Reference

ForgeRock Directory Services 5

Mark Craig

ForgeRock AS
201 Mission St., Suite 2900
San Francisco, CA 94105, USA
+1 415-599-1100 (US)
www.forgerock.com

Abstract

Reference for ForgeRock® Directory Services, including bundled tools.

Table of Contents

Preface

ForgeRock Identity Platform™ is the only offering for access management, identity management, user-managed access, directory services, and an identity gateway, designed and built as a single, unified platform.

The platform includes the following components that extend what is available in open source projects to provide fully featured, enterprise-ready software:

- ForgeRock Access Management (AM)
- ForgeRock Identity Management (IDM)
- ForgeRock Directory Services (DS)
- ForgeRock Identity Gateway (IG)

The ForgeRock Common REST API works across the platform to provide common ways to access web resources and collections of resources.

This reference covers ForgeRock Directory Services configuration, tools, and other topics such as supported languages and standards. For the **dsconfig** command, also see the *Server Configuration Reference*. For API specifications suitable for ForgeRock Directory Services developers, see the appropriate Javadoc.

Tools Reference

Find the bundled tools where you installed the server, as indicated in Section 2.2, "Server-Side Command-Line Tools" in the *Administration Guide*.

Table of Contents

Name

addrate — measure add and delete throughput and response time

Synopsis

`addrate` template-file-path

Description

This utility can be used to measure add and optionally delete throughput and response time of a directory server using user-defined entries. The {template-file-path} argument identifies a template file that has the same form as a template file for the makeldif command.

Examples:

This example adds entries and randomly deletes them while the number of entries added is greater than 10,000:

addrate -p 1389 -f -c 10 -C random -s 10000 addrate.template

This example adds entries and starts to delete them in the same order if their age is greater than a certain time:

addrate -p 1389 -f -c 10 -C fifo -a 2 addrate.template

For details about the template file, see makeldif.template.

When you do not use the `-f` option to keep connections open and rebind on the connections, the tool can exhaust its available ports, causing the tool to crash. You can work around this problem on test systems by changing TCP settings on the system.

For example, on Linux systems, set the following parameters in the `/etc/sysctl.conf` file:

```
net.ipv4.tcp_fin_timeout = 30
net.ipv4.tcp_tw_recycle = 1
net.ipv4.tcp_tw_reuse = 1
```

The parameter `net.ipv4.tcp_fin_timeout` sets the length of time in seconds to wait for a final FIN packet before forcing a close of the socket. The default is 60 (seconds).

The parameter `net.ipv4.tcp_tw_recycle` enables fast recycling of TIME_WAIT sockets. The default is 0 (false). Enabling this can cause Network Address Translation (NAT) issues.

The parameter `net.ipv4.tcp_tw_reuse` enables reuse of TIME_WAIT sockets for new connections. The default is 0 (false).

These settings are recommended only for testing, and *not for production systems*.

After making the changes to `/etc/sysctl.conf`, reload the configuration with the **sysctl** command:

```
# sysctl -p
```

Options

The **addrate** command takes the following options:

Command options:

-a | --deleteAgeThreshold {seconds}

Specifies the age at which added entries will become candidates for deletion.

-B | --warmUpDuration {warmUpDuration}

Warm up duration in seconds.

Default: 0

-c | --numConnections {numConnections}

Number of connections.

Default: 1

-C | --deleteMode {fifo | random | off}

The algorithm used for selecting entries to be deleted which must be one of "fifo", "random", or "off".

Default: FIFO

--connectTimeout {timeout}

Maximum length of time (in milliseconds) that can be taken to establish a connection. Use '0' to specify no time out.

Default: 30000

-d | --maxDuration {maxDuration}

Maximum duration in seconds, 0 for unlimited.

Default: 0

-e | --percentile {percentile}

Calculate max response time for a percentile of operations.

-f | --keepConnectionsOpen

Keep connections open.

Default: false

-F | --noRebind

Keep connections open and do not rebind.

Default: false

-g | --constant {name=value}

A constant that overrides the value set in the template file.

-i | --statInterval {statInterval}

Display results each specified number of seconds.

Default: 5

-m | --maxIterations {maxIterations}

Max iterations, 0 for unlimited.

Default: 0

-M | --targetThroughput {targetThroughput}

Target average throughput to achieve.

Default: 0

-n | --noPurge

Disable the purge phase when the tool stops.

Default: false

-r | --resourcePath {path}

Path to look for template resources (e.g. data files).

The utility looks for resources in the following locations in this order:

1. The current directory where the command is run.
2. The resource path directory.
3. The built-in files.

-R | --randomSeed {seed}

The seed to use for initializing the random number generator.

Default: 0

-s | --deleteSizeThreshold {count}

Specifies the number of entries to be added before deletion begins.

Default: 10000

-S | --scriptFriendly

Use script-friendly mode.

Default: false

-t | --numThreads {numThreads}

Number of worker threads per connection.

Default: 1

LDAP connection options:

-D | --bindDn {bindDN}

DN to use to bind to the server.

Default:

-E | --reportAuthzId

Use the authorization identity control.

Default: false

-h | --hostname {host}

The fully-qualified directory server host name that will be used when generating self-signed certificates for LDAP SSL/StartTLS, the administration connector, and replication.

Default: localhost.localdomain

-j | --bindPasswordFile {bindPasswordFile}

Bind password file.

-K | --keyStorePath {keyStorePath}

Certificate key store path.

-N | --certNickname {nickname}

Nickname of the certificate that the server should use when accepting SSL-based connections or performing StartTLS negotiation.

-o | --saslOption {name=value}

SASL bind options.

-p | --port {port}

Directory server port number.

Default: 389

-P | --trustStorePath {trustStorePath}

Certificate trust store path.

-q | --useStartTls

Use StartTLS to secure communication with the server.

Default: false

-T | --trustStorePassword {trustStorePassword}

Certificate trust store PIN.

-u | --keyStorePasswordFile {keyStorePasswordFile}

Certificate key store PIN file. A PIN is required when you specify to use an existing certificate as server certificate.

-U | --trustStorePasswordFile {path}

Certificate trust store PIN file.

--usePasswordPolicyControl

Use the password policy request control.

Default: false

-w | --bindPassword {bindPassword}

Password to use to bind to the server. Use -w - to ensure that the command prompts for the password, rather than entering the password as a command argument.

-W | --keyStorePassword {keyStorePassword}

Certificate key store PIN. A PIN is required when you specify to use an existing certificate as server certificate.

-X | --trustAll

Trust all server SSL certificates.

Default: false

-Z | --useSsl

Use SSL for secure communication with the server.

Default: false

Utility input/output options:

--noPropertiesFile

No properties file will be used to get default command line argument values.

Default: false

--propertiesFilePath {propertiesFilePath}

Path to the file containing default property values used for command line arguments.

-v | --verbose

Use verbose mode.

Default: false

General options:

-V | --version

Display Directory Server version information.

Default: false

-H | --help

Display this usage information.

Default: false

Exit Codes

0

The command completed successfully.

80

The command could not complete due to an input/output error.

89

An error occurred while parsing the command-line arguments.

Examples

The following examples use this template file, `addrate.template`:

```
define suffix=dc=example,dc=com
define maildomain=example.com
```

```
branch: [suffix]

branch: ou=People,[suffix]
subordinateTemplate: person

template: person
rdnAttr: uid
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
givenName: <first>
sn: <last>
cn: {givenName} {sn}
initials: {givenName:1}<random:chars:ABCDEFGHIJKLMNOPQRSTUVWXYZ:1>{sn:1}
employeeNumber: <sequential:0>
uid: user.{employeeNumber}
mail: {uid}@[maildomain]
userPassword: password
telephoneNumber: <random:telephone>
homePhone: <random:telephone>
pager: <random:telephone>
mobile: <random:telephone>
street: <random:numeric:5> <file:streets> Street
l: <file:cities>
st: <file:states>
postalCode: <random:numeric:5>
postalAddress: {cn}${street}${l}, {st}  {postalCode}
description: This is the description for {cn}.
```

The following example adds entries, and then randomly deletes them when more than 10,000 entries have been added:

```
$ addrate -p 1389 -D "cn=Directory Manager" -w password
 \
-f -c 10 -C random -s 10000 addrate.template

---------------------------------------------------------------------
    Throughput                          Response Time
  (ops/second)                         (milliseconds)
recent  average  recent  average  99.9%  99.99%  99.999%  err/sec  Add
%
---------------------------------------------------------------------
466.7    467.2  20.738   20.738  187.00  201.00   201.00      0.0  100.00
588.9    528.1  17.015   18.661  166.00  201.00   201.00      0.0  100.00
1584.9    880.3   6.076   11.109  150.00  196.00   201.00      0.0  79.87
1577.8   1054.5   6.138    9.252  132.00  192.00   201.00      0.0  50.00
1853.0   1214.4   5.188    8.010  124.00  187.00   201.00      0.0  49.99
^CPurge phase...
2482.7   1426.2   3.790    6.783  114.00  187.00   201.00      0.0  9.77
```

The following example also adds entries, and then deletes them in the order they were added after they are 10 seconds old:

```
$ addrate -p 1389 -D "cn=Directory Manager" -w password
 \
-f -c 10 -C fifo -a 10 addrate.template

-------------------------------------------------------------------------
    Throughput                            Response Time
  (ops/second)                           (milliseconds)
recent  average  recent  average  99.9%  99.99%  99.999%  err/sec  Add
%
-------------------------------------------------------------------------
2065.6   2068.1   4.646    4.646  30.00   51.00    58.00      0.0  100.00
1479.7   1773.3   6.567    5.449  46.00   59.00    67.00      0.0  99.23
1443.4   1663.3   6.730    5.820  56.00  112.00   120.00      0.0  50.01
1462.6   1613.0   6.635    6.005  56.00  102.00   120.00      0.0  50.08
1452.2   1580.8   6.678    6.129  62.00  110.00   120.00      0.0  49.97
^CPurge phase...
1344.5   1541.4   7.170    6.280  69.00  176.00  1900.00      0.0  17.30
1703.3   1564.6   5.449    6.151  68.00  176.00  3000.00      0.0  0.00
```

Name

authrate — measure bind throughput and response time

Synopsis

`authrate` [filter format string] [attributes ...]

Description

This utility can be used to measure bind throughput and response time of a directory service using user-defined bind or search-then-bind operations.

Format strings may be used in the bind DN option as well as the authid and authzid SASL bind options. A search operation may be used to retrieve the bind DN by specifying the base DN and a filter. The retrieved entry DN will be appended as the last argument in the argument list when evaluating format strings.

Example (bind only):

authrate -p 1389 -D "uid=user.%d,ou=people,dc=example,dc=com" \

-w password -f -c 10 -g "rand(0,2000)"

Example (search then bind):

authrate -p 1389 -D '%2$s' -w password -f -c 10 \

-b "ou=people,dc=example,dc=com" -s one -g "rand(0,2000)" "(uid=user.%d)"

Before trying the example, import 2000 randomly generated users.

When you do not use the `-f` option to keep connections open and rebind on the connections, the tool can exhaust its available ports, causing the tool to crash. You can work around this problem on test systems by changing TCP settings on the system.

For example, on Linux systems, set the following parameters in the `/etc/sysctl.conf` file:

```
net.ipv4.tcp_fin_timeout = 30
net.ipv4.tcp_tw_recycle = 1
net.ipv4.tcp_tw_reuse = 1
```

The parameter `net.ipv4.tcp_fin_timeout` sets the length of time in seconds to wait for a final FIN packet before forcing a close of the socket. The default is 60 (seconds).

The parameter `net.ipv4.tcp_tw_recycle` enables fast recycling of TIME_WAIT sockets. The default is 0 (false). Enabling this can cause Network Address Translation (NAT) issues.

The parameter `net.ipv4.tcp_tw_reuse` enables reuse of TIME_WAIT sockets for new connections. The default is 0 (false).

These settings are recommended only for testing, and *not for production systems*.

After making the changes to `/etc/sysctl.conf`, reload the configuration with the **sysctl** command:

```
# sysctl -p
```

Options

The **authrate** command takes the following options:

Command options:

-a | --dereferencePolicy {dereferencePolicy}

Alias dereference policy ('never', 'always', 'search', or 'find').

Default: never

-b | --baseDn {baseDN}

Base DN format string.

-B | --warmUpDuration {warmUpDuration}

Warm up duration in seconds.

Default: 0

-c | --numConnections {numConnections}

Number of connections.

Default: 1

--connectTimeout {timeout}

Maximum length of time (in milliseconds) that can be taken to establish a connection. Use '0' to specify no time out.

Default: 30000

-d | --maxDuration {maxDuration}

Maximum duration in seconds, 0 for unlimited.

Default: 0

-e | --percentile {percentile}

Calculate max response time for a percentile of operations.

-f | --keepConnectionsOpen

Keep connections open.

Default: false

-g | --argument {generator function or static string}

Argument used to evaluate the Java style format strings in program parameters (ie. Base DN, Search Filter). The set of all arguments provided form the the argument list in order. Besides static string arguments, they can be generated per iteration with the following functions:

"inc({filename})" Consecutive, incremental line from file

"inc({min},{max})" Consecutive, incremental number

"rand({filename})" Random line from file

"rand({min},{max})" Random number

"randstr({length},_charSet_)" Random string of specified length and optionally from characters in the charSet string. A range of character can be specified with [start-end] charSet notation. If no charSet is specified, the default charSet of [A-Z][a-z][0-9] will be used.

-i | --statInterval {statInterval}

Display results each specified number of seconds.

Default: 5

-I | --invalidPassword {invalidPassword}

Percent of bind operations with simulated invalid password.

Default: 0

-m | --maxIterations {maxIterations}

Max iterations, 0 for unlimited.

Default: 0

-M | --targetThroughput {targetThroughput}

Target average throughput to achieve.

Default: 0

-s | --searchScope {searchScope}

Search scope ('base', 'one', 'sub', or 'subordinates'). Note: 'subordinates' is an LDAP extension that might not work with all LDAP servers.

Default: sub

-S | --scriptFriendly

Use script-friendly mode.

Default: false

LDAP connection options:

-D | --bindDn {bindDN}

DN to use to bind to the server.

Default:

-E | --reportAuthzId

Use the authorization identity control.

Default: false

-h | --hostname {host}

The fully-qualified directory server host name that will be used when generating self-signed certificates for LDAP SSL/StartTLS, the administration connector, and replication.

Default: localhost.localdomain

-j | --bindPasswordFile {bindPasswordFile}

Bind password file.

-K | --keyStorePath {keyStorePath}

Certificate key store path.

-N | --certNickname {nickname}

Nickname of the certificate that the server should use when accepting SSL-based connections or performing StartTLS negotiation.

-o | --saslOption {name=value}

SASL bind options.

-p | --port {port}

Directory server port number.

Default: 389

-P | --trustStorePath {trustStorePath}

Certificate trust store path.

-q | --useStartTls

Use StartTLS to secure communication with the server.

Default: false

-T | --trustStorePassword {trustStorePassword}

Certificate trust store PIN.

-u | --keyStorePasswordFile {keyStorePasswordFile}

Certificate key store PIN file. A PIN is required when you specify to use an existing certificate as server certificate.

-U | --trustStorePasswordFile {path}

Certificate trust store PIN file.

--usePasswordPolicyControl

Use the password policy request control.

Default: false

-w | --bindPassword {bindPassword}

Password to use to bind to the server. Use -w - to ensure that the command prompts for the password, rather than entering the password as a command argument.

-W | --keyStorePassword {keyStorePassword}

Certificate key store PIN. A PIN is required when you specify to use an existing certificate as server certificate.

-X | --trustAll

Trust all server SSL certificates.

Default: false

-Z | --useSsl

Use SSL for secure communication with the server.

Default: false

Utility input/output options:

--noPropertiesFile

No properties file will be used to get default command line argument values.

Default: false

--propertiesFilePath {propertiesFilePath}

Path to the file containing default property values used for command line arguments.

-v | --verbose

Use verbose mode.

Default: false

General options:

-V | --version

Display Directory Server version information.

Default: false

-H | --help

Display this usage information.

Default: false

Exit Codes

0

The command completed successfully.

89

An error occurred while parsing the command-line arguments.

Examples

The following example demonstrates measuring simple bind performance:

```
$ authrate -p 1389 -g "rand(names.txt)" \
 -D "uid=%s,ou=people,dc=example,dc=com" -w password -c 10 -f
--------------------------------------------------------------------
     Throughput                          Response Time
   (ops/second)                         (milliseconds)
recent  average  recent  average  99.9%  99.99%  99.999%  err/
sec
--------------------------------------------------------------------
9796.9   9816.6   1.029    1.029  12.413  161.451  161.835      0.0
14201.1  12028.1   0.704    0.835  9.508  161.456  167.573      0.0
14450.0  12835.9   0.692    0.782  8.989  161.835  174.518      0.0
12934.3  12860.6   0.773    0.779  9.253  161.339  174.426      0.0
14154.5  13121.0   0.706    0.764  9.025  161.451  177.101      0.0
^C
```

The `names.txt` contains all the user IDs for the sample suffix. All user password values have been set to `password` for this example.

Name

backendstat — gather OpenDJ backend debugging information

Synopsis

`backendstat` {subcommand} {options}

Description

This utility can be used to debug a backend.

Options

The **backendstat** command takes the following options:

`-V | --version`

Display Directory Server version information.

Default: false

`-H | --help`

Display this usage information.

Default: false

Subcommands

The **backendstat** command supports the following subcommands:

backendstat dump-index

Dump records from an index, decoding keys and values. Depending on index size, this subcommand can generate lots of output.

Options

The **backendstat dump-index** command takes the following options:

`-n | --backendId {backendName}`

The backend ID of the backend.

`-b | --baseDn {baseDN}`

The base DN within the backend.

-i | --indexName {indexName}

The name of the index.

-q | --statsOnly

Do not display backend data, just statistics.

Default: false

-K | --maxKeyValue {maxKeyValue}

Only show records with keys that should be ordered before the provided value using the comparator for the database container.

-k | --minKeyValue {minKeyValue}

Only show records with keys that should be ordered after the provided value using the comparator for the database container.

-X | --maxHexKeyValue {maxKeyValue}

Only show records with keys that should be ordered before the provided value using the comparator for the database container.

-x | --minHexKeyValue {minKeyValue}

Only show records with keys that should be ordered after the provided value using the comparator for the database container.

-S | --maxDataSize {maxDataSize}

Only show records whose data is no larger than the provided value.

Default: -1

-s | --minDataSize {minDataSize}

Only show records whose data is no smaller than the provided value.

Default: -1

-p | --skipDecode

Do not try to decode backend data to their appropriate types.

Default: false

backendstat dump-raw-db

Dump the raw records in hexadecimal format for a low-level database within the pluggable backend's storage engine. Depending on index size, this subcommand can generate lots of output.

Options

The **backendstat dump-raw-db** command takes the following options:

-n | --backendId {backendName}

The backend ID of the backend.

-d | --dbName {databaseName}

The raw database name.

-q | --statsOnly

Do not display backend data, just statistics.

Default: false

-K | --maxKeyValue {maxKeyValue}

Only show records with keys that should be ordered before the provided value using the comparator for the database container.

-k | --minKeyValue {minKeyValue}

Only show records with keys that should be ordered after the provided value using the comparator for the database container.

-X | --maxHexKeyValue {maxKeyValue}

Only show records with keys that should be ordered before the provided value using the comparator for the database container.

-x | --minHexKeyValue {minKeyValue}

Only show records with keys that should be ordered after the provided value using the comparator for the database container.

-S | --maxDataSize {maxDataSize}

Only show records whose data is no larger than the provided value.

Default: -1

-s | --minDataSize {minDataSize}

Only show records whose data is no smaller than the provided value.

Default: -1

-l | --singleLine

Write hexadecimal data on a single line instead of pretty format.

Default: false

backendstat list-backends

List the pluggable backends.

backendstat list-base-dns

List the base DNs in a backend.

Options

The **backendstat list-base-dns** command takes the following options:

-n | --backendId {backendName}

The backend ID of the backend.

backendstat list-indexes

List the indexes associated with a pluggable backend. This subcommand may take a long time to complete depending on the size of the backend.

Options

The **backendstat list-indexes** command takes the following options:

-n | --backendId {backendName}

The backend ID of the backend.

-b | --baseDn {baseDN}

The base DN within the backend.

backendstat list-raw-dbs

List the low-level databases within a pluggable backend's storage engine. This subcommand may take a long time to complete depending on the size of the backend.

Options

The **backendstat list-raw-dbs** command takes the following options:

-n | --backendId {backendName}

The backend ID of the backend.

-u | --useSiUnits

Uses SI Units for printing sizes.

Default: false

backendstat show-index-status

Shows the status of indexes for a backend base DN. This subcommand can take a long time to complete, as it reads all indexes for all backends.

When you run the 'list-index-status' command, the result is a table, followed by a "Total", which is the total number of indexes, followed by a list of indexes with "Over index-entry-limit keys" to show the values for which the number of entries exceeded the index entry limit. The table has the following columns.

Index Name

Name of the index, which takes the form *attr.type* for attribute indexes, and vlv.*name* for VLV indexes. Some indexes are for OpenDJ directory server's internal use.

Example: `givenName.caseIgnoreSubstringsMatch:6`

Tree Name

Name of the backend tree, which reflects how OpenDJ directory server organizes the data in the database.

