Console. The synchronization server also includes a notification system to help administrators monitor significant events during synchronization.

Server Configuration

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

The Synchronization Server Management Console

The Synchronization Server Management Console is a web application that allows administrators to use a web browser to easily view and manage their topology.

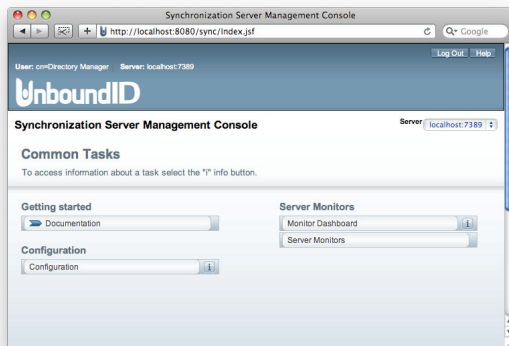


Figure 11. Synchronization Server Management Console

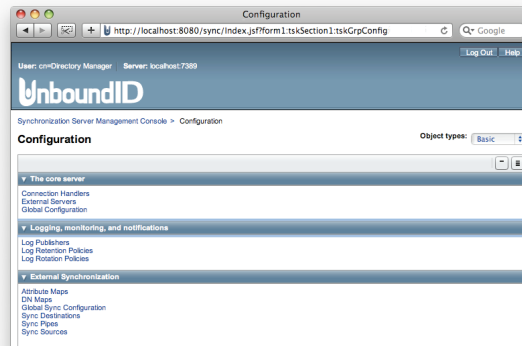


Figure 12. Synchronization Server Management Console Configuration Screen

The dsconfig Command-Line Tool

The **dsconfig** tool is a command-line utility with several modes of operation: a menu-driven interactive mode, 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 synchronization servers in your topology using an existing configuration or to move a configuration from a test environment into production.

```
>>>> UnboundID Synchronization Server configuration console main
menu

What do you want to configure?

1) Attribute Map           8) Log Publisher
2) Attribute Mapping       9) Log Retention Policy
3) Connection Handler     10) Log Rotation Policy
4) DN Map                 11) Sync Class
5) External Server        12) Sync Destination
6) Global Configuration   13) Sync Pipe
7) Global Sync Configuration 14) Sync Source

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

Enter choice:
```

Figure 13. The dsconfig Command-Line Utility

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.

Configuration Audit Log

The UnboundID Synchronization 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 performed the change, and a command that may be used to revert the change, if desired. In addition, the synchronization 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 reapply the change.

Logging

The UnboundID Synchronization Server offers the same rich logging subsystem as the UnboundID Directory Server, with additional logs to record information about synchronized entries. The sync log provides information about synchronization actions that occur during normal processing, such as all of the changes applied, detected, or failed, dropped operations that were not synchronized, and changes dropped because they are out of scope. A resync log provides information about synchronized or missing entries when performing bulk synchronization.

You can use multiple loggers and configure what goes into each log. For example, you can create one log that includes information about every change and a separate more detailed log for any changes that failed to be synchronized. Separate logs make it easy to notice and then take action when an operation fails to synchronize. Administrative alerts can be configured for failed changes and sent out as email, SMTP traps, or JMX notifications.

A message is generated for each detected change, dropped change, applied change, and failed change. You can configure which messages should be included in the log and the level of detail to include. During development and test, you may have it at a debugging level to track operations, and then use less detail once the server is in production.

Administrative Alerts

The UnboundID Synchronization 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.

The administrative alert framework of the UnboundID Synchronization Server enables administrators to control what types of events generate an alert and how the alert is delivered (email, SMTP trap, or JMX notification).

Alternatively, you can configure the synchronization 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 administrator can select the action to take 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 synchronization 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.
- **Synchronized resources unavailable:** Sends an alert whenever the resources being synchronized are unavailable.

Real-Time Monitoring

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

Sync Stats	Provides information about the synchronization path between a source and destination topology, including details about the number of changes detected, how many have not been synchronized, and how many existed previously.
Sync Topology	Provides information about redundant synchronization servers, including which server is currently working and which server or servers are on standby.

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

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 Synchronization Server.

Low Total Cost of Ownership

The UnboundID Synchronization Server's ability to synchronize data between multiple directory environments eases the complicated merging of new and legacy applications, lowering the total cost of ownership (TCO) over time. The server itself requires modest hardware and has a minimal administrative burden. The dataless architecture allows you to synchronize data without an investment in expensive hardware to support massive datasets.

Dataless Synchronization

The UnboundID Synchronization Server uses a dataless approach, in which the synchronization server synchronizes changes directly from the data sources in the background, so that applications can continue to update the data sources directly. As a result, the Synchronization Server does not store any data from the endpoints itself, thereby reducing hardware and administrative costs. This approach eliminates the need for backups and large disk requirements. The log files, administrator entries, configuration, and synchronization state information are stored as flat LDIF files within the system. No additional database is required. The synchronization servers periodically exchange information about what changes have been synchronized, making it easy to failover between the main synchronization server and its redundant instances.

The UnboundID Synchronization Server uses a dataless approach and never stores any directory data locally.

Platform Support

The UnboundID Synchronization Server provides a choice of real-time or scheduled synchronization across independent directory server topologies, enabling you to operate multiple architectures in parallel. The source and destination endpoints can use the following vendor directory servers:

- UnboundID Directory Server
- Sun Directory Server (5.x, 6.x, 7.x)
- Microsoft Active Directory

In addition to the directory servers listed above, the UnboundID Synchronization Server also supports synchronizing data with an endpoint consisting of a relational database management system (RDBMS). Official support is provided for synchronizing with the following relational databases:

- Oracle Database 10g and 11g
- Microsoft SQL Server 2005 and 2008

The architecture, however, does not make any assumptions about the type of database or schema being managed; any database with a Type 4 JDBC driver can be synchronized. Lastly, the UnboundID Synchronization Server, in conjunction with the Server SDK, provides a set of extension points that make it possible to synchronize with other endpoints, such as web services

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.

The UnboundID staff members are experts in the Identity Management field and have a proven track record of many successful deployments.



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