Example: `/dc=example,dc=com/givenName.caseIgnoreSubstringsMatch:6`

Index Valid

This is `true` for valid indexes. If this is `false`, the index might be degraded. Verify the index, and rebuild the index if necessary.

Record Count

Number of indexed keys. Use the **backendstat dump-tree** command to see how many entry IDs correspond to each key.

Over Index Entry Limit

Number of keys for which there are too many values to maintain an index, based on the index entry limit. This is recorded as `-` for VLV indexes.

In other words, with the default index entry limit of 4000, if every user in your large directory has an email address ending in `@example.com`, and a substring index with default substring length of 6 is maintained for `mail`, then OpenDJ directory server does not maintain indexes for keys corresponding to substrings in `@example.com`.

As a result, an LDAP search with the filter `"(mail=*@example.com)"` becomes an unindexed search even though a substring index exists for the mail attribute. By default OpenDJ directory server does not allow unindexed searches except by privileged users. This is usually exactly the behavior you want in order to prevent client applications from sending searches that return every user in the directory for example. Clients should refine their search filters instead.

95%, 90%, 85%

Number of keys for which the number of values is approaching the index entry limit, having at least the specified percentage. This is a measure of how full the entry ID lists are.

Options

The **backendstat show-index-status** command takes the following options:

-n | --backendId {backendName}

The backend ID of the backend.

-b | --baseDn {baseDN}

The base DN within the backend.

Exit Codes

0

The command completed successfully.

> 0

An error occurred.

Examples

The following example displays index information.

```
$ bin/backendstat dump-index  -n userRoot -b dc=example,dc=com -i id2childrencount

    Key (len 2): 1#52
    Value (len 8): 1
    Key (len 2): 2#52
    Value (len 8): 500000
    Key (len 9): Total Children Count
    Value (len 8): 500001

    Total Records: 3
    Total / Average Key Size: 13 bytes / 4 bytes
    Total / Average Data Size: 24 bytes / 8 bytes
```

Name

backup — back up directory data

Synopsis

```
backup
```

Description

This utility can be used to back up one or more Directory Server backends.

Options

The **backup** command takes the following options:

Command options:

-a | --backUpAll

Back up all backends in the server.

Default: false

-A | --hash

Generate a hash of the backup contents.

Default: false

-B | --incrementalBaseId {backupID}

Backup ID of the source archive for an incremental backup.

-c | --compress

Compress the backup contents.

Default: false

-d | --backupDirectory {backupDir}

Path to the target directory for the backup file(s).

-i | --incremental

Perform an incremental backup rather than a full backup.

Default: false

-I | --backupId {backupID}

Use the provided identifier for the backup.

-n | --backendId {backendName}

Backend ID for the backend to archive.

--offline

Indicates that the command must be run in offline mode.

Default: false

-s | --signHash

Sign the hash of the backup contents.

Default: false

-y | --encrypt

Encrypt the backup contents.

Default: false

Task Backend Connection Options

--connectTimeout {timeout}

Maximum length of time (in milliseconds) that can be taken to establish a connection. Use '0' to specify no time out.

Default: 30000

-D | --bindDn {bindDN}

DN to use to bind to the server.

Default: cn=Directory Manager

-h | --hostname {host}

The fully-qualified directory server host name that will be used when generating self-signed certificates for LDAP SSL/StartTLS, the administration connector, and replication.

Default: localhost.localdomain

-j | --bindPasswordFile {bindPasswordFile}

Bind password file.

-K | --keyStorePath {keyStorePath}

Certificate key store path.

-N | --certNickname {nickname}

Nickname of the certificate that the server should use when accepting SSL-based connections or performing StartTLS negotiation.

-o | --saslOption {name=value}

SASL bind options.

-p | --port {port}

Directory server administration port number.

Default: 4444

-P | --trustStorePath {trustStorePath}

Certificate trust store path.

-T | --trustStorePassword {trustStorePassword}

Certificate trust store PIN.

-u | --keyStorePasswordFile {keyStorePasswordFile}

Certificate key store PIN file. A PIN is required when you specify to use an existing certificate as server certificate.

-U | --trustStorePasswordFile {path}

Certificate trust store PIN file.

-w | --bindPassword {bindPassword}

Password to use to bind to the server. Use -w - to ensure that the command prompts for the password, rather than entering the password as a command argument.

-W | --keyStorePassword {keyStorePassword}

Certificate key store PIN. A PIN is required when you specify to use an existing certificate as server certificate.

-X | --trustAll

Trust all server SSL certificates.

Default: false

Task Scheduling Options

--completionNotify {emailAddress}

Email address of a recipient to be notified when the task completes. This option may be specified more than once.

--dependency {taskID}

ID of a task upon which this task depends. A task will not start execution until all its dependencies have completed execution.

--errorNotify {emailAddress}

Email address of a recipient to be notified if an error occurs when this task executes. This option may be specified more than once.

--failedDependencyAction {action}

Action this task will take should one if its dependent tasks fail. The value must be one of PROCESS,CANCEL,DISABLE. If not specified defaults to CANCEL.

--recurringTask {schedulePattern}

Indicates the task is recurring and will be scheduled according to the value argument expressed in crontab(5) compatible time/date pattern.

-t | --start {startTime}

Indicates the date/time at which this operation will start when scheduled as a server task expressed in YYYYMMDDhhmmssZ format for UTC time or YYYYMMDDhhmmss for local time. A value of '0' will cause the task to be scheduled for immediate execution. When this option is specified the operation will be scheduled to start at the specified time after which this utility will exit immediately.

Utility input/output options:

--noPropertiesFile

No properties file will be used to get default command line argument values.

Default: false

--propertiesFilePath {propertiesFilePath}

Path to the file containing default property values used for command line arguments.

General options:

-V | --version

Display Directory Server version information.

Default: false

-H | --help

Display this usage information.

Default: false

Exit Codes

0

The command completed successfully.

1

An error occurred.

Examples

The following example backs up all user data while the server is online.

```
$ backup -p 4444 -D "cn=Directory Manager" -w password \
 -a -d /path/to/opendj/bak -t 0
Backup task 20110613143801866 scheduled to start ...
```

The following example schedules back up of all user data every night at 2 AM when the server is online, and notifies diradmin@example.com when finished, or on error.

```
$ backup -p 4444 -D "cn=Directory Manager" -w password -a \
 -d /path/to/opendj/bak --recurringTask "00 02 * * *" \
 --completionNotify diradmin@example.com --errorNotify diradmin@example.com
Recurring Backup task BackupTask-988d6adf-4d65-44bf-8546-6ea74a2480b0
scheduled successfully
```

The following example backs up all user data while the server is offline.

```
$ stop-ds
Stopping Server..
.
...

$ backup --backupAll --backupDirectory /path/to/opendj/bak
... msg=The backup process completed successfully

$ start-ds
... The Directory Server has started successfully
```

Name

base64 — encode and decode base64 strings

Synopsis

`base64` {subcommand} {options}

Description

This utility can be used to encode and decode information using base64.

Options

The **base64** command takes the following options:

-V | --version

Display Directory Server version information.

Default: false

-H | --help

Display this usage information.

Default: false

Subcommands

The **base64** command supports the following subcommands:

base64 decode

Decode base64-encoded information into raw data. When no options are specified, this subcommand reads from standard input and writes to standard output.

Options

The **base64 decode** command takes the following options:

-d | --encodedData {data}

The base64-encoded data to be decoded.

-f | --encodedDataFile {path}

The path to a file containing the base64-encoded data to be decoded.

-o | --toRawFile {path}

The path to a file to which the raw base64-decoded data should be written.

base64 encode

Encode raw data using base64. When no options are specified, this subcommand reads from standard input and writes to standard output.

Options

The **base64 encode** command takes the following options:

-d | --rawData {data}

The raw data to be base64 encoded.

-f | --rawDataFile {path}

The path to a file containing the raw data to be base64 encoded.

-o | --toEncodedFile {path}

The path to a file to which the base64-encoded data should be written.

Exit Codes

0

The command completed successfully.

> 0

An error occurred.

Examples

The following command shows the changes from the external change log in human-readable format.

```
$ base64 decode -d YWRkOiBkZXNjcmlwdGlvbgpkZXNjcmlwdGlvbjogQSB0aGlyZCBjaGFuZ2UK\
LQpyZXBsYWNlOiBtb2RpZmllcnNOYW1lCm1vZGlmaWVyc05hbWU6IGNuPURpcmVjdG9yeSBNYW5hZ2V\
yLGNuPVJvb3QgRE5zLGNuPWNvbmZpZwotCnJlcGxhY2U6IG1vZGlmeVRpbWVzdGFtcAptb2RpZnlUaW\
1lc3RhbXA6IDIwMTEwNjEzMDcxMjEwWgotCg==
add: description
description: A third change
-
replace: modifiersName
modifiersName: cn=Directory Manager,cn=Root DNs
,cn=config
-
replace: modifyTimestamp
modifyTimestamp: 20110613071210Z
-
```

Name

control-panel — start the graphical admin interface

Synopsis

```
control-panel
```

Description

This utility can be used to display the Control Panel window which displays basic server information and allows to do some basic administration tasks on the server.

If no host name or port is provided, the tool will try to connect to the local server.

Options

The **control-panel** command takes the following options:

Command options:

--connectTimeout {timeout}

Maximum length of time (in milliseconds) that can be taken to establish a connection. Use '0' to specify no time out.

Default: 30000

-r | --remote

Connect to a remote server.

Default: false

LDAP connection options:

-D | --bindDn {bindDN}

DN to use to bind to the server.

Default: cn=Directory Manager

-h | --hostname {host}

The fully-qualified directory server host name that will be used when generating self-signed certificates for LDAP SSL/StartTLS, the administration connector, and replication.

Default: localhost.localdomain

-j | --bindPasswordFile {bindPasswordFile}

Bind password file.

-p | --port {port}

Directory server administration port number.

Default: 4444

-w | --bindPassword {bindPassword}

Password to use to bind to the server. Use -w - to ensure that the command prompts for the password, rather than entering the password as a command argument.

-X | --trustAll

Trust all server SSL certificates.

Default: false

General options:

-V | --version

Display Directory Server version information.

Default: false

-H | --help

Display this usage information.

Default: false

Exit Codes

0

The command completed successfully.

> 0

An error occurred.

Examples

The following example starts the Control Panel on a remote host.

```
$ control-panel -r -h opendj.example.com -p 4444 &
```

Name

create-rc-script — script to manage OpenDJ as a service on UNIX

Synopsis

```
create-rc-script
```

Description

Create an RC script that may be used to start, stop, and restart the Directory Server on UNIX-based systems.

Options

The **create-rc-script** command takes the following options:

Command options:

-f | --outputFile {path}

The path to the output file to create.

-j | --javaHome {path}

The path to the Java installation that should be used to run the server.

-J | --javaArgs {args}

A set of arguments that should be passed to the JVM when running the server.

-u | --userName {userName}

The name of the user account under which the server should run.

General options:

-V | --version

Display Directory Server version information.

Default: false

-H | --help

Display this usage information.

Default: false

Exit Codes

0

The command completed successfully.

> 0

An error occurred.

Examples

The following example adds a script to start OpenDJ at boot time on a Debian-based system, and then updates the runlevel system to use the script.

```
$ sudo create-rc-script -f /etc/init.d/opendj -u opendj-user
$ sudo update-rc.d opendj
```

Name

dsconfig — manage OpenDJ server configuration

Synopsis

`dsconfig` {subcommand} {options}

Description

This utility can be used to define a base configuration for the Directory Server.

The **dsconfig** command is the primary command-line tool for viewing and editing OpenDJ configuration. When started without arguments, **dsconfig** prompts you for administration connection information, including the host name, administration port number, administrator bind DN and administrator password. The **dsconfig** command then connects securely to the directory server over the administration port. Once connected it presents you with a menu-driven interface to the server configuration.

When you pass connection information, subcommands, and additional options to **dsconfig**, the command runs in script mode and so is not interactive, though it can prompt you to ask whether to apply changes and whether to trust certificates (unless you use the `--no-prompt` and `--trustAll` options, respectively).

You can prepare **dsconfig** batch scripts by running the tool with the `--commandFilePath` option in interactive mode, then reading from the batch file with the `--batchFilePath` option in script mode. Batch files can be useful when you have many **dsconfig** commands to run and want to avoid starting the JVM for each command. Alternatively, you can read commands from standard input by using the `--batch` option.

The **dsconfig** command categorizes directory server configuration into *components*, also called *managed objects*. Actual components often inherit from a parent component type. For example, one component is a Connection Handler. An LDAP Connection Handler is a type of Connection Handler. You configure the LDAP Connection Handler component to specify how OpenDJ directory server handles LDAP connections coming from client applications.

Configuration components have *properties*. For example, the LDAP Connection Handler component has properties such as `listen-port` and `allow-start-tls`. You can set the component's `listen-port` property to `389` to use the default LDAP port number. You can set the component's `allow-start-tls` property to `true` to permit LDAP client applications to use StartTLS. Much of the configuration you do with **dsconfig** involves setting component properties.

Options

The **dsconfig** command takes the following options:

Command options:

--batch

Reads from standard input a set of commands to be executed.

Default: false

--commandFilePath {path}

The full path to the file where the equivalent non-interactive commands will be written when this command is run in interactive mode.

--connectTimeout {timeout}

Maximum length of time (in milliseconds) that can be taken to establish a connection. Use '0' to specify no time out.

Default: 30000

--displayCommand

Display the equivalent non-interactive argument in the standard output when this command is run in interactive mode.

Default: false

--help-all

Display all subcommands.

Default: false

--help-core-server

Display subcommands relating to core server.

Default: false

--help-database

Display subcommands relating to caching and backends.

Default: false

--help-logging

Display subcommands relating to logging.

Default: false

--help-proxy

Display subcommands relating to directory proxy.

Default: false

--help-replication

Display subcommands relating to replication.

Default: false

--help-security

Display subcommands relating to authentication and authorization.

Default: false

--help-service-discovery

Display subcommands relating to service discovery mechanism.

Default: false

--help-user-management

Display subcommands relating to user management.

Default: false

Configuration Options

--advanced

Allows the configuration of advanced components and properties.

Default: false

LDAP connection options:

-D | --bindDn {bindDN}

DN to use to bind to the server.

Default: cn=Directory Manager

-E | --reportAuthzId

Use the authorization identity control.

Default: false

-h | --hostname {host}

The fully-qualified directory server host name that will be used when generating self-signed certificates for LDAP SSL/StartTLS, the administration connector, and replication.

Default: localhost.localdomain

-j | --bindPasswordFile {bindPasswordFile}

Bind password file.

-K | --keyStorePath {keyStorePath}

Certificate key store path.

-N | --certNickname {nickname}

Nickname of the certificate that the server should use when accepting SSL-based connections or performing StartTLS negotiation.

-o | --saslOption {name=value}

SASL bind options.

-p | --port {port}

Directory server administration port number.

Default: 4444

-P | --trustStorePath {trustStorePath}

Certificate trust store path.

-T | --trustStorePassword {trustStorePassword}

Certificate trust store PIN.

-u | --keyStorePasswordFile {keyStorePasswordFile}

Certificate key store PIN file. A PIN is required when you specify to use an existing certificate as server certificate.

-U | --trustStorePasswordFile {path}

Certificate trust store PIN file.

--usePasswordPolicyControl

Use the password policy request control.

Default: false

-w | --bindPassword {bindPassword}

Password to use to bind to the server. Use -w - to ensure that the command prompts for the password, rather than entering the password as a command argument.

-W | --keyStorePassword {keyStorePassword}

Certificate key store PIN. A PIN is required when you specify to use an existing certificate as server certificate.

-X | --trustAll

Trust all server SSL certificates.

Default: false

Utility input/output options:

-F | --batchFilePath {batchFilePath}

Path to a batch file containing a set of commands to be executed.

-n | --no-prompt

Use non-interactive mode. If data in the command is missing, the user is not prompted and the tool will fail.

Default: false

--noPropertiesFile

No properties file will be used to get default command line argument values.

Default: false

--propertiesFilePath {propertiesFilePath}

Path to the file containing default property values used for command line arguments.

-Q | --quiet

Use quiet mode.

Default: false

-s | --script-friendly

Use script-friendly mode.

Default: false

-v | --verbose

Use verbose mode.

Default: false

General options:

-V | --version

Display Directory Server version information.

Default: false

-H | --help

Display this usage information.

Default: false

Subcommands

The **dsconfig** command provides many subcommands.

Subcommands let you create, list, and delete entire configuration components, and also let you get and set component properties. Subcommands therefore have names that reflect these five actions.

- create-*component*
- list-*component*s
- delete-*component*
- get-*component*-prop
- set-*component*-prop

Here, *component* names are names of managed object types. Subcommand *component* names are lower-case, hyphenated versions of the friendly names. When you act on an actual configuration component, you provide the name of the component as an option argument.

For example, the Log Publisher component has these corresponding subcommands.

- **create-log-publisher**
- **list-log-publishers**
- **delete-log-publisher**
- **get-log-publisher-prop**
- **set-log-publisher-prop**

When you create or delete Log Publisher components and when you get and set their configuration properties, you provide the name of the actual log publisher, which you can find by using the **list-log-publishers** subcommand:

```
# Get the log publishers' names:
$ dsconfig \
 list-log-publishers \
 --hostname opendj.example.com \
 --port 4444 \
 --bindDN "cn=Directory Manager" \
 --bindPassword password \
 --trustAll \
 --no-prompt
Log Publisher                     : Type                   : enabled
----------------------------------:------------------------:--------
...
Json File-Based Access Logger     : json-file-access       : true
...

# Use the name to read a property:
$ dsconfig \
 get-log-publisher-prop \
 --publisher-name "Json File-Based Access Logger" \
 --property rotation-policy \
```

```
 --hostname opendj.example.com \
 --port 4444 \
 --bindDN "cn=Directory Manager" \
 --bindPassword password \
 --trustAll \
 --no-prompt
Property        : Value(s)
----------------:---------------------------------------------------------
rotation-policy : 24 Hours Time Limit Rotation Policy, Size Limit Rotation
                : Policy
```

Many subcommands let you set property values. Notice in the reference for the subcommands below that specific options are available for handling multi-valued properties. Whereas you can assign a single property value by using the `--set` option, you assign multiple values to a multi-valued property by using the `--add` option. You can reset the values of the multi-valued property by using the `--reset` option.

Some property values take a time duration. Durations are expressed as numbers followed by units. For example `1 s` means one second, and `2 w` means two weeks. Some durations have minimum granularity or maximum units, so you cannot necessary specify every duration in milliseconds or weeks for example. Some durations allow you to use a special value to mean unlimited. Units are specified as follows.

- `ms`: milliseconds
- `s`: seconds
- `m`: minutes
- `h`: hours
- `d`: days
- `w`: weeks

Use the `--help*` options described above to view help for subcommands.

For help with individual subcommands, either use **dsconfig *`subcommand`* --help**, or start **dsconfig** in interactive mode, without specifying a subcommand.

To view all component properties, use the **dsconfig list-properties** command.

Exit Codes

0

The command completed successfully.

> 0

An error occurred.

Examples

Much of the *OpenDJ Administration Guide* consists of **dsconfig** examples with text in between. This section therefore remains short.

The following example starts **dsconfig** in interactive, menu-driven mode on the default port of the current host.

```
$ dsconfig -h opendj.example.com -p 4444 -D "cn=Directory Manager" -w password

>>>> OpenDJ configuration console main menu

What do you want to configure?

    1)   Access Control Handler               23)  Log Publisher
    2)   Access Log Filtering Criteria        24)  Log Retention Policy
    3)   Account Status Notification Handler  25)  Log Rotation Policy
    4)   Administration Connector             26)  Monitor Provider
    5)   Alert Handler                        27)  Password Generator
    6)   Backend                              28)  Password Policy
    7)   Backend Index                        29)  Password Storage Scheme
    8)   Backend VLV Index                    30)  Password Validator
    9)   Certificate Mapper                   31)  Plugin
    10)  Connection Handler                   32)  Plugin Root
    11)  Crypto Manager                       33)  Replication Domain
    12)  Debug Target                         34)  Replication Server
    13)  Entry Cache                          35)  Root DN
    14)  Extended Operation Handler           36)  Root DSE Backend
    15)  External Changelog Domain            37)  SASL Mechanism Handler
    16)  Global Access Control Policy         38)  Schema Provider
    17)  Global Configuration                 39)  Service Discovery Mechanism
    18)  Group Implementation                 40)  Synchronization Provider
    19)  HTTP Authorization Mechanism         41)  Trust Manager Provider
    20)  HTTP Endpoint                        42)  Virtual Attribute
    21)  Identity Mapper                      43)  Work Queue
    22)  Key Manager Provider

    q)   quit

Enter choice:
```

The following example demonstrates generating a batch file that corresponds to an interactive session enabling the debug log. The example then demonstrates using a modified batch file to disable the debug log.

```
$ dsconfig \
 --hostname opendj.example.com \
 --port 4444 \
 --bindDN "cn=Directory Manager" \
 --bindPassword password \
 --commandFilePath ~/enable-debug-log.batch
 ...
$ cat ~/enable-debug-log.batch
# dsconfig session start date: 19/Oct/2011:08:52:22 +0000
```

```
# Session operation number: 1
# Operation date: 19/Oct/2011:08:55:06 +0000
dsconfig set-log-publisher-prop \
          --publisher-name File-Based\ Debug\ Logger \
          --set enabled:true \
          --hostname opendj.example.com \
          --port 4444 \
          --trustStorePath /path/to/opendj/config/admin-truststore \
          --bindDN cn=Directory\ Manager \
          --bindPassword ****** \
          --no-prompt

$ cp ~/enable-debug-log.batch ~/disable-debug-log.batch
$ vi ~/disable-debug-log.batch
$ cat ~/disable-debug-log.batch
set-log-publisher-prop \
          --publisher-name File-Based\ Debug\ Logger \
          --set enabled:false \
          --hostname opendj.example.com \
          --port 4444 \
          --trustStorePath /path/to/opendj/config/admin-truststore \
          --bindDN cn=Directory\ Manager \
          --bindPassword password \
          --no-prompt

$ dsconfig --batchFilePath ~/disable-debug-log.batch --no-prompt
set-log-publisher-
prop
--publisher-name
File-Based Debug
 Logger
--set
enabled:false
--hostname
opendj.example
.com
--port
4444
--
trustStorePath
/path/to/opendj/config/admin-
truststore
--bindDN
cn=Directory Manager
--bindPassword
password
--no-prompt

$
```

Notice that the original command file looks like a shell script with the bind password value replaced by asterisks. To pass the content as a batch file to **dsconfig**, strip `dsconfig` itself, and include the bind password for the administrative user or replace that option with an alternative, such as reading the password from a file.

Name

dsreplication — manage directory data replication

Synopsis

`dsreplication` {subcommand} {options}

Description

This utility can be used to configure replication between servers so that the data of the servers is synchronized. For replication to work you must first enable replication using the 'enable' subcommand and then initialize the contents of one of the servers with the contents of the other using the 'initialize' subcommand.

Options

The **dsreplication** command takes the following options:

Command options:

-b | --baseDn {baseDN}

Base DN of the data to be replicated, initialized or for which we want to unconfigure replication. Multiple base DNs can be provided by using this option multiple times.

--commandFilePath {path}

The full path to the file where the equivalent non-interactive commands will be written when this command is run in interactive mode.

--connectTimeout {timeout}

Maximum length of time (in milliseconds) that can be taken to establish a connection. Use '0' to specify no time out.

Default: 30000

--displayCommand

Display the equivalent non-interactive argument in the standard output when this command is run in interactive mode.

Default: false

-j | --adminPasswordFile {bindPasswordFile}

The file containing the password of the global administrator.

-w | --adminPassword {bindPassword}

The global administrator password.

Configuration Options

--advanced

Allows the configuration of advanced components and properties.

Default: false

LDAP connection options:

-I | --adminUid {adminUID}

User ID of the Global Administrator to use to bind to the server. For the 'enable' subcommand if no Global Administrator was defined previously for none of the server the Global Administrator will be created using the provided data.

Default: admin

-K | --keyStorePath {keyStorePath}

Certificate key store path.

-N | --certNickname {nickname}

Nickname of the certificate that the server should use when accepting SSL-based connections or performing StartTLS negotiation.

-o | --saslOption {name=value}

SASL bind options.

-P | --trustStorePath {trustStorePath}

Certificate trust store path.

-T | --trustStorePassword {trustStorePassword}

Certificate trust store PIN.

-u | --keyStorePasswordFile {keyStorePasswordFile}

Certificate key store PIN file. A PIN is required when you specify to use an existing certificate as server certificate.

-U | --trustStorePasswordFile {path}

Certificate trust store PIN file.

-W | --keyStorePassword {keyStorePassword}

Certificate key store PIN. A PIN is required when you specify to use an existing certificate as server certificate.

-X | --trustAll

Trust all server SSL certificates.

Default: false

Utility input/output options:

-n | --no-prompt

Use non-interactive mode. If data in the command is missing, the user is not prompted and the tool will fail.

Default: false

--noPropertiesFile

No properties file will be used to get default command line argument values.

Default: false

--propertiesFilePath {propertiesFilePath}

Path to the file containing default property values used for command line arguments.

-Q | --quiet

Use quiet mode.

Default: false

General options:

-V | --version

Display Directory Server version information.

Default: false

-H | --help

Display this usage information.

Default: false

Subcommands

The **dsreplication** command supports the following subcommands:

dsreplication configure

Updates the configuration of the servers to replicate the data under the specified base DN. If one of the specified servers is already replicating the data under the base DN with other servers, executing this subcommand will update the configuration of all the servers (so it is sufficient to execute the command line once for each server we add to the replication topology).

Options

The **dsreplication configure** command takes the following options:

-h | --host1 {host}

Fully qualified host name or IP address of the first server whose contents will be replicated.

Default: localhost.localdomain

-p | --port1 {port}

Directory server administration port number of the first server whose contents will be replicated.

Default: 4444

-D | --bindDn1 {bindDN}

DN to use to bind to the first server whose contents will be replicated. If not specified the global administrator will be used to bind.

Default: cn=Directory Manager

--bindPassword1 {bindPassword}

Password to use to bind to the first server whose contents will be replicated. If no bind DN was specified for the first server the password of the global administrator will be used to bind.

--bindPasswordFile1 {bindPasswordFile}

File containing the password to use to bind to the first server whose contents will be replicated. If no bind DN was specified for the first server the password of the global administrator will be used to bind.

-r | --replicationPort1 {port}

Port that will be used by the replication mechanism in the first server to communicate with the other servers. You have to specify this option only if replication was not previously configured in the first server.

Default: 8989

--secureReplication1

Specifies whether the communication through the replication port of the first server is encrypted or not. This option will only be taken into account the first time replication is configured on the first server.

Default: false

--noReplicationServer1

Do not configure a replication port or change log on the first server. The first server will contain replicated data but will not contain a change log of modifications made to the replicated data. Note that each replicated topology must contain at least two servers with a change log to avoid a single point of failure.

Default: false

--onlyReplicationServer1

Configure only a change log and replication port on the first server. The first server will not contain replicated data, but will contain a change log of the modifications made to the replicated data on other servers.

Default: false

-O | --host2 {host}

Fully qualified host name or IP address of the second server whose contents will be replicated.

Default: localhost.localdomain

--port2 {port}

Directory server administration port number of the second server whose contents will be replicated.

Default: 4444

--bindDn2 {bindDN}

DN to use to bind to the second server whose contents will be replicated. If not specified the global administrator will be used to bind.

Default: cn=Directory Manager

--bindPassword2 {bindPassword}

Password to use to bind to the second server whose contents will be replicated. If no bind DN was specified for the second server the password of the global administrator will be used to bind.

-F | --bindPasswordFile2 {bindPasswordFile}

File containing the password to use to bind to the second server whose contents will be replicated. If no bind DN was specified for the second server the password of the global administrator will be used to bind.

-R | --replicationPort2 {port}

Port that will be used by the replication mechanism in the second server to communicate with the other servers. You have to specify this option only if replication was not previously configured in the second server.

Default: 8989

--secureReplication2

Specifies whether the communication through the replication port of the second server is encrypted or not. This option will only be taken into account the first time replication is configured on the second server.

Default: false

--noReplicationServer2

Do not configure a replication port or change log on the second server. The second server will contain replicated data but will not contain a change log of modifications made to the replicated data. Note that each replicated topology must contain at least two servers with a change log to avoid a single point of failure.

Default: false

--onlyReplicationServer2

Configure only a change log and replication port on the second server. The second server will not contain replicated data, but will contain a change log of the modifications made to the replicated data on other servers.

Default: false

-S | --skipPortCheck

Skip the check to determine whether the specified replication ports are usable.

Default: false

--noSchemaReplication

Do not replicate the schema between the servers.

Default: false

--useSecondServerAsSchemaSource

Use the second server to initialize the schema of the first server. If this option nor option --noSchemaReplication are specified the schema of the first server will be used to initialize the schema of the second server.

Default: false

dsreplication initialize

Initialize the contents of the data under the specified base DN on the destination server with the contents on the source server. This operation is required after enabling replication in order replication to work ('initialize-all' can also be used for this purpose).

Options

The **dsreplication initialize** command takes the following options:

-h | --hostSource {host}

Fully qualified host name or IP address of the source server whose contents will be used to initialize the destination server.

Default: localhost.localdomain

-p | --portSource {port}

Directory server administration port number of the source server whose contents will be used to initialize the destination server.

Default: 4444

-O | --hostDestination {host}

Fully qualified host name or IP address of the destination server whose contents will be initialized.

Default: localhost.localdomain

--portDestination {port}

Directory server administration port number of the destination server whose contents will be initialized.

Default: 4444

dsreplication initialize-all

Initialize the contents of the data under the specified base DN on all the servers whose contents are being replicated with the contents on the specified server. This operation is required after enabling replication for replication to work ('initialize' applied to each server can also be used for this purpose).

Options

The **dsreplication initialize-all** command takes the following options:

-h | --hostname {host}

The fully-qualified directory server host name that will be used when generating self-signed certificates for LDAP SSL/StartTLS, the administration connector, and replication.

Default: localhost.localdomain

-p | --port {port}

Directory server administration port number.

Default: 4444

dsreplication post-external-initialization

This subcommand must be called after initializing the contents of all the replicated servers using the tool import-ldif or the binary copy method. You must specify the list of base DNs that have been

initialized and you must provide the credentials of any of the servers that are being replicated. See the usage of the subcommand 'pre-external-initialization' for more information.

Options

The **dsreplication post-external-initialization** command takes the following options:

-h | --hostname {host}

The fully-qualified directory server host name that will be used when generating self-signed certificates for LDAP SSL/StartTLS, the administration connector, and replication.

Default: localhost.localdomain

-p | --port {port}

Directory server administration port number.

Default: 4444

dsreplication pre-external-initialization

This subcommand must be called before initializing the contents of all the replicated servers using the tool import-ldif or the binary copy method. You must specify the list of base DNs that will be initialized and you must provide the credentials of any of the servers that are being replicated. After calling this subcommand, initialize the contents of all the servers in the topology (use the same LDIF file/binary copy on each of the servers), then call the subcommand 'post-external-initialization'.

Options

The **dsreplication pre-external-initialization** command takes the following options:

-h | --hostname {host}

The fully-qualified directory server host name that will be used when generating self-signed certificates for LDAP SSL/StartTLS, the administration connector, and replication.

Default: localhost.localdomain

-p | --port {port}

Directory server administration port number.

Default: 4444

dsreplication purge-historical

Launches a purge processing of the historical informations stored in the user entries by replication. Since this processing may take a while, you must specify the maximum duration for this processing.

Options

The **dsreplication purge-historical** command takes the following options:

-h | --hostname {host}

The fully-qualified directory server host name that will be used when generating self-signed certificates for LDAP SSL/StartTLS, the administration connector, and replication.

Default: localhost.localdomain

-p | --port {port}

Directory server administration port number.

Default: 4444

--maximumDuration {maximum duration}

This argument specifies the maximum duration the purge processing must last expressed in seconds.

Default: 3600

-t | --start {startTime}

Indicates the date/time at which this operation will start when scheduled as a server task expressed in YYYYMMDDhhmmssZ format for UTC time or YYYYMMDDhhmmss for local time. A value of '0' will cause the task to be scheduled for immediate execution. When this option is specified the operation will be scheduled to start at the specified time after which this utility will exit immediately.

--recurringTask {schedulePattern}

Indicates the task is recurring and will be scheduled according to the value argument expressed in crontab(5) compatible time/date pattern.

--completionNotify {emailAddress}

Email address of a recipient to be notified when the task completes. This option may be specified more than once.

--errorNotify {emailAddress}

Email address of a recipient to be notified if an error occurs when this task executes. This option may be specified more than once.

--dependency {taskID}

ID of a task upon which this task depends. A task will not start execution until all its dependencies have completed execution.

--failedDependencyAction {action}

Action this task will take should one if its dependent tasks fail. The value must be one of PROCESS,CANCEL,DISABLE. If not specified defaults to CANCEL.

dsreplication reset-change-number

Re-synchronizes the change-log changenumber on one server with the change-log changenumber of another.

Options

The **dsreplication reset-change-number** command takes the following options:

-h | --hostSource {host}

Fully qualified host name or IP address of the source server whose contents will be used to initialize the destination server.

Default: localhost.localdomain

-p | --portSource {port}

Directory server administration port number of the source server whose contents will be used to initialize the destination server.

Default: 4444

-O | --hostDestination {host}

Fully qualified host name or IP address of the destination server whose contents will be initialized.

Default: localhost.localdomain

--portDestination {port}

Directory server administration port number of the destination server whose contents will be initialized.

Default: 4444

--change-number {change number}

The change number to use as the basis for re-synchronization.

dsreplication resume

Resumes replication on the specified server.

Options

The **dsreplication resume** command takes the following options:

-h | --hostname {host}

The fully-qualified directory server host name that will be used when generating self-signed certificates for LDAP SSL/StartTLS, the administration connector, and replication.

Default: localhost.localdomain

-p | --port {port}

Directory server administration port number.

Default: 4444

dsreplication status

Displays a list with the basic replication configuration of the base DNs of the servers defined in the registration information. If no base DNs are specified as parameter the information for all base DNs is displayed.

Options

The **dsreplication status** command takes the following options:

-h | --hostname {host}

The fully-qualified directory server host name that will be used when generating self-signed certificates for LDAP SSL/StartTLS, the administration connector, and replication.

Default: localhost.localdomain

-p | --port {port}

Directory server administration port number.

Default: 4444

-s | --script-friendly

Use script-friendly mode.

Default: false

dsreplication suspend

Suspends (pauses) replication on the specified server.

Options

The **dsreplication suspend** command takes the following options:

-h | --hostname {host}

The fully-qualified directory server host name that will be used when generating self-signed certificates for LDAP SSL/StartTLS, the administration connector, and replication.

Default: localhost.localdomain

-p | --port {port}

Directory server administration port number.

Default: 4444

dsreplication unconfigure

Unconfigures replication on the specified server for the provided base DN and removes references in the other servers with which it is replicating data.

Options

The **dsreplication unconfigure** command takes the following options:

-h | --hostname {host}

The fully-qualified directory server host name that will be used when generating self-signed certificates for LDAP SSL/StartTLS, the administration connector, and replication.

Default: localhost.localdomain

-p | --port {port}

Directory server administration port number.

Default: 4444

-D | --bindDn {bindDN}

DN to use to bind to the server where we want to unconfigure replication. This option must be used when no Global Administrator has been defined on the server or if the user does not want to remove references in the other replicated servers. The password provided for the Global Administrator will be used when specifying this option.

Default: cn=Directory Manager

-a | --unconfigureReplicationServer

Unconfigure the replication server. The replication port and change log will be unconfigured on the specified server.

Default: false

--unconfigureAll

Unconfigure the replication configuration on the specified server. The contents of the server are no longer replicated and the replication server (changelog and replication port) is unconfigured if it is configured.

Default: false

Exit Codes

0

The command completed successfully.

> 0

An error occurred.

Examples

The following example configures and then initializes replication for a new replica on `opendj2.example.com` from an existing replica on `opendj.example.com`.

```
$ dsreplication configure -I admin -w password -X -n -b dc=example,dc=com \
 --host1 opendj.example.com --port1 4444 --bindDN1 "cn=Directory Manager" \
 --bindPassword1 password --replicationPort1 8989 \
 --host2 opendj2.example.com --port2 4444 --bindDN2 "cn=Directory Manager" \
 --bindPassword2 password --replicationPort2 8989

$ dsreplication initialize-all -I admin -w password -X -n -b dc=example,dc=com \
 -h opendj.example.com -p 4444
```

Name

encode-password — encode a password with a storage scheme

Synopsis

```
encode-password
```

Description

This utility can be used to encode user passwords with a specified storage scheme, or to determine whether a given clear-text value matches a provided encoded password.

Options

The **encode-password** command takes the following options:

Command options:

-a | --authPasswordSyntax

Use the authentication password syntax rather than the user password syntax.

Default: false

-c | --clearPassword {clearPW}

Clear-text password to encode or to compare against an encoded password.

-e | --encodedPassword {encodedPW}

Encoded password to compare against the clear-text password.

-E | --encodedPasswordFile {file}

Encoded password file.

-f | --clearPasswordFile {file}

Clear-text password file.

-i | --interactivePassword

The password to encode or to compare against an encoded password is interactively asked to the user.

Default: false

-l | --listSchemes

List available password storage schemes.

Default: false

-r | --useCompareResultCode

Use the LDAP compare result as an exit code for the password comparison.

Default: false

-s | --storageScheme {scheme}

Scheme to use for the encoded password.

General options:

-V | --version

Display Directory Server version information.

Default: false

-H | --help

Display this usage information.

Default: false

Exit Codes

0

The command completed successfully.

5

The `-r` option was used, and the compare did not match.

6

The `-r` option was used, and the compare did match.

other

An error occurred.

Examples

The following example encodes a password, and also shows comparison of a password with the encoded value.

```
$ encode-password -l
3DES
AES
BASE64
BCRYPT
BLOWFISH
CLEAR
CRYPT
MD5
PBKDF2
PKCS5S2
RC4
SHA
SMD5
SSHA
SSHA256
SSHA384
SSHA512

$ encode-password -c secret12 -s CRYPT
Encoded Password:  "{CRYPT}ZulJ6Dy3TFnrE"

$ encode-password -c secret12 -s CRYPT -e "{CRYPT}ZulJ6Dy3TFnrE" -r
The provided clear-text and encoded passwords match

$ echo $?
6
```

Name

export-ldif — export directory data in LDIF

Synopsis

```
export-ldif
```

Description

This utility can be used to export data from a Directory Server backend in LDIF form.

Options

The **export-ldif** command takes the following options:

Command options:

-a | --appendToLdif

Append an existing LDIF file rather than overwriting it.

Default: false

-b | --includeBranch {branchDN}

Base DN of a branch to include in the LDIF export.

-B | --excludeBranch {branchDN}

Base DN of a branch to exclude from the LDIF export.

-c | --compress

Compress the LDIF data as it is exported.

Default: false

-e | --excludeAttribute {attribute}

Attribute to exclude from the LDIF export.

-E | --excludeFilter {filter}

Filter to identify entries to exclude from the LDIF export.

-i | --includeAttribute {attribute}

Attribute to include in the LDIF export.

-I | --includeFilter {filter}

Filter to identify entries to include in the LDIF export.

-l | --ldifFile {ldifFile}

Path to the LDIF file to be written.

-n | --backendId {backendName}

Backend ID for the backend to export.

-O | --excludeOperational

Exclude operational attributes from the LDIF export.

Default: false

--offline

Indicates that the command must be run in offline mode.

Default: false

Task Backend Connection Options

--connectTimeout {timeout}

Maximum length of time (in milliseconds) that can be taken to establish a connection. Use '0' to specify no time out.

Default: 30000

-D | --bindDn {bindDN}

DN to use to bind to the server.

Default: cn=Directory Manager

-h | --hostname {host}

The fully-qualified directory server host name that will be used when generating self-signed certificates for LDAP SSL/StartTLS, the administration connector, and replication.

Default: localhost.localdomain

-j | --bindPasswordFile {bindPasswordFile}

Bind password file.

-K | --keyStorePath {keyStorePath}

Certificate key store path.

-N | --certNickname {nickname}

Nickname of the certificate that the server should use when accepting SSL-based connections or performing StartTLS negotiation.

-o | --saslOption {name=value}

SASL bind options.

-p | --port {port}

Directory server administration port number.

Default: 4444

-P | --trustStorePath {trustStorePath}

Certificate trust store path.

-T | --trustStorePassword {trustStorePassword}

Certificate trust store PIN.

-u | --keyStorePasswordFile {keyStorePasswordFile}

Certificate key store PIN file. A PIN is required when you specify to use an existing certificate as server certificate.

-U | --trustStorePasswordFile {path}

Certificate trust store PIN file.

-w | --bindPassword {bindPassword}

Password to use to bind to the server. Use -w - to ensure that the command prompts for the password, rather than entering the password as a command argument.

-W | --keyStorePassword {keyStorePassword}

Certificate key store PIN. A PIN is required when you specify to use an existing certificate as server certificate.

-X | --trustAll

Trust all server SSL certificates.

Default: false

Task Scheduling Options

--completionNotify {emailAddress}

Email address of a recipient to be notified when the task completes. This option may be specified more than once.

--dependency {taskID}

ID of a task upon which this task depends. A task will not start execution until all its dependencies have completed execution.

--errorNotify {emailAddress}

Email address of a recipient to be notified if an error occurs when this task executes. This option may be specified more than once.

--failedDependencyAction {action}

Action this task will take should one if its dependent tasks fail. The value must be one of PROCESS,CANCEL,DISABLE. If not specified defaults to CANCEL.

--recurringTask {schedulePattern}

Indicates the task is recurring and will be scheduled according to the value argument expressed in crontab(5) compatible time/date pattern.

-t | --start {startTime}

Indicates the date/time at which this operation will start when scheduled as a server task expressed in YYYYMMDDhhmmssZ format for UTC time or YYYYMMDDhhmmss for local time. A value of '0' will cause the task to be scheduled for immediate execution. When this option is specified the operation will be scheduled to start at the specified time after which this utility will exit immediately.

Utility input/output options:

--noPropertiesFile

No properties file will be used to get default command line argument values.

Default: false

--propertiesFilePath {propertiesFilePath}

Path to the file containing default property values used for command line arguments.

--wrapColumn {wrapColumn}

Column at which to wrap long lines (0 for no wrapping).

Default: 0

General options:

-V | --version

Display Directory Server version information.

Default: false

-H | --help

Display this usage information.

Default: false

Exit Codes

0

The command completed successfully.

> 0

An error occurred.

Examples

The following example exports data to a file, `Example.ldif`, with the server offline.

```
$ export-ldif -b dc=example,dc=com -n userRoot -l ../ldif/Example.ldif
... category=BACKEND severity=INFORMATION ..
.
...Exported 160 entries and skipped 0 in 0 seconds (average rate 1428.6/sec)
```

Name

import-ldif — import directory data from LDIF

Synopsis

```
import-ldif
```

Description

This utility can be used to import LDIF data into a Directory Server backend, overwriting existing data. It cannot be used to append data to the backend database.

Options

The **import-ldif** command takes the following options:

Command options:

-A | --templateFile {templateFile}

Path to a MakeLDIF template to use to generate the import data.

-b | --includeBranch {branchDN}

Base DN of a branch to include in the LDIF import.

-B | --excludeBranch {branchDN}

Base DN of a branch to exclude from the LDIF import.

-c | --isCompressed

LDIF file is compressed.

Default: false

--countRejects

Count the number of entries rejected by the server and return that value as the exit code (values > 255 will be reduced to 255 due to exit code restrictions).

Default: false

-e | --excludeAttribute {attribute}

Attribute to exclude from the LDIF import.

-E | --excludeFilter {filter}

Filter to identify entries to exclude from the LDIF import.

-F | --clearBackend

Remove all entries for all base DNs in the backend before importing.

Default: false

-i | --includeAttribute {attribute}

Attribute to include in the LDIF import.

-I | --includeFilter {filter}

Filter to identify entries to include in the LDIF import.

-l | --ldifFile {ldifFile}

Path to the LDIF file to be imported.

-n | --backendId {backendName}

Backend ID for the backend to import.

-O | --overwrite

Overwrite an existing rejects and/or skip file rather than appending to it.

Default: false

--offline

Indicates that the command must be run in offline mode.

Default: false

-R | --rejectFile {rejectFile}

Write rejected entries to the specified file.

-s | --randomSeed {seed}

Seed for the MakeLDIF random number generator.

Default: 0

-S | --skipSchemaValidation

Skip schema validation during the LDIF import.

Default: false

--skipFile {skipFile}

Write skipped entries to the specified file.

--threadCount {count}

Number of threads used to read LDIF file during import. Default value (0) equals: 2 x (number of CPUs).

Default: 0

--tmpDirectory {directory}

Path to temporary directory for index scratch files during LDIF import.

Default: import-tmp

Task Backend Connection Options

--connectTimeout {timeout}

Maximum length of time (in milliseconds) that can be taken to establish a connection. Use '0' to specify no time out.

Default: 30000

-D | --bindDn {bindDN}

DN to use to bind to the server.

Default: cn=Directory Manager

-h | --hostname {host}

The fully-qualified directory server host name that will be used when generating self-signed certificates for LDAP SSL/StartTLS, the administration connector, and replication.

Default: localhost.localdomain

-j | --bindPasswordFile {bindPasswordFile}

Bind password file.

-K | --keyStorePath {keyStorePath}

Certificate key store path.

-N | --certNickname {nickname}

Nickname of the certificate that the server should use when accepting SSL-based connections or performing StartTLS negotiation.

-o | --saslOption {name=value}

SASL bind options.

-p | --port {port}

Directory server administration port number.

Default: 4444

-P | --trustStorePath {trustStorePath}

Certificate trust store path.

-T | --trustStorePassword {trustStorePassword}

Certificate trust store PIN.

-u | --keyStorePasswordFile {keyStorePasswordFile}

Certificate key store PIN file. A PIN is required when you specify to use an existing certificate as server certificate.

-U | --trustStorePasswordFile {path}

Certificate trust store PIN file.

-w | --bindPassword {bindPassword}

Password to use to bind to the server. Use -w - to ensure that the command prompts for the password, rather than entering the password as a command argument.

-W | --keyStorePassword {keyStorePassword}

Certificate key store PIN. A PIN is required when you specify to use an existing certificate as server certificate.

-X | --trustAll

Trust all server SSL certificates.

Default: false

Task Scheduling Options

--completionNotify {emailAddress}

Email address of a recipient to be notified when the task completes. This option may be specified more than once.

--dependency {taskID}

ID of a task upon which this task depends. A task will not start execution until all its dependencies have completed execution.

--errorNotify {emailAddress}

Email address of a recipient to be notified if an error occurs when this task executes. This option may be specified more than once.

--failedDependencyAction {action}

Action this task will take should one if its dependent tasks fail. The value must be one of PROCESS,CANCEL,DISABLE. If not specified defaults to CANCEL.

--recurringTask {schedulePattern}

Indicates the task is recurring and will be scheduled according to the value argument expressed in crontab(5) compatible time/date pattern.

-t | --start {startTime}

Indicates the date/time at which this operation will start when scheduled as a server task expressed in YYYYMMDDhhmmssZ format for UTC time or YYYYMMDDhhmmss for local time. A value of '0' will cause the task to be scheduled for immediate execution. When this option is specified the operation will be scheduled to start at the specified time after which this utility will exit immediately.

Utility input/output options:

--noPropertiesFile

No properties file will be used to get default command line argument values.

Default: false

--propertiesFilePath {propertiesFilePath}

Path to the file containing default property values used for command line arguments.

-Q | --quiet

Use quiet mode (no output).

Default: false

General options:

-V | --version

Display Directory Server version information.

Default: false

-H | --help

Display this usage information.

Default: false

Exit Codes

0

The command completed successfully.

> 0

An error occurred.

Examples

The following example imports the content of a file, `Example.ldif`, with the server offline.

```
$ import-ldif -b dc=example,dc=com -n userRoot -l /path/to/Example.ldif
... category=RUNTIME_INFORMATION severity=NOTICE..
.
... msg=Import LDIF environment close took 0 seconds
```

Name

ldapcompare — perform LDAP compare operations

Synopsis

`ldapcompare` attribute:value DN

Description

This utility can be used to perform LDAP compare operations in the Directory Server.

Options

The **ldapcompare** command takes the following options:

Command options:

`--assertionFilter {filter}`

Use the LDAP assertion control with the provided filter.

`--connectTimeout {timeout}`

Maximum length of time (in milliseconds) that can be taken to establish a connection. Use '0' to specify no time out.

Default: 30000

`-J | --control {controloid[:criticality[:value|::b64value|:<filePath]]}`

Use a request control with the provided information.

For some *`controloid`* values, you can replace object identifiers with user-friendly strings. The strings are listed here in lower case, but the case is not important. You can use camelCase if you prefer, for example.

`Assertion`
`LdapAssertion`

Assertion Request Control, Object Identifier: 1.3.6.1.1.12

`AccountUsable`
`AccountUsability`

Account Usability Request Control, Object Identifier: 1.3.6.1.4.1.42.2.27.9.5.8

`AuthzId`
`AuthorizationIdentity`

Authorization Identity Request Control, Object Identifier: 2.16.840.1.113730.3.4.16

`Csn`
`ChangeNumber`
`ChangeSequenceNumber`

Change Sequence Number Request Control, Object Identifier: 1.3.6.1.4.1.42.2.27.9.5.9

This is an internal OpenDJ control.

`EffectiveRights`
`GetEffectiveRights`

Get Effective Rights Request Control, Object Identifier: 1.3.6.1.4.1.42.2.27.9.5.2

`ManageDsaIt`

Manage DSAIT Request Control, Object Identifier: 2.16.840.1.113730.3.4.2

`Noop`
`No-Op`

No-Op Request Control, Object Identifier: 1.3.6.1.4.1.4203.1.10.2

`PwPolicy`
`PasswordPolicy`

Password Policy Request Control, Object Identifier: 1.3.6.1.4.1.42.2.27.8.5.1

`PermissiveModify`

Permissive Modify Request Control, Object Identifier: 1.2.840.113556.1.4.1413

`PSearch`
`PersistentSearch`

Persistent Search Request Control, Object Identifier: 2.16.840.1.113730.3.4.3

`PostRead`

Post Read Request Control, Object Identifier: 1.3.6.1.1.13.2

`PreRead`

Pre Read Request Control, Object Identifier: 1.3.6.1.1.13.1

`ProxiedAuthV1`

Proxied Authorization Request Control V1, Object Identifier: 2.16.840.1.113730.3.4.12

`ProxiedAuth`
`ProxiedAuthV2`

Proxied Authorization Request Control V2, Object Identifier: 2.16.840.1.113730.3.4.18

`RealAttrsOnly`
`RealAttributesOnly`

Real Attributes Only Request Control, Object Identifier: 2.16.840.1.113730.3.4.17

`TreeDelete`
`SubTreeDelete`

Subtree Delete Request Control, Object Identifier: 1.2.840.113556.1.4.805

`Sort`
`ServerSideSort`

Server Side Sort Request Control, Object Identifier: 1.2.840.113556.1.4.473

`PagedResults`
`SimplePagedResults`

Simple Paged Results Control, Object Identifier: 1.2.840.113556.1.4.319

`SubEntries`

Sub-Entries Request Control, Object Identifier: 1.3.6.1.4.1.4203.1.10.1

`TxnId`
`TransactionId`

Transaction ID Control, Object Identifier: 1.3.6.1.4.1.36733.2.1.5.1

This is an internal ForgeRock control.

`VirtualAttrsOnly`
`VirtualAttributesOnly`

Virtual Attributes Only Request Control, Object Identifier: 2.16.840.1.113730.3.4.19

`Vlv`
`VirtualListView`

Virtual List View Request Control, Object Identifier: 2.16.840.1.113730.3.4.9

`-m | --useCompareResultCode`

Use the LDAP compare result as an exit code for the LDAP compare operations.

Default: false

`-n | --dry-run`

Show what would be done but do not perform any operation.

Default: false

`-S | --scriptFriendly`

Use script-friendly mode.

Default: false

`-Y | --proxyAs {authzID}`

Use the proxied authorization control with the given authorization ID.

LDAP connection options:

-D | --bindDn {bindDN}

DN to use to bind to the server.

Default:

-E | --reportAuthzId

Use the authorization identity control.

Default: false

-h | --hostname {host}

The fully-qualified directory server host name that will be used when generating self-signed certificates for LDAP SSL/StartTLS, the administration connector, and replication.

Default: localhost.localdomain

-j | --bindPasswordFile {bindPasswordFile}

Bind password file.

-K | --keyStorePath {keyStorePath}

Certificate key store path.

-N | --certNickname {nickname}

Nickname of the certificate that the server should use when accepting SSL-based connections or performing StartTLS negotiation.

-o | --saslOption {name=value}

SASL bind options.

-p | --port {port}

Directory server port number.

Default: 389

-P | --trustStorePath {trustStorePath}

Certificate trust store path.

-q | --useStartTls

Use StartTLS to secure communication with the server.

Default: false

-T | --trustStorePassword {trustStorePassword}

Certificate trust store PIN.

-u | --keyStorePasswordFile {keyStorePasswordFile}

Certificate key store PIN file. A PIN is required when you specify to use an existing certificate as server certificate.

-U | --trustStorePasswordFile {path}

Certificate trust store PIN file.

--usePasswordPolicyControl

Use the password policy request control.

Default: false

-w | --bindPassword {bindPassword}

Password to use to bind to the server. Use -w - to ensure that the command prompts for the password, rather than entering the password as a command argument.

-W | --keyStorePassword {keyStorePassword}

Certificate key store PIN. A PIN is required when you specify to use an existing certificate as server certificate.

-X | --trustAll

Trust all server SSL certificates.

Default: false

-Z | --useSsl

Use SSL for secure communication with the server.

Default: false

Utility input/output options:

--noPropertiesFile

No properties file will be used to get default command line argument values.

Default: false

--propertiesFilePath {propertiesFilePath}

Path to the file containing default property values used for command line arguments.

-v | --verbose

Use verbose mode.

Default: false

General options:

-V | --version

Display Directory Server version information.

Default: false

-H | --help

Display this usage information.

Default: false

Exit Codes

0

The command completed successfully.

5

The LDAP compare operation did not match.

6

The `-m` option was used, and the LDAP compare operation did match.

ldap-error

An LDAP error occurred while processing the operation.

LDAP result codes are described in RFC 4511. Also see the additional information for details.

89

An error occurred while parsing the command-line arguments.

Files

You can use `~/.opendj/tools.properties` to set the defaults for bind DN, host name, and port number as in the following example.

```
hostname=directory.example.com
port=1389
bindDN=uid=kvaughan,ou=People,dc=example,dc=com

ldapcompare.port=1389
ldapdelete.port=1389
ldapmodify.port=1389
ldappasswordmodify.port=1389
ldapsearch.port=1389
```

Examples

The following examples demonstrate comparing Babs Jensen's UID.

The following example uses a matching UID value.

```
$ ldapcompare -p 1389 uid:bjensen uid=bjensen,ou=people,dc=example,dc=com
Comparing type uid with value bjensen in entry
uid=bjensen,ou=people,dc=example,dc=com
Compare operation returned true for entry
uid=bjensen,ou=people,dc=example,dc=com
```

The following example uses a UID value that does not match.

```
$ ldapcompare -p 1389 uid:beavis uid=bjensen,ou=people,dc=example,dc=com
Comparing type uid with value beavis in entry
uid=bjensen,ou=people,dc=example,dc=com
Compare operation returned false for entry
uid=bjensen,ou=people,dc=example,dc=com
```

Name

ldapdelete — perform LDAP delete operations

Synopsis

`ldapdelete` [DN]

Description

This utility can be used to perform LDAP delete operations in the Directory Server.

If standard input is used to specify entries to remove, end your input with EOF (Ctrl+D on UNIX, Ctrl +Z on Windows).

Options

The **ldapdelete** command takes the following options:

Command options:

-c | --continueOnError

Continue processing even if there are errors.

Default: false

--connectTimeout {timeout}

Maximum length of time (in milliseconds) that can be taken to establish a connection. Use '0' to specify no time out.

Default: 30000

-J | --control {controloid[:criticality[:value|::b64value|:<filePath]]}

Use a request control with the provided information.

For some *controloid* values, you can replace object identifiers with user-friendly strings. The strings are listed here in lower case, but the case is not important. You can use camelCase if you prefer, for example.

Assertion
LdapAssertion

Assertion Request Control, Object Identifier: 1.3.6.1.1.12

AccountUsable
AccountUsability

Account Usability Request Control, Object Identifier: 1.3.6.1.4.1.42.2.27.9.5.8

`AuthzId`
`AuthorizationIdentity`

Authorization Identity Request Control, Object Identifier: 2.16.840.1.113730.3.4.16

`Csn`
`ChangeNumber`
`ChangeSequenceNumber`

Change Sequence Number Request Control, Object Identifier: 1.3.6.1.4.1.42.2.27.9.5.9

This is an internal OpenDJ control.

`EffectiveRights`
`GetEffectiveRights`

Get Effective Rights Request Control, Object Identifier: 1.3.6.1.4.1.42.2.27.9.5.2

`ManageDsaIt`

Manage DSAIT Request Control, Object Identifier: 2.16.840.1.113730.3.4.2

`Noop`
`No-Op`

No-Op Request Control, Object Identifier: 1.3.6.1.4.1.4203.1.10.2

`PwPolicy`
`PasswordPolicy`

Password Policy Request Control, Object Identifier: 1.3.6.1.4.1.42.2.27.8.5.1

`PermissiveModify`

Permissive Modify Request Control, Object Identifier: 1.2.840.113556.1.4.1413

`PSearch`
`PersistentSearch`

Persistent Search Request Control, Object Identifier: 2.16.840.1.113730.3.4.3

`PostRead`

Post Read Request Control, Object Identifier: 1.3.6.1.1.13.2

`PreRead`

Pre Read Request Control, Object Identifier: 1.3.6.1.1.13.1

`ProxiedAuthV1`

Proxied Authorization Request Control V1, Object Identifier: 2.16.840.1.113730.3.4.12

`ProxiedAuth`
`ProxiedAuthV2`

Proxied Authorization Request Control V2, Object Identifier: 2.16.840.1.113730.3.4.18

`RealAttrsOnly`
`RealAttributesOnly`

Real Attributes Only Request Control, Object Identifier: 2.16.840.1.113730.3.4.17

`TreeDelete`
`SubTreeDelete`

Subtree Delete Request Control, Object Identifier: 1.2.840.113556.1.4.805

`Sort`
`ServerSideSort`

Server Side Sort Request Control, Object Identifier: 1.2.840.113556.1.4.473

`PagedResults`
`SimplePagedResults`

Simple Paged Results Control, Object Identifier: 1.2.840.113556.1.4.319

`SubEntries`

Sub-Entries Request Control, Object Identifier: 1.3.6.1.4.1.4203.1.10.1

`TxnId`
`TransactionId`

Transaction ID Control, Object Identifier: 1.3.6.1.4.1.36733.2.1.5.1

This is an internal ForgeRock control.

`VirtualAttrsOnly`
`VirtualAttributesOnly`

Virtual Attributes Only Request Control, Object Identifier: 2.16.840.1.113730.3.4.19

`Vlv`
`VirtualListView`

Virtual List View Request Control, Object Identifier: 2.16.840.1.113730.3.4.9

`-n | --dry-run`

Show what would be done but do not perform any operation.

Default: false

`-x | --deleteSubtree`

Delete the specified entry and all entries below it.

Default: false

LDAP connection options:

`-D | --bindDn {bindDN}`

DN to use to bind to the server.

Default:

-E | --reportAuthzId

Use the authorization identity control.

Default: false

-h | --hostname {host}

The fully-qualified directory server host name that will be used when generating self-signed certificates for LDAP SSL/StartTLS, the administration connector, and replication.

Default: localhost.localdomain

-j | --bindPasswordFile {bindPasswordFile}

Bind password file.

-K | --keyStorePath {keyStorePath}

Certificate key store path.

-N | --certNickname {nickname}

Nickname of the certificate that the server should use when accepting SSL-based connections or performing StartTLS negotiation.

-o | --saslOption {name=value}

SASL bind options.

-p | --port {port}

Directory server port number.

Default: 389

-P | --trustStorePath {trustStorePath}

Certificate trust store path.

-q | --useStartTls

Use StartTLS to secure communication with the server.

Default: false

-T | --trustStorePassword {trustStorePassword}

Certificate trust store PIN.

-u | --keyStorePasswordFile {keyStorePasswordFile}

Certificate key store PIN file. A PIN is required when you specify to use an existing certificate as server certificate.

-U | --trustStorePasswordFile {path}

Certificate trust store PIN file.

--usePasswordPolicyControl

Use the password policy request control.

Default: false

-w | --bindPassword {bindPassword}

Password to use to bind to the server. Use -w - to ensure that the command prompts for the password, rather than entering the password as a command argument.

-W | --keyStorePassword {keyStorePassword}

Certificate key store PIN. A PIN is required when you specify to use an existing certificate as server certificate.

-X | --trustAll

Trust all server SSL certificates.

Default: false

-Z | --useSsl

Use SSL for secure communication with the server.

Default: false

Utility input/output options:

--noPropertiesFile

No properties file will be used to get default command line argument values.

Default: false

--propertiesFilePath {propertiesFilePath}

Path to the file containing default property values used for command line arguments.

-v | --verbose

Use verbose mode.

Default: false

General options:

-V | --version

Display Directory Server version information.

Default: false

-H | --help

Display this usage information.

Default: false

Exit Codes

0

The command completed successfully.

ldap-error

An LDAP error occurred while processing the operation.

LDAP result codes are described in RFC 4511. Also see the additional information for details.

89

An error occurred while parsing the command-line arguments.

Files

You can use `~/.opendj/tools.properties` to set the defaults for bind DN, host name, and port number as in the following example.

```
hostname=directory.example.com
port=1389
bindDN=uid=kvaughan,ou=People,dc=example,dc=com

ldapcompare.port=1389
ldapdelete.port=1389
ldapmodify.port=1389
ldappasswordmodify.port=1389
ldapsearch.port=1389
```

Examples

The following command deletes a user entry from the directory.

```
$ ldapdelete -p 1389 -D "cn=Directory Manager" -w password \
 uid=bjensen,ou=people,dc=example,dc=com
Processing DELETE request for uid=bjensen,ou=people,dc=example,dc=com
DELETE operation successful for DN uid=bjensen,ou=people,dc=example,dc=com
```

The following command deletes the `ou=Groups` entry and all entries underneath `ou=Groups`.

```
$ ldapdelete -p 1389 -D "cn=Directory Manager" -w password -x \
 ou=groups,dc=example,dc=com
Processing DELETE request for ou=groups,dc=example,dc=com
DELETE operation successful for DN ou=groups,dc=example,dc=com
```

Name

ldapmodify — perform LDAP modify, add, delete, mod DN operations

Synopsis

`ldapmodify` [changes_files ...]

Description

This utility can be used to perform LDAP modify, add, delete, and modify DN operations in the Directory Server. When not using file(s) to specify modifications, end your input with EOF (Ctrl+D on UNIX, Ctrl+Z on Windows).

Options

The **ldapmodify** command takes the following options:

Command options:

`--assertionFilter {filter}`

Use the LDAP assertion control with the provided filter.

`-c | --continueOnError`

Continue processing even if there are errors.

Default: false

`--connectTimeout {timeout}`

Maximum length of time (in milliseconds) that can be taken to establish a connection. Use '0' to specify no time out.

Default: 30000

`-J | --control {controloid[:criticality[:value|::b64value|:<filePath]]}`

Use a request control with the provided information.

For some *`controloid`* values, you can replace object identifiers with user-friendly strings. The strings are listed here in lower case, but the case is not important. You can use camelCase if you prefer, for example.

`Assertion`
`LdapAssertion`

Assertion Request Control, Object Identifier: 1.3.6.1.1.12

`AccountUsable`
`AccountUsability`

Account Usability Request Control, Object Identifier: 1.3.6.1.4.1.42.2.27.9.5.8

`AuthzId`
`AuthorizationIdentity`

Authorization Identity Request Control, Object Identifier: 2.16.840.1.113730.3.4.16

`Csn`
`ChangeNumber`
`ChangeSequenceNumber`

Change Sequence Number Request Control, Object Identifier: 1.3.6.1.4.1.42.2.27.9.5.9

This is an internal OpenDJ control.

`EffectiveRights`
`GetEffectiveRights`

Get Effective Rights Request Control, Object Identifier: 1.3.6.1.4.1.42.2.27.9.5.2

`ManageDsaIt`

Manage DSAIT Request Control, Object Identifier: 2.16.840.1.113730.3.4.2

`Noop`
`No-Op`

No-Op Request Control, Object Identifier: 1.3.6.1.4.1.4203.1.10.2

`PwPolicy`
`PasswordPolicy`

Password Policy Request Control, Object Identifier: 1.3.6.1.4.1.42.2.27.8.5.1

`PermissiveModify`

Permissive Modify Request Control, Object Identifier: 1.2.840.113556.1.4.1413

`PSearch`
`PersistentSearch`

Persistent Search Request Control, Object Identifier: 2.16.840.1.113730.3.4.3

`PostRead`

Post Read Request Control, Object Identifier: 1.3.6.1.1.13.2

`PreRead`

Pre Read Request Control, Object Identifier: 1.3.6.1.1.13.1

`ProxiedAuthV1`

Proxied Authorization Request Control V1, Object Identifier: 2.16.840.1.113730.3.4.12

`ProxiedAuth`
`ProxiedAuthV2`

Proxied Authorization Request Control V2, Object Identifier: 2.16.840.1.113730.3.4.18

`RealAttrsOnly`
`RealAttributesOnly`

Real Attributes Only Request Control, Object Identifier: 2.16.840.1.113730.3.4.17

`TreeDelete`
`SubTreeDelete`

Subtree Delete Request Control, Object Identifier: 1.2.840.113556.1.4.805

`Sort`
`ServerSideSort`

Server Side Sort Request Control, Object Identifier: 1.2.840.113556.1.4.473

`PagedResults`
`SimplePagedResults`

Simple Paged Results Control, Object Identifier: 1.2.840.113556.1.4.319

`SubEntries`

Sub-Entries Request Control, Object Identifier: 1.3.6.1.4.1.4203.1.10.1

`TxnId`
`TransactionId`

Transaction ID Control, Object Identifier: 1.3.6.1.4.1.36733.2.1.5.1

This is an internal ForgeRock control.

`VirtualAttrsOnly`
`VirtualAttributesOnly`

Virtual Attributes Only Request Control, Object Identifier: 2.16.840.1.113730.3.4.19

`Vlv`
`VirtualListView`

Virtual List View Request Control, Object Identifier: 2.16.840.1.113730.3.4.9

`-n | --dry-run`

Show what would be done but do not perform any operation.

Default: false

`--postReadAttributes {attrList}`

Use the LDAP ReadEntry post-read control.

--preReadAttributes {attrList}

Use the LDAP ReadEntry pre-read control.

-Y | --proxyAs {authzID}

Use the proxied authorization control with the given authorization ID.

LDAP connection options:

-D | --bindDn {bindDN}

DN to use to bind to the server.

Default:

-E | --reportAuthzId

Use the authorization identity control.

Default: false

-h | --hostname {host}

The fully-qualified directory server host name that will be used when generating self-signed certificates for LDAP SSL/StartTLS, the administration connector, and replication.

Default: localhost.localdomain

-j | --bindPasswordFile {bindPasswordFile}

Bind password file.

-K | --keyStorePath {keyStorePath}

Certificate key store path.

-N | --certNickname {nickname}

Nickname of the certificate that the server should use when accepting SSL-based connections or performing StartTLS negotiation.

-o | --saslOption {name=value}

SASL bind options.

-p | --port {port}

Directory server port number.

Default: 389

-P | --trustStorePath {trustStorePath}

Certificate trust store path.

-q | --useStartTls

Use StartTLS to secure communication with the server.

Default: false

-T | --trustStorePassword {trustStorePassword}

Certificate trust store PIN.

-u | --keyStorePasswordFile {keyStorePasswordFile}

Certificate key store PIN file. A PIN is required when you specify to use an existing certificate as server certificate.

-U | --trustStorePasswordFile {path}

Certificate trust store PIN file.

--usePasswordPolicyControl

Use the password policy request control.

Default: false

-w | --bindPassword {bindPassword}

Password to use to bind to the server. Use -w - to ensure that the command prompts for the password, rather than entering the password as a command argument.

-W | --keyStorePassword {keyStorePassword}

Certificate key store PIN. A PIN is required when you specify to use an existing certificate as server certificate.

-X | --trustAll

Trust all server SSL certificates.

Default: false

-Z | --useSsl

Use SSL for secure communication with the server.

Default: false

Utility input/output options:

--noPropertiesFile

No properties file will be used to get default command line argument values.

Default: false

--propertiesFilePath {propertiesFilePath}

Path to the file containing default property values used for command line arguments.

-v | --verbose

Use verbose mode.

Default: false

General options:

-V | --version

Display Directory Server version information.

Default: false

-H | --help

Display this usage information.

Default: false

Exit Codes

0

The command completed successfully.

ldap-error

An LDAP error occurred while processing the operation.

LDAP result codes are described in RFC 4511. Also see the additional information for details.

89

An error occurred while parsing the command-line arguments.

Files

You can use `~/.opendj/tools.properties` to set the defaults for bind DN, host name, and port number as in the following example.

```
hostname=directory.example.com
port=1389
bindDN=uid=kvaughan,ou=People,dc=example,dc=com

ldapcompare.port=1389
ldapdelete.port=1389
ldapmodify.port=1389
ldappasswordmodify.port=1389
ldapsearch.port=1389
```

Examples

The following example demonstrates use of the command to add an entry to the directory:

```
$ cat newuser.ldif
dn: uid=newuser,ou=People,dc=example,dc=com
uid: newuser
facsimileTelephoneNumber: +1 408 555 1213
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: top
givenName: New
cn: New User
cn: Real Name
telephoneNumber: +1 408 555 1212
sn: Jensen
roomNumber: 1234
homeDirectory: /home/newuser
uidNumber: 10389
mail: newuser@example.com
l: South Pole
ou: Product Development
ou: People
gidNumber: 10636

$ ldapmodify -p 1389 -D uid=kvaughan,ou=people,dc=example,dc=com -w bribery newuser.ldif
Processing ADD request for uid=newuser,ou=People,dc=example,dc=com
ADD operation successful for DN uid=newuser,ou=People,dc=example,dc=com
```

The following listing shows a UNIX shell script that adds a user entry:

```
#!/bin/sh
#
# Add a new user with the ldapmodify utility.
#

usage(){
        echo "Usage: $0 uid firstname lastname"
        exit 1
}
[[ $# -lt 3 ]] && usage

LDAPMODIFY=/path/to/opendj/bin/ldapmodify
HOST=opendj.example.com
PORT=1389
ADMIN=uid=kvaughan,ou=people,dc=example,dc=com
PWD=bribery

$LDAPMODIFY -h $HOST -p $PORT -D $ADMIN -w $PWD <<EOF
dn: uid=$1,ou=people,dc=example,dc=com
uid: $1
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
cn: $2 $3
givenName: $2
sn: $3
mail: $1@example.com
EOF
```

The following example demonstrates adding a Description attribute to the new user's entry:

```
$ cat newdesc.ldif
dn: uid=newuser,ou=People,dc=example,dc=com
changetype: modify
add: description
description: A new user's entry

$ ldapmodify -p 1389 -D uid=kvaughan,ou=people,dc=example,dc=com -w bribery newdesc.ldif
Processing MODIFY request for uid=newuser,ou=People,dc=example,dc=com
MODIFY operation successful for DN uid=newuser,ou=People,dc=example,dc=com
```

The following example demonstrates changing the Description attribute for the new user's entry:

```
$ cat moddesc.ldif
dn: uid=newuser,ou=People,dc=example,dc=com
changetype: modify
replace: description
description: Another description

$ ldapmodify -p 1389 -D uid=kvaughan,ou=people,dc=example,dc=com -w bribery moddesc.ldif
Processing MODIFY request for uid=newuser,ou=People,dc=example,dc=com
MODIFY operation successful for DN uid=newuser,ou=People,dc=example,dc=com
```

The following example demonstrates deleting the new user's entry:

```
$ cat deluser.ldif
dn: uid=newuser,ou=People,dc=example,dc=com
changetype: delete

$ ldapmodify -p 1389 -D uid=kvaughan,ou=people,dc=example,dc=com -w bribery deluser.ldif
Processing DELETE request for uid=newuser,ou=People,dc=example,dc=com
DELETE operation successful for DN uid=newuser,ou=People,dc=example,dc=com
```

Name

ldappasswordmodify — perform LDAP password modifications

Synopsis

```
ldappasswordmodify
```

Description

This utility can be used to perform LDAP password modify operations in the Directory Server.

Options

The **ldappasswordmodify** command takes the following options:

Command options:

-a | --authzId {authzID}

Authorization ID for the user entry whose password should be changed. The authorization ID is a string having either the prefix "dn:" followed by the user's distinguished name, or the prefix "u:" followed by a user identifier that depends on the identity mapping used to match the user identifier to an entry in the directory. Examples include "dn:uid=bjensen,ou=People,dc=example,dc=com", and, if we assume that "bjensen" is mapped to Barbara Jensen's entry, "u:bjensen".

-c | --currentPassword {currentPassword}

Current password for the target user.

-C | --currentPasswordFile {file}

Path to a file containing the current password for the target user.

--connectTimeout {timeout}

Maximum length of time (in milliseconds) that can be taken to establish a connection. Use '0' to specify no time out.

Default: 30000

-F | --newPasswordFile {file}

Path to a file containing the new password to provide for the target user.

-J | --control {controloid[:criticality[:value|::b64value|:<filePath]]}

Use a request control with the provided information.

For some *controloid* values, you can replace object identifiers with user-friendly strings. The strings are listed here in lower case, but the case is not important. You can use camelCase if you prefer, for example.

Assertion
LdapAssertion

Assertion Request Control, Object Identifier: 1.3.6.1.1.12

AccountUsable
AccountUsability

Account Usability Request Control, Object Identifier: 1.3.6.1.4.1.42.2.27.9.5.8

AuthzId
AuthorizationIdentity

Authorization Identity Request Control, Object Identifier: 2.16.840.1.113730.3.4.16

Csn
ChangeNumber
ChangeSequenceNumber

Change Sequence Number Request Control, Object Identifier: 1.3.6.1.4.1.42.2.27.9.5.9

This is an internal OpenDJ control.

EffectiveRights
GetEffectiveRights

Get Effective Rights Request Control, Object Identifier: 1.3.6.1.4.1.42.2.27.9.5.2

ManageDsaIt

Manage DSAIT Request Control, Object Identifier: 2.16.840.1.113730.3.4.2

Noop
No-Op

No-Op Request Control, Object Identifier: 1.3.6.1.4.1.4203.1.10.2

PwPolicy
PasswordPolicy

Password Policy Request Control, Object Identifier: 1.3.6.1.4.1.42.2.27.8.5.1

PermissiveModify

Permissive Modify Request Control, Object Identifier: 1.2.840.113556.1.4.1413

PSearch
PersistentSearch

Persistent Search Request Control, Object Identifier: 2.16.840.1.113730.3.4.3

`PostRead`

Post Read Request Control, Object Identifier: 1.3.6.1.1.13.2

`PreRead`

Pre Read Request Control, Object Identifier: 1.3.6.1.1.13.1

`ProxiedAuthV1`

Proxied Authorization Request Control V1, Object Identifier: 2.16.840.1.113730.3.4.12

`ProxiedAuth`
`ProxiedAuthV2`

Proxied Authorization Request Control V2, Object Identifier: 2.16.840.1.113730.3.4.18

`RealAttrsOnly`
`RealAttributesOnly`

Real Attributes Only Request Control, Object Identifier: 2.16.840.1.113730.3.4.17

`TreeDelete`
`SubTreeDelete`

Subtree Delete Request Control, Object Identifier: 1.2.840.113556.1.4.805

`Sort`
`ServerSideSort`

Server Side Sort Request Control, Object Identifier: 1.2.840.113556.1.4.473

`PagedResults`
`SimplePagedResults`

Simple Paged Results Control, Object Identifier: 1.2.840.113556.1.4.319

`SubEntries`

Sub-Entries Request Control, Object Identifier: 1.3.6.1.4.1.4203.1.10.1

`TxnId`
`TransactionId`

Transaction ID Control, Object Identifier: 1.3.6.1.4.1.36733.2.1.5.1

This is an internal ForgeRock control.

`VirtualAttrsOnly`
`VirtualAttributesOnly`

Virtual Attributes Only Request Control, Object Identifier: 2.16.840.1.113730.3.4.19

`Vlv`
`VirtualListView`

Virtual List View Request Control, Object Identifier: 2.16.840.1.113730.3.4.9

-n | --newPassword {newPassword}

New password to provide for the target user.

LDAP connection options:

-D | --bindDn {bindDN}

DN to use to bind to the server.

Default:

-E | --reportAuthzId

Use the authorization identity control.

Default: false

-h | --hostname {host}

The fully-qualified directory server host name that will be used when generating self-signed certificates for LDAP SSL/StartTLS, the administration connector, and replication.

Default: localhost.localdomain

-j | --bindPasswordFile {bindPasswordFile}

Bind password file.

-K | --keyStorePath {keyStorePath}

Certificate key store path.

-N | --certNickname {nickname}

Nickname of the certificate that the server should use when accepting SSL-based connections or performing StartTLS negotiation.

-o | --saslOption {name=value}

SASL bind options.

-p | --port {port}

Directory server port number.

Default: 389

-P | --trustStorePath {trustStorePath}

Certificate trust store path.

-q | --useStartTls

Use StartTLS to secure communication with the server.

Default: false

-T | --trustStorePassword {trustStorePassword}

Certificate trust store PIN.

-u | --keyStorePasswordFile {keyStorePasswordFile}

Certificate key store PIN file. A PIN is required when you specify to use an existing certificate as server certificate.

-U | --trustStorePasswordFile {path}

Certificate trust store PIN file.

--usePasswordPolicyControl

Use the password policy request control.

Default: false

-w | --bindPassword {bindPassword}

Password to use to bind to the server. Use -w - to ensure that the command prompts for the password, rather than entering the password as a command argument.

-W | --keyStorePassword {keyStorePassword}

Certificate key store PIN. A PIN is required when you specify to use an existing certificate as server certificate.

-X | --trustAll

Trust all server SSL certificates.

Default: false

-Z | --useSsl

Use SSL for secure communication with the server.

Default: false

Utility input/output options:

--noPropertiesFile

No properties file will be used to get default command line argument values.

Default: false

--propertiesFilePath {propertiesFilePath}

Path to the file containing default property values used for command line arguments.

-v | --verbose

Use verbose mode.

Default: false

General options:

-V | --version

Display Directory Server version information.

Default: false

-H | --help

Display this usage information.

Default: false

Exit Codes

0

The command completed successfully.

ldap-error

An LDAP error occurred while processing the operation.

LDAP result codes are described in RFC 4511. Also see the additional information for details.

89

An error occurred while parsing the command-line arguments.

Files

You can use `~/.opendj/tools.properties` to set the defaults for bind DN, host name, and port number as in the following example.

```
hostname=directory.example.com
port=1389
bindDN=uid=kvaughan,ou=People,dc=example,dc=com

ldapcompare.port=1389
ldapdelete.port=1389
ldapmodify.port=1389
ldappasswordmodify.port=1389
ldapsearch.port=1389
```

Examples

The following example demonstrates a user changing their own password.

```
$ cat /tmp/currpwd.txt /tmp/newpwd.txt
bribery
secret12

$ ldappasswordmodify -p 1389 -C /tmp/currpwd.txt --newPasswordFile /tmp/newpwd.txt \
 -D uid=kvaughan,ou=people,dc=example,dc=com -w bribery
The LDAP password modify operation was successful
```

Name

ldapsearch — perform LDAP search operations

Synopsis

`ldapsearch` filter [attributes ...]

Description

This utility can be used to perform LDAP search operations in the Directory Server.

Options

The **ldapsearch** command takes the following options:

Command options:

-a | --dereferencePolicy {dereferencePolicy}

Alias dereference policy ('never', 'always', 'search', or 'find').

Default: never

-A | --typesOnly

Only retrieve attribute names but not their values.

Default: false

--assertionFilter {filter}

Use the LDAP assertion control with the provided filter.

-b | --baseDn {baseDN}

Search base DN.

-c | --continueOnError

Continue processing even if there are errors.

Default: false

-C | --persistentSearch ps[:changetype[:changesonly[:entrychgcontrols]]]

Use the persistent search control.

A persistent search allows the client to continue receiving new results whenever changes are made to data that is in the scope of the search, thus using the search as a form of change notification.

The optional `changetype` setting defines the kinds of updates that result in notification. If you do not set the `changetype`, the default behavior is to send notifications for all updates.

`add`

Send notifications for LDAP add operations.

`del`
`delete`

Send notifications for LDAP delete operations.

`mod`
`modify`

Send notifications for LDAP modify operations.

`moddn`
`modrdn`
`modifydn`

Send notifications for LDAP modify DN (rename and move) operations.

`all`
`any`

Send notifications for all LDAP update operations.

The optional `changesonly` setting defines whether the server returns existing entries as well as changes.

`true`

Do not return existing entries, but instead only notifications about changes.

This is the default setting.

`false`

Also return existing entries.

The optional `entrychgcontrols` setting defines whether the server returns an Entry Change Notification control with each entry notification. The Entry Change Notification control provides additional information about the change that caused the entry to be returned by the search. In particular, it indicates the change type, the change number if available, and the previous DN if the change type was a modify DN operation.

`true`

Do request the Entry Change Notification control.

This is the default setting.

`false`

Do not request the Entry Change Notification control.

`--connectTimeout {timeout}`

Maximum length of time (in milliseconds) that can be taken to establish a connection. Use '0' to specify no time out.

Default: 30000

`--countEntries`

Count the number of entries returned by the server.

Default: false

`-e | --getEffectiveRightsAttribute {attribute}`

Specifies geteffectiverights control specific attribute list.

`-g | --getEffectiveRightsAuthzId {authzID}`

Use geteffectiverights control with the provided authzid.

`-G | --virtualListView {before:after:index:count | before:after:value}`

Use the virtual list view control to retrieve the specified results page.

`-J | --control {controloid[:criticality[:value|::b64value|:<filePath]]}`

Use a request control with the provided information.

For some *`controloid`* values, you can replace object identifiers with user-friendly strings. The strings are listed here in lower case, but the case is not important. You can use camelCase if you prefer, for example.

`Assertion`
`LdapAssertion`

Assertion Request Control, Object Identifier: 1.3.6.1.1.12

`AccountUsable`
`AccountUsability`

Account Usability Request Control, Object Identifier: 1.3.6.1.4.1.42.2.27.9.5.8

`AuthzId`
`AuthorizationIdentity`

Authorization Identity Request Control, Object Identifier: 2.16.840.1.113730.3.4.16

`Csn`
`ChangeNumber`
`ChangeSequenceNumber`

Change Sequence Number Request Control, Object Identifier: 1.3.6.1.4.1.42.2.27.9.5.9

This is an internal OpenDJ control.

`EffectiveRights`
`GetEffectiveRights`

Get Effective Rights Request Control, Object Identifier: 1.3.6.1.4.1.42.2.27.9.5.2

`ManageDsaIt`

Manage DSAIT Request Control, Object Identifier: 2.16.840.1.113730.3.4.2

`Noop`
`No-Op`

No-Op Request Control, Object Identifier: 1.3.6.1.4.1.4203.1.10.2

`PwPolicy`
`PasswordPolicy`

Password Policy Request Control, Object Identifier: 1.3.6.1.4.1.42.2.27.8.5.1

`PermissiveModify`

Permissive Modify Request Control, Object Identifier: 1.2.840.113556.1.4.1413

`PSearch`
`PersistentSearch`

Persistent Search Request Control, Object Identifier: 2.16.840.1.113730.3.4.3

`PostRead`

Post Read Request Control, Object Identifier: 1.3.6.1.1.13.2

`PreRead`

Pre Read Request Control, Object Identifier: 1.3.6.1.1.13.1

`ProxiedAuthV1`

Proxied Authorization Request Control V1, Object Identifier: 2.16.840.1.113730.3.4.12

`ProxiedAuth`
`ProxiedAuthV2`

Proxied Authorization Request Control V2, Object Identifier: 2.16.840.1.113730.3.4.18

`RealAttrsOnly`
`RealAttributesOnly`

Real Attributes Only Request Control, Object Identifier: 2.16.840.1.113730.3.4.17

`TreeDelete`
`SubTreeDelete`

Subtree Delete Request Control, Object Identifier: 1.2.840.113556.1.4.805

`Sort`
`ServerSideSort`

Server Side Sort Request Control, Object Identifier: 1.2.840.113556.1.4.473

`PagedResults`
`SimplePagedResults`

Simple Paged Results Control, Object Identifier: 1.2.840.113556.1.4.319

`SubEntries`

Sub-Entries Request Control, Object Identifier: 1.3.6.1.4.1.4203.1.10.1

`TxnId`
`TransactionId`

Transaction ID Control, Object Identifier: 1.3.6.1.4.1.36733.2.1.5.1

This is an internal ForgeRock control.

`VirtualAttrsOnly`
`VirtualAttributesOnly`

Virtual Attributes Only Request Control, Object Identifier: 2.16.840.1.113730.3.4.19

`Vlv`
`VirtualListView`

Virtual List View Request Control, Object Identifier: 2.16.840.1.113730.3.4.9

`-l | --timeLimit {timeLimit}`

Maximum length of time in seconds to allow for the search.

Default: 0

`--matchedValuesFilter {filter}`

Use the LDAP matched values control with the provided filter.

`-n | --dry-run`

Show what would be done but do not perform any operation.

Default: false

`-s | --searchScope {searchScope}`

Search scope ('base', 'one', 'sub', or 'subordinates'). Note: 'subordinates' is an LDAP extension that might not work with all LDAP servers.

Default: sub

`-S | --sortOrder {sortOrder}`

Sort the results using the provided sort order.

--simplePageSize {numEntries}

Use the simple paged results control with the given page size.

Default: 1000

--subEntries

Use subentries control to specify that subentries are visible and normal entries are not.

Default: false

-Y | --proxyAs {authzID}

Use the proxied authorization control with the given authorization ID.

-z | --sizeLimit {sizeLimit}

Maximum number of entries to return from the search.

Default: 0

LDAP connection options:

-D | --bindDn {bindDN}

DN to use to bind to the server.

Default:

-E | --reportAuthzId

Use the authorization identity control.

Default: false

-h | --hostname {host}

The fully-qualified directory server host name that will be used when generating self-signed certificates for LDAP SSL/StartTLS, the administration connector, and replication.

Default: localhost.localdomain

-j | --bindPasswordFile {bindPasswordFile}

Bind password file.

-K | --keyStorePath {keyStorePath}

Certificate key store path.

-N | --certNickname {nickname}

Nickname of the certificate that the server should use when accepting SSL-based connections or performing StartTLS negotiation.

-o | --saslOption {name=value}

SASL bind options.

-p | --port {port}

Directory server port number.

Default: 389

-P | --trustStorePath {trustStorePath}

Certificate trust store path.

-q | --useStartTls

Use StartTLS to secure communication with the server.

Default: false

-T | --trustStorePassword {trustStorePassword}

Certificate trust store PIN.

-u | --keyStorePasswordFile {keyStorePasswordFile}

Certificate key store PIN file. A PIN is required when you specify to use an existing certificate as server certificate.

-U | --trustStorePasswordFile {path}

Certificate trust store PIN file.

--usePasswordPolicyControl

Use the password policy request control.

Default: false

-w | --bindPassword {bindPassword}

Password to use to bind to the server. Use -w - to ensure that the command prompts for the password, rather than entering the password as a command argument.

-W | --keyStorePassword {keyStorePassword}

Certificate key store PIN. A PIN is required when you specify to use an existing certificate as server certificate.

-X | --trustAll

Trust all server SSL certificates.

Default: false

-Z | --useSsl

Use SSL for secure communication with the server.

Default: false

Utility input/output options:

--noPropertiesFile

No properties file will be used to get default command line argument values.

Default: false

--propertiesFilePath {propertiesFilePath}

Path to the file containing default property values used for command line arguments.

-t | --wrapColumn {wrapColumn}

Maximum length of an output line (0 for no wrapping).

Default: 0

-v | --verbose

Use verbose mode.

Default: false

General options:

-V | --version

Display Directory Server version information.

Default: false

-H | --help

Display this usage information.

Default: false

Filters

The filter argument is a string representation of an LDAP search filter as in `(cn=Babs Jensen)`, `(&(objectClass=Person)(|(sn=Jensen)(cn=Babs J*)))`, or `(cn:caseExactMatch:=Fred Flintstone)`.

Attributes

The optional attribute list specifies the attributes to return in the entries found by the search. In addition to identifying attributes by name such as `cn sn mail` and so forth, you can use the following notations, too.

*

Return all user attributes such as `cn`, `sn`, and `mail`.

`+`

Return all operational attributes such as `etag` and `pwdPolicySubentry`.

`@objectclass`

Return all attributes of the specified object class, where *`objectclass`* is one of the object classes on the entries returned by the search.

`1.1`

Return no attributes, only the DNs of matching entries.

Exit Codes

0

The command completed successfully.

ldap-error

An LDAP error occurred while processing the operation.

LDAP result codes are described in RFC 4511. Also see the additional information for details.

89

An error occurred while parsing the command-line arguments.

Files

You can use `~/.opendj/tools.properties` to set the defaults for bind DN, host name, and port number as in the following example.

```
hostname=directory.example.com
port=1389
bindDN=uid=kvaughan,ou=People,dc=example,dc=com

ldapcompare.port=1389
ldapdelete.port=1389
ldapmodify.port=1389
ldappasswordmodify.port=1389
ldapsearch.port=1389
```

Examples

The following example searches for entries with UID containing `jensen`, returning only DNs and uid values:

```
$ ldapsearch -p 1389 -b dc=example,dc=com "(uid=*jensen*)" uid
dn: uid=ajensen,ou=People,dc=example,dc=com
uid: ajensen

dn: uid=bjensen,ou=People,dc=example,dc=com
uid: bjensen

dn: uid=gjensen,ou=People,dc=example,dc=com
uid: gjensen

dn: uid=jjensen,ou=People,dc=example,dc=com
uid: jjensen

dn: uid=kjensen,ou=People,dc=example,dc=com
uid: kjensen

dn: uid=rjensen,ou=People,dc=example,dc=com
uid: rjensen

dn: uid=tjensen,ou=People,dc=example,dc=com
uid: tjensen


Result Code:  0 (Success)
```

You can also use `@objectclass` notation in the attribute list to return the attributes of a particular object class. The following example shows how to return attributes of the `inetOrgPerson` object class:

```
$ ldapsearch -p 1389 -b dc=example,dc=com "(uid=bjensen)" @inetorgperson
dn: uid=bjensen,ou=People,dc=example,dc=com
givenName: Barbara
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: top
uid: bjensen
cn: Barbara Jensen
cn: Babs Jensen
telephoneNumber: +1 408 555 1862
sn: Jensen
roomNumber: 0209
mail: bjensen@example.com
l: San Francisco
ou: Product Development
ou: People
facsimileTelephoneNumber: +1 408 555 1992
```

You can use `+` in the attribute list to return all operational attributes, as in the following example:

```
$ ldapsearch -p 1389 -b dc=example,dc=com "(uid=bjensen)" +
dn: uid=bjensen,ou=People,dc=example,dc=com
numSubordinates: 0
structuralObjectClass: inetOrgPerson
etag: 0000000073c29972
subschemaSubentry: cn=schema
hasSubordinates: false
entryDN: uid=bjensen,ou=people,dc=example,dc=com
entryUUID: fc252fd9-b982-3ed6-b42a-c76d2546312c
```

Name

ldifdiff — compare small LDIF files

Synopsis

`ldifdiff` source target

Description

This utility can be used to compare two LDIF files and report the differences in LDIF format.

If standard input is used to specify source or target, end your input with EOF (Ctrl+D on UNIX, Ctrl +Z on Windows).

Options

The **ldifdiff** command takes the following options:

Command options:

-B | --excludeBranch {branchDN}

Base DN of a branch to exclude when comparing entries.

-e | --excludeAttribute {attribute}

Attribute to ignore when comparing entries.

-o | --outputLdif {file}

Write differences to {file} instead of stdout.

Default: stdout

Utility input/output options:

-t | --wrapColumn {wrapColumn}

Maximum length of an output line (0 for no wrapping).

Default: 0

General options:

-V | --version

Display Directory Server version information.

Default: false

-H | --help

Display this usage information.

Default: false

Exit Codes

0

No differences were found.

1

Differences were found.

other

An error occurred.

Examples

The following example demonstrates use of the command with two small LDIF files.

```
$ cat /path/to/newuser.ldif
dn: uid=newuser,ou=People,dc=example,dc=com
uid: newuser
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: top
cn: New User
sn: User
ou: People
mail: newuser@example.com
userPassword: changeme

$ cat /path/to/neweruser.ldif
dn: uid=newuser,ou=People,dc=example,dc=com
uid: newuser
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: top
cn: New User
sn: User
ou: People
mail: newuser@example.com
userPassword: secret12
description: A new description.

$ ldifdiff -s /path/to/newuser.ldif -t /path/to/neweruser.ldif
dn: uid=newuser,ou=People,dc=example,dc=com
changetype: modify
add: userPassword
```

```
userPassword: secret12
-
delete: userPassword
userPassword: changeme
-
add: description
description: A new description.
```

Name

ldifmodify — apply LDIF changes to LDIF

Synopsis

`ldifmodify` source_file [changes_files...]

Description

This utility can be used to apply a set of modify, add, and delete operations to entries contained in an LDIF file.

If standard input is used to specify source or changes, end your input with EOF (Ctrl+D on UNIX, Ctrl +Z on Windows).

Options

The **ldifmodify** command takes the following options:

Command options:

`-c | --continueOnError`

Continue processing even if there are errors.

Default: false

`-o | --outputLdif {file}`

Write updated entries to {file} instead of stdout.

Default: stdout

Utility input/output options:

`-t | --wrapColumn {wrapColumn}`

Maximum length of an output line (0 for no wrapping).

Default: 0

General options:

`-V | --version`

Display Directory Server version information.

Default: false

-H | --help

Display this usage information.

Default: false

Exit Codes

0

The command completed successfully.

> 0

An error occurred.

Examples

The following example demonstrates use of the command.

```
$ cat /path/to/newuser.ldif
dn: uid=newuser,ou=People,dc=example,dc=com
uid: newuser
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: top
cn: New User
sn: User
ou: People
mail: newuser@example.com
userPassword: changeme

$ cat /path/to/newdiff.ldif
dn: uid=newuser,ou=People,dc=example,dc=com
changetype: modify
add: userPassword
userPassword: secret12
-
delete: userPassword
userPassword: changeme
-
add: description
description: A new description.

$ ldifmodify -o neweruser.ldif /path/to/newuser.ldif /path/to/newdiff.ldif

$ cat neweruser.ldif
dn: uid=newuser,ou=People,dc=example,dc=com
uid: newuser
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: top
cn: New User
```

```
sn: User
ou: People
mail: newuser@example.com
userPassword: secret12
description: A new description.
```

Name

ldifsearch — search LDIF with LDAP filters

Synopsis

`ldifsearch` source filter [attributes ...]

Description

This utility can be used to perform search operations against entries contained in an LDIF file.

If standard input is used to specify source, end your input with EOF (Ctrl+D on UNIX, Ctrl+Z on Windows).

Options

The **ldifsearch** command takes the following options:

Command options:

-A | --typesOnly

Only retrieve attribute names but not their values.

Default: false

-b | --baseDn {baseDN}

The base DN for the search. If no base DN is provided, then the root DSE will be used.

Default:

-l | --timeLimit {timeLimit}

Maximum length of time in seconds to allow for the search.

Default: 0

-o | --outputLdif {file}

Write search results to {file} instead of stdout.

Default: stdout

-s | --searchScope {searchScope}

Search scope ('base', 'one', 'sub', or 'subordinates'). Note: 'subordinates' is an LDAP extension that might not work with all LDAP servers.

Default: sub

-z | --sizeLimit {sizeLimit}

Maximum number of entries to return from the search.

Default: 0

Utility input/output options:

-t | --wrapColumn {wrapColumn}

Maximum length of an output line (0 for no wrapping).

Default: 0

General options:

-V | --version

Display Directory Server version information.

Default: false

-H | --help

Display this usage information.

Default: false

Exit Codes

0

The command completed successfully.

> 0

An error occurred.

Examples

The following example demonstrates use of the command.

```
$ ldifsearch -b dc=example,dc=com Example.ldif uid=bjensen
dn: uid=bjensen,ou=People,dc=example,dc=com
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: top
uid: bjensen
userpassword: hifalutin
facsimiletelephonenumber: +1 408 555 1992
givenname: Barbara
cn: Barbara Jensen
cn: Babs Jensen
telephonenumber: +1 408 555 1862
sn: Jensen
roomnumber: 0209
homeDirectory: /home/bjensen
mail: bjensen@example.com
l: San Francisco
ou: Product Development
ou: People
uidNumber: 1076
gidNumber: 1000
```

You can also use `@objectclass` notation in the attribute list to return the attributes of a particular object class. The following example shows how to return attributes of the `posixAccount` object class.

```
$ ldifsearch -b dc=example,dc=com Example.ldif "(uid=bjensen)" @posixaccount
dn: uid=bjensen,ou=People,dc=example,dc=com
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: top
uid: bjensen
userpassword: hifalutin
cn: Barbara Jensen
cn: Babs Jensen
homeDirectory: /home/bjensen
uidNumber: 1076
gidNumber: 1000
```

Name

makeldif — generate test LDIF

Synopsis

`makeldif` template-file-path

Description

This utility can be used to generate LDIF data based on a definition in a template file.

The *`template-file-path`* can be one of the following:

- A full path to the template file such as `/path/to/opendj/config/MakeLDIF/example.template`.
- A relative path to the template file such as `../../my-test-data.template`.
- A file name that specifies one of the template files that are built into the ForgeRock Directory Services LDAP Toolkit, such as `example.template`, or `people_and_groups.template`.

The ForgeRock Directory Services LDAP Toolkit includes these built-in template and data files:

`cities`

List of more than 200 cities.

`example.template`

Template to generate a base entry and users in a branch `ou=people,[suffix]`, where the default setting for suffix is `suffix=dc=example,dc=com`.

`first.names`

List of more than 8000 first names.

`last.names`

List of more than 13000 last names.

`people_and_groups.template`

Template to generate a base entry, users, and groups.

`states`

List of US states by their two-character codes.

`streets`

List of more than 70 street names.

Options

The **makeldif** command takes the following options:

Command options:

-c | --constant {name=value}

A constant that overrides the value set in the template file.

-o | --outputLdif {file}

The path to the LDIF file to be written. If the filename ends in .gz, the output will be gzipped.

-r | --resourcePath {path}

Path to look for MakeLDIF resources (e.g., data files).

The utility looks for resources in the following locations in this order:

1. The current directory where the command is run.
2. The resource path directory.
3. The built-in files.

-s | --randomSeed {seed}

The seed to use to initialize the random number generator.

Default: 0

Utility input/output options:

-t | --wrapColumn {wrapColumn}

Maximum length of an output line (0 for no wrapping).

Default: 0

General options:

-V | --version

Display Directory Server version information.

Default: false

-H | --help

Display this usage information.

Default: false

Exit Codes

0

The command completed successfully.

1

An error occurred.

Examples

The following example uses the default template to generate LDIF.

```
$ makeldif -o ../ldif/generated.ldif ../config/MakeLDIF/example.template
Processed 1000 entries
Processed 2000 entries
...
Processed 10000 entries
LDIF processing complete.  10003 entries written
```

See Also

makeldif.template(5)

Name

makeldif.template — template file for the makeldif command

Synopsis

```
# Comment lines start with #.
#
# Notice that this synopsis includes blank lines after entries.
# In the same way you would use blank lines after entries in normal LDIF,
# leave empty lines after "entries" in template files.

# Optionally define constants used in the template.
# To reference constants later, put brackets around the name: [constant-name]
#
define constant-name=value
...

# Define branches by suffix DN, such as the following:
#
#  dc=example,dc=com
#  ou=People,dc=example,dc=com
#  ou=Groups,dc=example,dc=com
#
# makeldif generates the necessary object class definitions and RDNs.
#
# A branch can have subordinateTemplates that define templates to use for
# the branch entry. The optional number at the end
# of the subordinateTemplate specification defines how many entries to generate.
# If you do not specify a number, makeldif continues to generate entries
# indefinitely until you interrupt the command.
#
# A branch can have additional attributes generated on the branch entry. See
# the Description below for more information on specifying attribute values.
#
branch: suffix-dn
objectClass: top
objectClass: suffix-object-class
[subordinateTemplate: template-name[:number]
...]
[attribute: attr-value
...]

...

# Define entries using templates.
#
# A template can extend another template.
# A template defines the RDN attribute(s) used for generated entries.
# A template can have a subordinateTemplate that defines a template to use for
# the generated entries.
#
# A template then defines attributes. See the Description below for more
# information on specifying attribute values.
#
template: template-name
[extends: template-name]
rdnAttr: attribute[+attribute ...]
```

```
[subordinateTemplate: template-name:number]
[attribute: attr-value
...]

...
```

Description

Template files specify how to build LDIF. They allow you to define variables, insert random values from other files, and generally build arbitrarily large LDIF files for testing purposes. You pass template files to the **makeldif** command when generating LDIF.

The Synopsis above shows the layout for a **makeldif** template file. This section focuses on what you can do to specify entry attribute values, called *attr-value* in the Synopsis section.

Specifying Attribute Values

When specifying attribute values in **makeldif** templates, you can use static text and constants that you have defined, enclosing names for constants in brackets, `[myConstant]`. You can use more than one constant per line, as in the following example:

```
description: Description for [org] under [suffix]
```

You can also use two kinds of tags when specifying attribute values. One kind of tag is replaced with the value of another attribute in the generated entry. Such tags are delimited with braces, `{ }`. For example, if your template includes definitions for first name and last name attributes, use:

```
givenName: <first>
sn: <last>
```

Then you can define a mail attribute that uses the values of both attributes, and an initials attribute that takes the first character of each:

```
mail: {givenName}.{sn}@[myDomain]
initials: {givenName:1}{sn:1}
```

The other kind of tag is delimited with `<` and `>`, as shown above in the example with `<first>` and `<last>`. Tag names are not case sensitive. Many tags can take arguments separated by colons, `:`, from the tag names within the tag.

Use backslashes to escape literal start tag characters (`< [ {`) as shown in the following example, and to escape literal end tag characters within tags (`> ] }`):

```
scimMail: \{"emails": \[\{"value": "{mail}", "type": "work", "primary": true}]}
xml: \<id>{uid}\</id>
```

The **makeldif** command supports the following tags:

<DN>

The DN tag is replaced by the distinguished name of the current entry. An optional integer argument specifies the subcomponents of the DN to generate. For example, if the DN of the

entry is `uid=bjensen,ou=People,dc=example,dc=com`, then `<DN:1>` is replaced by `uid=bjensen`, and `<DN:-2>` is replaced by `dc=example,dc=com`.

<File>

The File tag is replaced by a line from a text file you specify. The File tag takes a required argument, the path to the text file, and an optional second argument, either `random` or `sequential`. For the file argument, either specify an absolute path to the file such as `<file:/path/to/myDescriptions>`, or specify a path relative to the template file such as `<file:streets>`. For the second argument, if you specify `sequential` then lines from the file are read in sequential order. Otherwise, lines from the file are read in random order.

<First>

The first name tag is replaced by a random line from `first.names`. Combinations of generated first and last names are unique, with integers appended to the name strings if not enough combinations are available.

<GUID>

The GUID tag is replaced by a 128-bit, type 4 (random) universally unique identifier, such as `f47ac10b-58cc-4372-a567-0e02b2c3d479`.

<IfAbsent>

The IfAbsent tag takes as its first argument the name of another attribute, and optionally, as its second argument, a value to use. This tag causes the attribute to be generated only if the named attribute is not present on the generated entry. Use this tag when you have used `<Presence>` to define another attribute that is not always present on generated entries.

<IfPresent>

The IfPresent takes as its first argument the name of another attribute, and optionally, as its second argument, a value to use. This tag causes the attribute to be generated only if the named attribute is also present on the generated entry. Use this tag when you have used `<Presence>` to define another attribute that is sometimes present on generated entries.

<Last>

The last name tag is replaced by a random line from the last names template file, `last.names`. Combinations of generated first and last names are unique, with integers appended to the name strings if not enough combinations are available.

<List>

The List tag is replaced by one of the values from the list of arguments you provide. For example, `<List:bronze:silver:gold>` is replaced with `bronze`, `silver`, or `gold`.

You can weight arguments to ensure that some arguments are selected more often than others. For example, if you want two bronze for one silver and one gold, use `<List:bronze;2:silver;1:gold;1>`.

<ParentDN>

The ParentDN tag is replaced by the distinguished name of the parent entry. For example, if the DN of the entry is `uid=bjensen,ou=People,dc=example,dc=com`, `<ParentDN>` is replaced by `ou=People,dc=example,dc=com`.

<Presence>

The Presence tag takes a percent argument. It results in the attribute value being generated or not based on the percentage of entries you specify in the argument. For example, `description: <Presence:50>A description` generates `description: A description` on half the entries.

<Random>

The Random tag lets you generate a variety of random numbers and strings. The Random tag has the following subtypes, which you include as arguments, that is `<Random:subtype>`:

- `alpha:length`
- `alpha:min-length:max-length`
- `numeric:length`
- `numeric:minvalue:maxvalue`
- `numeric:minvalue:maxvalue:format`, where *`format`* is a `java.text.DecimalFormat` pattern
- `alphanumeric:length`
- `alphanumeric:min-length:max-length`
- `chars:characters:length`
- `chars:characters:min-length:max-length`
- `hex:length`
- `hex:min-length:max-length`
- `base64:length`
- `base64:min-length:max-length`
- `month`
- `month:max-length`
- `telephone`, a telephone number starting with the country code `+1`

<RDN>

The RDN tag is replaced with the RDN of the entry. Use this in the template after you have specified `rdnAttr` so that the RDN has already been generated when this tag is replaced.

An optional integer argument specifies the subcomponents of the RDN to generate.

<Sequential>

The Sequential tag is replaced by a sequentially increasing generated integer. The first optional integer argument specifies the starting number. The second optional boolean argument specifies whether to start over when generating entries for a new parent entry. For example, `<Sequential>:42:true` starts counting from 42, and starts over when the parent entry changes from `o=Engineering` to `o=Marketing`.

<_DN>

The _DN tag is replaced by the DN of the current entry with underscores in the place of commas.

<_ParentDN>

The _ParentDN tag is replaced by the DN the parent entry with underscores in the place of commas.

Examples

The following example generates 10 organization units, each containing 50 entries. Add it next to the supporting files, such as `first.names` and `last.names` needed to generate the output:

```
define suffix=dc=example,dc=com
define maildomain=example.com
define numusers=50
define numorgs=10

branch: [suffix]
objectClass: top
objectClass: domain

branch: ou=People,[suffix]
objectClass: top
objectClass: organizationalUnit
subordinateTemplate: orgunit:[numorgs]
description: This is the People container
telephoneNumber: +33 00010002

template: orgunit
subordinateTemplate: person:[numusers]
rdnAttr: ou
ou: Org-<sequential:0>
objectClass: top
objectClass: organizationalUnit
description: This is the {ou} organizational unit

template: person
rdnAttr: uid
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
givenName: <first>
sn: <last>
```

```
cn: {givenName} {sn}
initials: {givenName:1}<random:chars:ABCDEFGHIJKLMNOPQRSTUVWXYZ:1>{sn:1}
employeeNumber: <sequential:0>
uid: user.{employeeNumber}
mail: {uid}@[maildomain]
userPassword: password
telephoneNumber: <random:telephone>
homePhone: <random:telephone>
pager: <random:telephone>
mobile: <random:telephone>
street: <random:numeric:5> <file:streets> Street
l: <file:cities>
st: <file:states>
postalCode: <random:numeric:5>
postalAddress: {cn}${street}${l}, {st}  {postalCode}
description: This is the description for {cn}.
```

See Also

makeldif(1), the server template file `config/MakeLDIF/example.template`

Name

manage-account — manage state of OpenDJ server accounts

Synopsis

`manage-account` {subcommand} {options}

Description

This utility can be used to retrieve and manipulate the values of password policy state variables.

Options

The **manage-account** command takes the following options:

Command options:

-b | --targetDn {targetDN}

The DN of the user entry for which to get and set password policy state information.

LDAP connection options:

-D | --bindDn {bindDN}

The DN to use to bind to the server.

-h | --hostname {host}

Directory server hostname or IP address.

Default: localhost.localdomain

-j | --bindPasswordFile {bindPasswordFile}

The path to the file containing the bind password.

-K | --keyStorePath {keyStorePath}

Certificate key store path.

-N | --certNickname {nickname}

Nickname of certificate for SSL client authentication.

-o | --saslOption {name=value}

SASL bind options.

-p | --port {port}

Directory server administration port number.

Default: 4444

-P | --trustStorePath {trustStorePath}

Certificate trust store path.

-T | --trustStorePassword {trustStorePassword}

Certificate trust store PIN.

-u | --keyStorePasswordFile {keyStorePasswordFile}

Certificate key store PIN file.

-U | --trustStorePasswordFile {path}

Certificate trust store PIN file.

-w | --bindPassword {bindPassword}

The password to use to bind to the server.

-W | --keyStorePassword {keyStorePassword}

Certificate key store PIN.

-X | --trustAll

Trust all server SSL certificates.

Default: false

Utility input/output options:

-v | --verbose

Use verbose mode.

Default: false

General options:

-V | --version

Display Directory Server version information.

Default: false

-H | --help

Display this usage information.

Default: false

Subcommands

The **manage-account** command supports the following subcommands:

manage-account clear-account-is-disabled

Clear account disabled state information from the user account.

manage-account get-account-expiration-time

Display when the user account will expire.

manage-account get-account-is-disabled

Display information about whether the user account has been administratively disabled.

manage-account get-all

Display all password policy state information for the user.

manage-account get-authentication-failure-times

Display the authentication failure times for the user.

manage-account get-grace-login-use-times

Display the grace login use times for the user.

manage-account get-last-login-time

Display the time that the user last authenticated to the server.

manage-account get-password-changed-by-required-time

Display the required password change time with which the user last complied.

manage-account get-password-changed-time

Display the time that the user's password was last changed.

manage-account get-password-expiration-warned-time

Display the time that the user first received an expiration warning notice.

manage-account get-password-history

Display password history state values for the user.

manage-account get-password-is-reset

Display information about whether the user will be required to change his or her password on the next successful authentication.

manage-account get-password-policy-dn

Display the DN of the password policy for the user.

manage-account get-remaining-authentication-failure-count

Display the number of remaining authentication failures until the user's account is locked.

manage-account get-remaining-grace-login-count

Display the number of grace logins remaining for the user.

manage-account get-seconds-until-account-expiration

Display the length of time in seconds until the user account expires.

manage-account get-seconds-until-authentication-failure-unlock

Display the length of time in seconds until the authentication failure lockout expires.

manage-account get-seconds-until-idle-lockout

Display the length of time in seconds until user's account is locked because it has remained idle for too long.

manage-account get-seconds-until-password-expiration

Display length of time in seconds until the user's password expires.

manage-account get-seconds-until-password-expiration-warning

Display the length of time in seconds until the user should start receiving password expiration warning notices.

manage-account get-seconds-until-password-reset-lockout

Display the length of time in seconds until user's account is locked because the user failed to change the password in a timely manner after an administrative reset.

manage-account get-seconds-until-required-change-time

Display the length of time in seconds that the user has remaining to change his or her password before the account becomes locked due to the required change time.

manage-account set-account-is-disabled

Specify whether the user account has been administratively disabled.

Options

The **manage-account set-account-is-disabled** command takes the following options:

-O | --operationValue {true|false}

'true' to indicate that the account is disabled, or 'false' to indicate that it is not disabled.

Exit Codes

0

The command completed successfully.

89

An error occurred while parsing the command-line arguments.

Examples

For the following examples the directory admin user, Kirsten Vaughan, has `ds-privilege-name: password-reset` and the following ACI on `ou=People,dc=example,dc=com`.

```
(target="ldap:///ou=People,dc=example,dc=com") (targetattr ="*||+")(
 version 3.0;acl "Admins can run amok"; allow(all) groupdn =
 "ldap:///cn=Directory Administrators,ou=Groups,dc=example,dc=com";)
```

The following command locks a user account.

```
$ manage-account -p 4444 -D "uid=kvaughan,ou=people,dc=example,dc=com" \
 -w bribery set-account-is-disabled -O true \
 -b uid=bjensen,ou=people,dc=example,dc=com -X
Account Is Disabled:  true
```

The following command unlocks a user account.

```
$ manage-account -p 4444 -D "uid=kvaughan,ou=people,dc=example,dc=com" \
 -w bribery clear-account-is-disabled \
 -b uid=bjensen,ou=people,dc=example,dc=com -X
Account Is Disabled:  false
```

Name

manage-tasks — manage server administration tasks

Synopsis

```
manage-tasks
```

Description

This utility can be used to obtain a list of tasks scheduled to run within the Directory Server as well as information about individual tasks.

Options

The **manage-tasks** command takes the following options:

Command options:

-c | --cancel {taskID}

ID of a particular task to cancel.

--connectTimeout {timeout}

Maximum length of time (in milliseconds) that can be taken to establish a connection. Use '0' to specify no time out.

Default: 30000

-i | --info {taskID}

ID of a particular task about which this tool will display information.

-s | --summary

Print a summary of tasks.

Default: false

LDAP connection options:

-D | --bindDn {bindDN}

DN to use to bind to the server.

Default: cn=Directory Manager

-h | --hostname {host}

The fully-qualified directory server host name that will be used when generating self-signed certificates for LDAP SSL/StartTLS, the administration connector, and replication.

Default: localhost.localdomain

-j | --bindPasswordFile {bindPasswordFile}

Bind password file.

-K | --keyStorePath {keyStorePath}

Certificate key store path.

-N | --certNickname {nickname}

Nickname of the certificate that the server should use when accepting SSL-based connections or performing StartTLS negotiation.

-o | --saslOption {name=value}

SASL bind options.

-p | --port {port}

Directory server administration port number.

Default: 4444

-P | --trustStorePath {trustStorePath}

Certificate trust store path.

-T | --trustStorePassword {trustStorePassword}

Certificate trust store PIN.

-u | --keyStorePasswordFile {keyStorePasswordFile}

Certificate key store PIN file. A PIN is required when you specify to use an existing certificate as server certificate.

-U | --trustStorePasswordFile {path}

Certificate trust store PIN file.

-w | --bindPassword {bindPassword}

Password to use to bind to the server. Use -w - to ensure that the command prompts for the password, rather than entering the password as a command argument.

-W | --keyStorePassword {keyStorePassword}

Certificate key store PIN. A PIN is required when you specify to use an existing certificate as server certificate.

-X | --trustAll

Trust all server SSL certificates.

Default: false

Utility input/output options:

-n | --no-prompt

Use non-interactive mode. If data in the command is missing, the user is not prompted and the tool will fail.

Default: false

--noPropertiesFile

No properties file will be used to get default command line argument values.

Default: false

--propertiesFilePath {propertiesFilePath}

Path to the file containing default property values used for command line arguments.

General options:

-V | --version

Display Directory Server version information.

Default: false

-H | --help

Display this usage information.

Default: false

Exit Codes

0

The command completed successfully.

> 0

An error occurred.

Examples

The following example demonstrates use of the command with a server that does daily backups at 2:00 AM.

```
$ manage-tasks -p 4444 -h opendj.example.com -D "cn=Directory Manager" \
 -w password -s

  ID                               Type    Status
  ---------------------------------------------------------------
  example-backup                   Backup  Recurring
  example-backup-20110622020000000  Backup  Waiting on start time
```

Name

modrate — measure modification throughput and response time

Synopsis

`modrate` [(attribute:value format string) ...]

Description

This utility can be used to measure modify throughput and response time of a directory service using user-defined modifications.

Example:

modrate -p 1389 -D "cn=directory manager" -w password \

-F -c 4 -t 4 -b "uid=user.%d,ou=people,dc=example,dc=com" \

-g "rand(0,2000)" -g "randstr(16)" 'description:%2$s'

Before trying the example, import 2000 randomly generated users.

When you do not use the `-f` option to keep connections open and rebind on the connections, the tool can exhaust its available ports, causing the tool to crash. You can work around this problem on test systems by changing TCP settings on the system.

For example, on Linux systems, set the following parameters in the `/etc/sysctl.conf` file:

```
net.ipv4.tcp_fin_timeout = 30
net.ipv4.tcp_tw_recycle = 1
net.ipv4.tcp_tw_reuse = 1
```

The parameter `net.ipv4.tcp_fin_timeout` sets the length of time in seconds to wait for a final FIN packet before forcing a close of the socket. The default is 60 (seconds).

The parameter `net.ipv4.tcp_tw_recycle` enables fast recycling of TIME_WAIT sockets. The default is 0 (false). Enabling this can cause Network Address Translation (NAT) issues.

The parameter `net.ipv4.tcp_tw_reuse` enables reuse of TIME_WAIT sockets for new connections. The default is 0 (false).

These settings are recommended only for testing, and *not for production systems*.

After making the changes to `/etc/sysctl.conf`, reload the configuration with the **sysctl** command:

```
# sysctl -p
```

Options

The **modrate** command takes the following options:

Command options:

-b | --targetDn {targetDN}

Target entry DN format string.

-B | --warmUpDuration {warmUpDuration}

Warm up duration in seconds.

Default: 0

-c | --numConnections {numConnections}

Number of connections.

Default: 1

--connectTimeout {timeout}

Maximum length of time (in milliseconds) that can be taken to establish a connection. Use '0' to specify no time out.

Default: 30000

-d | --maxDuration {maxDuration}

Maximum duration in seconds, 0 for unlimited.

Default: 0

-e | --percentile {percentile}

Calculate max response time for a percentile of operations.

-f | --keepConnectionsOpen

Keep connections open.

Default: false

-F | --noRebind

Keep connections open and do not rebind.

Default: false

-g | --argument {generator function or static string}

Argument used to evaluate the Java style format strings in program parameters (ie. Base DN, Search Filter). The set of all arguments provided form the the argument list in order. Besides static string arguments, they can be generated per iteration with the following functions:

"inc({filename})" Consecutive, incremental line from file

"inc({min},{max})" Consecutive, incremental number

"rand({filename})" Random line from file

"rand({min},{max})" Random number

"randstr({length},_charSet_)" Random string of specified length and optionally from characters in the charSet string. A range of character can be specified with [start-end] charSet notation. If no charSet is specified, the default charSet of [A-Z][a-z][0-9] will be used.

-i | --statInterval {statInterval}

Display results each specified number of seconds.

Default: 5

-m | --maxIterations {maxIterations}

Max iterations, 0 for unlimited.

Default: 0

-M | --targetThroughput {targetThroughput}

Target average throughput to achieve.

Default: 0

-S | --scriptFriendly

Use script-friendly mode.

Default: false

-t | --numThreads {numThreads}

Number of worker threads per connection.

Default: 1

LDAP connection options:

-D | --bindDn {bindDN}

DN to use to bind to the server.

Default:

-E | --reportAuthzId

Use the authorization identity control.

Default: false

-h | --hostname {host}

The fully-qualified directory server host name that will be used when generating self-signed certificates for LDAP SSL/StartTLS, the administration connector, and replication.

Default: localhost.localdomain

-j | --bindPasswordFile {bindPasswordFile}

Bind password file.

-K | --keyStorePath {keyStorePath}

Certificate key store path.

-N | --certNickname {nickname}

Nickname of the certificate that the server should use when accepting SSL-based connections or performing StartTLS negotiation.

-o | --saslOption {name=value}

SASL bind options.

-p | --port {port}

Directory server port number.

Default: 389

-P | --trustStorePath {trustStorePath}

Certificate trust store path.

-q | --useStartTls

Use StartTLS to secure communication with the server.

Default: false

-T | --trustStorePassword {trustStorePassword}

Certificate trust store PIN.

-u | --keyStorePasswordFile {keyStorePasswordFile}

Certificate key store PIN file. A PIN is required when you specify to use an existing certificate as server certificate.

-U | --trustStorePasswordFile {path}

Certificate trust store PIN file.

--usePasswordPolicyControl

Use the password policy request control.

Default: false

-w | --bindPassword {bindPassword}

Password to use to bind to the server. Use -w - to ensure that the command prompts for the password, rather than entering the password as a command argument.

-W | --keyStorePassword {keyStorePassword}

Certificate key store PIN. A PIN is required when you specify to use an existing certificate as server certificate.

-X | --trustAll

Trust all server SSL certificates.

Default: false

-Z | --useSsl

Use SSL for secure communication with the server.

Default: false

Utility input/output options:

--noPropertiesFile

No properties file will be used to get default command line argument values.

Default: false

--propertiesFilePath {propertiesFilePath}

Path to the file containing default property values used for command line arguments.

-v | --verbose

Use verbose mode.

Default: false

General options:

-V | --version

Display Directory Server version information.

Default: false

-H | --help

Display this usage information.

Default: false

Exit Codes

0

The command completed successfully.

89

An error occurred while parsing the command-line arguments.

Examples

The following example demonstrates testing directory performance by using the **modrate** command to write random 16-character description values to all entries in a sample file:

```
$ grep ^uid: /path/to/Example.ldif | sed -e "s/uid: //" > names.txt
$ modrate -p 1389 -D "cn=Directory Manager" -w password  -A -F -c 4 -t 4 \
 -b "uid=%s,ou=people,dc=example,dc=com" -g "rand(names.txt)" \
 -g "randstr(16)" 'description:%2$s'
--------------------------------------------------------------------------------
     Throughput                          Response Time
   (ops/second)                         (milliseconds)
recent  average  recent  average  99.9%  99.99%  99.999%  err/sec  req/
res
--------------------------------------------------------------------------------
1085.9   1088.5  993.849  993.849  2135.220  2510.361  2510.361  0.0  2.3
2086.7   1648.8  1963.980  1683.038  3015.025  3078.628  3215.050  0.0  1.0
3097.3   2092.6  1332.930  1524.278  2940.131  3024.811  3215.050  0.0  1.0
3848.3   2501.4  1045.000  1352.583  2902.235  3015.863  3215.050  0.0  1.0
3641.2   2717.4  1106.157  1290.003  2901.379  3015.597  3215.050  0.0  1.0
3759.4   2883.0  1065.732  1243.534  2900.400  3015.501  3215.050  0.0  1.0
^C
```

Name

rebuild-index — rebuild index after configuration change

Synopsis

```
rebuild-index
```

Description

This utility can be used to rebuild index data within an indexed backend database.

Options

The **rebuild-index** command takes the following options:

Command options:

-b | --baseDn {baseDN}

Base DN of a backend supporting indexing. Rebuild is performed on indexes within the scope of the given base DN.

--clearDegradedState

Indicates that indexes do not need rebuilding because they are known to be empty and forcefully marks them as valid. This is an advanced option which must only be used in cases where a degraded index is known to be empty and does not therefore need rebuilding. This situation typically arises when an index is created for an attribute which has just been added to the schema.

Default: false

-i | --index {index}

Names of index(es) to rebuild. For an attribute index this is simply an attribute name. At least one index must be specified for rebuild. Cannot be used with the "--rebuildAll" option.

--offline

Indicates that the command must be run in offline mode.

Default: false

--rebuildAll

Rebuild all indexes, including any DN2ID, DN2URI, VLV and extensible indexes. Cannot be used with the "-i" option or the "--rebuildDegraded" option.

Default: false

--rebuildDegraded

Rebuild all degraded indexes, including any DN2ID, DN2URI, VLV and extensible indexes. Cannot be used with the "-i" option or the "--rebuildAll" option.

Default: false

--tmpDirectory {directory}

Path to temporary directory for index scratch files during index rebuilding.

Default: import-tmp

Task Backend Connection Options

--connectTimeout {timeout}

Maximum length of time (in milliseconds) that can be taken to establish a connection. Use '0' to specify no time out.

Default: 30000

-D | --bindDn {bindDN}

DN to use to bind to the server.

Default: cn=Directory Manager

-h | --hostname {host}

The fully-qualified directory server host name that will be used when generating self-signed certificates for LDAP SSL/StartTLS, the administration connector, and replication.

Default: localhost.localdomain

-j | --bindPasswordFile {bindPasswordFile}

Bind password file.

-K | --keyStorePath {keyStorePath}

Certificate key store path.

-N | --certNickname {nickname}

Nickname of the certificate that the server should use when accepting SSL-based connections or performing StartTLS negotiation.

-o | --saslOption {name=value}

SASL bind options.

-p | --port {port}

Directory server administration port number.

Default: 4444

-P | --trustStorePath {trustStorePath}

Certificate trust store path.

-T | --trustStorePassword {trustStorePassword}

Certificate trust store PIN.

-u | --keyStorePasswordFile {keyStorePasswordFile}

Certificate key store PIN file. A PIN is required when you specify to use an existing certificate as server certificate.

-U | --trustStorePasswordFile {path}

Certificate trust store PIN file.

-w | --bindPassword {bindPassword}

Password to use to bind to the server. Use -w - to ensure that the command prompts for the password, rather than entering the password as a command argument.

-W | --keyStorePassword {keyStorePassword}

Certificate key store PIN. A PIN is required when you specify to use an existing certificate as server certificate.

-X | --trustAll

Trust all server SSL certificates.

Default: false

Task Scheduling Options

--completionNotify {emailAddress}

Email address of a recipient to be notified when the task completes. This option may be specified more than once.

--dependency {taskID}

ID of a task upon which this task depends. A task will not start execution until all its dependencies have completed execution.

--errorNotify {emailAddress}

Email address of a recipient to be notified if an error occurs when this task executes. This option may be specified more than once.

--failedDependencyAction {action}

Action this task will take should one if its dependent tasks fail. The value must be one of PROCESS,CANCEL,DISABLE. If not specified defaults to CANCEL.

--recurringTask {schedulePattern}

Indicates the task is recurring and will be scheduled according to the value argument expressed in crontab(5) compatible time/date pattern.

-t | --start {startTime}

Indicates the date/time at which this operation will start when scheduled as a server task expressed in YYYYMMDDhhmmssZ format for UTC time or YYYYMMDDhhmmss for local time. A value of '0' will cause the task to be scheduled for immediate execution. When this option is specified the operation will be scheduled to start at the specified time after which this utility will exit immediately.

Utility input/output options:

--noPropertiesFile

No properties file will be used to get default command line argument values.

Default: false

--propertiesFilePath {propertiesFilePath}

Path to the file containing default property values used for command line arguments.

General options:

-V | --version

Display Directory Server version information.

Default: false

-H | --help

Display this usage information.

Default: false

Exit Codes

0

The command completed successfully.

> 0

An error occurred.

Examples

The following example schedules a task to start immediately that rebuilds the `cn` (common name) index.

```
$ rebuild-index -p 4444 -h opendj.example.com -D "cn=Directory Manager" \
 -w password -b dc=example,dc=com -i cn -t 0
Rebuild Index task 20110607160349596 scheduled to start Jun 7, 2011 4:03:49 PM
```

Name

restore — restore directory data backups

Synopsis

```
restore
```

Description

This utility can be used to restore a backup of a Directory Server backend.

Options

The **restore** command takes the following options:

Command options:

-d | --backupDirectory {backupDir}

Path to the directory containing the backup file(s).

-I | --backupId {backupID}

Backup ID of the backup to restore.

-l | --listBackups

List available backups in the backup directory.

Default: false

-n | --dry-run

Verify the contents of the backup but do not restore it.

Default: false

--offline

Indicates that the command must be run in offline mode.

Default: false

Task Backend Connection Options

--connectTimeout {timeout}

Maximum length of time (in milliseconds) that can be taken to establish a connection. Use '0' to specify no time out.

Default: 30000

-D | --bindDn {bindDN}

DN to use to bind to the server.

Default: cn=Directory Manager

-h | --hostname {host}

The fully-qualified directory server host name that will be used when generating self-signed certificates for LDAP SSL/StartTLS, the administration connector, and replication.

Default: localhost.localdomain

-j | --bindPasswordFile {bindPasswordFile}

Bind password file.

-K | --keyStorePath {keyStorePath}

Certificate key store path.

-N | --certNickname {nickname}

Nickname of the certificate that the server should use when accepting SSL-based connections or performing StartTLS negotiation.

-o | --saslOption {name=value}

SASL bind options.

-p | --port {port}

Directory server administration port number.

Default: 4444

-P | --trustStorePath {trustStorePath}

Certificate trust store path.

-T | --trustStorePassword {trustStorePassword}

Certificate trust store PIN.

-u | --keyStorePasswordFile {keyStorePasswordFile}

Certificate key store PIN file. A PIN is required when you specify to use an existing certificate as server certificate.

-U | --trustStorePasswordFile {path}

Certificate trust store PIN file.

-w | --bindPassword {bindPassword}

Password to use to bind to the server. Use -w - to ensure that the command prompts for the password, rather than entering the password as a command argument.

-W | --keyStorePassword {keyStorePassword}

Certificate key store PIN. A PIN is required when you specify to use an existing certificate as server certificate.

-X | --trustAll

Trust all server SSL certificates.

Default: false

Task Scheduling Options

--completionNotify {emailAddress}

Email address of a recipient to be notified when the task completes. This option may be specified more than once.

--dependency {taskID}

ID of a task upon which this task depends. A task will not start execution until all its dependencies have completed execution.

--errorNotify {emailAddress}

Email address of a recipient to be notified if an error occurs when this task executes. This option may be specified more than once.

--failedDependencyAction {action}

Action this task will take should one if its dependent tasks fail. The value must be one of PROCESS,CANCEL,DISABLE. If not specified defaults to CANCEL.

--recurringTask {schedulePattern}

Indicates the task is recurring and will be scheduled according to the value argument expressed in crontab(5) compatible time/date pattern.

-t | --start {startTime}

Indicates the date/time at which this operation will start when scheduled as a server task expressed in YYYYMMDDhhmmssZ format for UTC time or YYYYMMDDhhmmss for local time. A value of '0' will cause the task to be scheduled for immediate execution. When this option is specified the operation will be scheduled to start at the specified time after which this utility will exit immediately.

Utility input/output options:

--noPropertiesFile

No properties file will be used to get default command line argument values.

Default: false

--propertiesFilePath {propertiesFilePath}

Path to the file containing default property values used for command line arguments.

General options:

-V | --version

Display Directory Server version information.

Default: false

-H | --help

Display this usage information.

Default: false

Exit Codes

0

The command completed successfully.

> 0

An error occurred.

Examples

The following example schedules a restore as a task to begin immediately while OpenDJ directory server is online.

```
$ restore -p 4444 -D "cn=Directory Manager" -w password
 -d /path/to/opendj/bak -I 20110613080032 -t 0
Restore task 20110613155052932 scheduled to start Jun 13, 2011 3:50:52 PM CEST
```

The following example restores data while OpenDJ is offline.

```
$ stop-ds
Stopping Server..
.
...

$ restore --backupDirectory /path/to/opendj/bak/userRoot \
 --listBackups
Backup ID:           20120928102414Z
Backup Date:         28/Sep/2012:12:24:17 +0200
Is Incremental:      false
Is Compressed:       false
Is Encrypted:        false
Has Unsigned Hash:   false
Has Signed Hash:     false
Dependent Upon:      none

$ restore --backupDirectory /path/to/opendj/bak/userRoot \
 --backupID 20120928102414Z
[28/Sep/2012:12:26:20 +0200] ... msg=Restored: 00000000.jdb (size 355179)

$ start-ds
[28/Sep/2012:12:27:29 +0200] ... The Directory Server has started successfully
```

Name

searchrate — measure search throughput and response time

Synopsis

`searchrate` [filter format string] [attributes ...]

Description

This utility can be used to measure search throughput and response time of a directory service using user-defined searches.

Example:

searchrate -p 1389 -D "cn=directory manager" -w password \

-F -c 4 -t 4 -b "dc=example,dc=com" -g "rand(0,2000)" "(uid=user.%d)"

Before trying the example, import 2000 randomly generated users.

When you do not use the `-f` option to keep connections open and rebind on the connections, the tool can exhaust its available ports, causing the tool to crash. You can work around this problem on test systems by changing TCP settings on the system.

For example, on Linux systems, set the following parameters in the `/etc/sysctl.conf` file:

```
net.ipv4.tcp_fin_timeout = 30
net.ipv4.tcp_tw_recycle = 1
net.ipv4.tcp_tw_reuse = 1
```

The parameter `net.ipv4.tcp_fin_timeout` sets the length of time in seconds to wait for a final FIN packet before forcing a close of the socket. The default is 60 (seconds).

The parameter `net.ipv4.tcp_tw_recycle` enables fast recycling of TIME_WAIT sockets. The default is 0 (false). Enabling this can cause Network Address Translation (NAT) issues.

The parameter `net.ipv4.tcp_tw_reuse` enables reuse of TIME_WAIT sockets for new connections. The default is 0 (false).

These settings are recommended only for testing, and *not for production systems*.

After making the changes to `/etc/sysctl.conf`, reload the configuration with the **sysctl** command:

```
# sysctl -p
```

Options

The **searchrate** command takes the following options:

Command options:

-a | --dereferencePolicy {dereferencePolicy}

Alias dereference policy ('never', 'always', 'search', or 'find').

Default: never

-b | --baseDn {baseDN}

Base DN format string.

-B | --warmUpDuration {warmUpDuration}

Warm up duration in seconds.

Default: 0

-c | --numConnections {numConnections}

Number of connections.

Default: 1

--connectTimeout {timeout}

Maximum length of time (in milliseconds) that can be taken to establish a connection. Use '0' to specify no time out.

Default: 30000

-d | --maxDuration {maxDuration}

Maximum duration in seconds, 0 for unlimited.

Default: 0

-e | --percentile {percentile}

Calculate max response time for a percentile of operations.

-f | --keepConnectionsOpen

Keep connections open.

Default: false

-F | --noRebind

Keep connections open and do not rebind.

Default: false

-g | --argument {generator function or static string}

Argument used to evaluate the Java style format strings in program parameters (ie. Base DN, Search Filter). The set of all arguments provided form the the argument list in order. Besides static string arguments, they can be generated per iteration with the following functions:

"inc({filename})" Consecutive, incremental line from file

"inc({min},{max})" Consecutive, incremental number

"rand({filename})" Random line from file

"rand({min},{max})" Random number

"randstr({length},_charSet_)" Random string of specified length and optionally from characters in the charSet string. A range of character can be specified with [start-end] charSet notation. If no charSet is specified, the default charSet of [A-Z][a-z][0-9] will be used.

-i | --statInterval {statInterval}

Display results each specified number of seconds.

Default: 5

-m | --maxIterations {maxIterations}

Max iterations, 0 for unlimited.

Default: 0

-M | --targetThroughput {targetThroughput}

Target average throughput to achieve.

Default: 0

-s | --searchScope {searchScope}

Search scope ('base', 'one', 'sub', or 'subordinates'). Note: 'subordinates' is an LDAP extension that might not work with all LDAP servers.

Default: sub

-S | --scriptFriendly

Use script-friendly mode.

Default: false

-t | --numThreads {numThreads}

Number of worker threads per connection.

Default: 1

LDAP connection options:

-D | --bindDn {bindDN}

DN to use to bind to the server.

Default:

-E | --reportAuthzId

Use the authorization identity control.

Default: false

-h | --hostname {host}

The fully-qualified directory server host name that will be used when generating self-signed certificates for LDAP SSL/StartTLS, the administration connector, and replication.

Default: localhost.localdomain

-j | --bindPasswordFile {bindPasswordFile}

Bind password file.

-K | --keyStorePath {keyStorePath}

Certificate key store path.

-N | --certNickname {nickname}

Nickname of the certificate that the server should use when accepting SSL-based connections or performing StartTLS negotiation.

-o | --saslOption {name=value}

SASL bind options.

-p | --port {port}

Directory server port number.

Default: 389

-P | --trustStorePath {trustStorePath}

Certificate trust store path.

-q | --useStartTls

Use StartTLS to secure communication with the server.

Default: false

-T | --trustStorePassword {trustStorePassword}

Certificate trust store PIN.

-u | --keyStorePasswordFile {keyStorePasswordFile}

Certificate key store PIN file. A PIN is required when you specify to use an existing certificate as server certificate.

-U | --trustStorePasswordFile {path}

Certificate trust store PIN file.

--usePasswordPolicyControl

Use the password policy request control.

Default: false

-w | --bindPassword {bindPassword}

Password to use to bind to the server. Use -w - to ensure that the command prompts for the password, rather than entering the password as a command argument.

-W | --keyStorePassword {keyStorePassword}

Certificate key store PIN. A PIN is required when you specify to use an existing certificate as server certificate.

-X | --trustAll

Trust all server SSL certificates.

Default: false

-Z | --useSsl

Use SSL for secure communication with the server.

Default: false

Utility input/output options:

--noPropertiesFile

No properties file will be used to get default command line argument values.

Default: false

--propertiesFilePath {propertiesFilePath}

Path to the file containing default property values used for command line arguments.

-v | --verbose

Use verbose mode.

Default: false

General options:

-V | --version

Display Directory Server version information.

Default: false

-H | --help

Display this usage information.

Default: false

Exit Codes

0

The command completed successfully.

89

An error occurred while parsing the command-line arguments.

Examples

The following example demonstrates measuring search performance:

```
$ grep ^uid: /path/to/Example.ldif | sed -e "s/uid: //" > names.txt
$ searchrate -p 1389 -b dc=example,dc=com -A -F -c 4 -t 4 \
 -g "rand(names.txt)" "(uid=%s)"
---------------------------------------------------------------------------
     Throughput                          Response Time
   (ops/second)                         (milliseconds)
recent  average  recent  average  99.9%  99.99%  99.999%  err/sec  Entries/
Srch
---------------------------------------------------------------------------
1475.9   1475.9   0.423    0.423  6.938  126.236 126.236      0.0          1.0
2596.5   2038.4   0.254    0.315  6.866  12.980  126.236      0.0          1.0
3210.7   2428.2   0.205    0.267  5.733  11.710  126.236      0.0          1.0
3080.5   2591.0   0.215    0.252  5.733  10.541  126.236      0.0          1.0
3236.9   2720.1   0.203    0.240  5.258  10.514  126.236      0.0          1.0
3181.1   2796.8   0.207    0.234  5.258  10.384  126.236      0.0          1.0
3202.5   2854.8   0.206    0.229  4.825  10.384  126.236      0.0          1.0
^C
```

Name

setup

Synopsis

`setup` {subcommand} {options}

Description

This utility can be used to install an OpenDJ instance either as a directory server, a replication server or a proxy server.

Options

The **setup** command takes the following options:

Command options:

`--acceptLicense`

Automatically accepts the product license (if present).

Default: false

`--adminConnectorPort {port}`

Port on which the Administration Connector should listen for communication.

`-D | --rootUserDn {rootUserDN}`

DN for the initial root user for the Directory Server.

Default: cn=Directory Manager

`--instancePath {path}`

Path were the instance should be set up.

Default: /root/workspace/OpenDJ_-_Release/target/checkout/.

`-j | --rootUserPasswordFile {rootUserPasswordFile}`

Path to a file containing the password for the initial root user for the Directory Server.

`-N | --certNickname {nickname}`

Nickname of a keystore entry containing a certificate that the server should use when negotiating secure connections using StartTLS or SSL. Multiple keystore entries may be provided by using this option multiple times.

-O | --doNotStart

Do not start the server when the configuration is completed.

Default: false

--productionMode

Harden default configuration for production use.

Default: false

-Q | --quiet

Use quiet mode.

Default: false

-S | --skipPortCheck

Skip the check to determine whether the specified ports are usable.

Default: false

-u | --keyStorePasswordFile {keyStorePasswordFile}

Path to a file containing the keystore password. The keystore password is required when you specify an existing file-based keystore (JKS, JCEKS, PKCS#12).

--useJavaKeyStore {keyStorePath}

Path of a JKS keystore containing the certificate(s) that the server should use when negotiating secure connections using StartTLS or SSL.

--useJceks {keyStorePath}

Path of a JCEKS keystore containing the certificate(s) that the server should use when negotiating secure connections using StartTLS or SSL.

--usePkcs11KeyStore

Use certificate(s) in a PKCS#11 token that the server should use when accepting SSL-based connections or performing StartTLS negotiation.

Default: false

--usePkcs12KeyStore {keyStorePath}

Path of a PKCS#12 keystore containing the certificate(s) that the server should use when negotiating secure connections using StartTLS or SSL.

-w | --rootUserPassword {rootUserPassword}

Password for the initial root user for the Directory Server.

-W | --keyStorePassword {keyStorePassword}

Keystore cleartext password. The keystore password is required when you specify an existing file-based keystore (JKS, JCEKS, PKCS#12).

General options:

-V | --version

Display Directory Server version information.

Default: false

-H | --help

Display this usage information.

Default: false

Subcommands

The **setup** command supports the following subcommands:

setup directory-server

Install an OpenDJ directory server instance. See "setup directory-server --help" for specific options.

Options

The **setup directory-server** command takes the following options:

-q | --enableStartTls

Enable StartTLS to allow secure communication with the server using the LDAP port.

Default: false

-p | --ldapPort {port}

Port on which the Directory Server should listen for LDAP communication.

-Z | --ldapsPort {port}

Port on which the Directory Server should listen for LDAPS communication. The LDAPS port will be configured and SSL will be enabled only if this argument is explicitly specified.

-a | --addBaseEntry

Indicates whether to create the base entry in the Directory Server database.

Default: false

-t | --backendType {backendType}

The type of the userRoot backend. Available backend type(s): je.

Default: je

-b | --baseDn {baseDN}

Base DN for user information in the Directory Server. Multiple base DNs may be provided by using this option multiple times.

-l | --ldifFile {ldifFile}

Path to an LDIF file containing data that should be added to the Directory Server database. Multiple LDIF files may be provided by using this option multiple times.

-R | --rejectFile {rejectFile}

Write rejected entries to the specified file.

-d | --sampleData {numEntries}

Specifies that the database should be populated with the specified number of sample entries.

--skipFile {skipFile}

Write skipped entries to the specified file.

-h | --hostname {host}

The fully-qualified directory server host name that will be used when generating self-signed certificates for LDAP SSL/StartTLS, the administration connector, and replication.

Default: localhost.localdomain

--httpPort {port}

Port on which the server should listen for HTTP communication.

--httpsPort {port}

Port on which the server should listen for HTTPS communication.

setup proxy-server

Install an OpenDJ proxy server instance. There are two ways to specify the servers to be contacted by the proxy. They can either be listed exhaustively or retrieved from an existing replication topology. See "setup proxy-server --help" for specific options.

Options

The **setup proxy-server** command takes the following options:

-q | --enableStartTls

Enable StartTLS to allow secure communication with the server using the LDAP port.

Default: false

-p | --ldapPort {port}

Port on which the Directory Server should listen for LDAP communication.

-Z | --ldapsPort {port}

Port on which the Directory Server should listen for LDAPS communication. The LDAPS port will be configured and SSL will be enabled only if this argument is explicitly specified.

--useJceksTrustStore {trustStorePath}

Use existing JCEKS truststore file to use to trust the remote server certificates.

--useJavaTrustStore {trustStorePath}

Use existing JKS truststore file to use to trust the remote server certificates.

--loadBalancingAlgorithm {algorithm}

Algorithm to use to load balance between servers. Available algorithms are 'affinity, least-requests'.

Default: affinity

--usePkcs12TrustStore {trustStorePath}

Use existing PKCS12 truststore file to use to trust the remote server certificates.

--staticPrimaryServer {host:port}

Static server to contact when available before contacting secondary servers. Multiple servers may be provided by using this option multiple times.

--proxyUserBindDn {proxyBindDN}

The bind DN for forwarding LDAP requests to remote servers. This bind DN must be present on all the remote servers.

Default: cn=proxy

--proxyUserBindPassword {proxyBindPassword}

Password associated with the proxy bind DN. The bind password must be the same on all the remote servers.

--proxyUserBindPasswordFile {proxyBindPasswordFile}

Path to a file containing the password associated with the proxy bind DN. The bind password must be the same on all the remote servers.

--replicationBindDn {bindDN}

The bind DN for periodically reading replication server configurations. The bind DN must be present on all replication servers and directory servers, it must be able to read the server configuration.

--replicationBindPassword {bindPassword}

The bind password for periodically reading replication server configurations. The bind password must be the same on all replication and directory servers.

--replicationBindPasswordFile {bindPasswordFile}

Path to a file containing the bind password for periodically reading replication server configurations. The bind password must be the same on all replication and directory servers.

--replicationPreferredGroupId {domainGroupIDNumber}

Replication domain group ID number of directory server replicas to contact when available before contacting other replicas. If this option is not specified then all replicas will be treated the same.

--replicationServer {host:port}

Replication server to contact periodically in order to discover backend servers. Multiple replication servers may be provided by using this option multiple times.

--baseDn {baseDN}

Base DN for user information in the Proxy Server. Multiple base DNs may be provided by using this option multiple times. If no base DNs are defined then the proxy will forward requests to all public naming contexts of the remote servers.

--staticSecondaryServer {host:port}

Static server to contact when all primary servers are unavailable. Multiple servers may be provided by using this option multiple times.

--proxyUsingSsl

Use SSL to secure communications with remote servers.

Default: false

--proxyUsingStartTls

Use Start TLS to secure communication with remote servers.

Default: false

-T | --trustStorePassword {trustStorePassword}

Truststore cleartext password.

-U | --trustStorePasswordFile {path}

Path to a file containing the truststore password.

--useJvmTrustStore

Use the JVM truststore for validating remote server certificates.

Default: false

-X | --trustAll

Trust all server SSL certificates.

Default: false

-h | --hostname {host}

The fully-qualified directory server host name that will be used when generating self-signed certificates for LDAP SSL/StartTLS, the administration connector, and replication.

Default: localhost.localdomain

--httpPort {port}

Port on which the server should listen for HTTP communication.

--httpsPort {port}

Port on which the server should listen for HTTPS communication.

Exit Codes

0

The command completed successfully.

> 0

An error occurred.

Examples

The following command installs OpenDJ directory server, enabling StartTLS and importing 100 example entries without interaction.

```
$ /path/to/opendj/setup directory-server --adminConnectorPort 4444 -t pdb -b dc=example,dc=com -d 100 \
 -D "cn=Directory Manager" -w password -h opendj.example.com -p 1389 \
 --enableStartTLS


Validating parameters..... Done
Configuring certificates..... Done
Configuring server..... Done
Importing automatically-generated data (100 entries)......... Done
Starting directory server.............. Done

To see basic server status and configuration, you can
 launch
/path/to/opendj/bin/status
```

Name

start-ds — start OpenDJ server

Synopsis

```
start-ds
```

Description

This utility can be used to start the Directory Server, as well as to obtain the server version and other forms of general server information.

Options

The **start-ds** command takes the following options:

Command options:

-L | --useLastKnownGoodConfig

Attempt to start using the configuration that was in place at the last successful startup (if it is available) rather than using the current active configuration.

Default: false

-N | --noDetach

Do not detach from the terminal and continue running in the foreground. This option cannot be used with the -t, --timeout option.

Default: false

-s | --systemInfo

Display general system information.

Default: false

-t | --timeout {seconds}

Maximum time (in seconds) to wait before the command returns (the server continues the startup process, regardless). A value of '0' indicates an infinite timeout, which means that the command returns only when the server startup is completed. The default value is 60 seconds. This option cannot be used with the -N, --nodetach option.

Default: 200

Utility input/output options:

-Q | --quiet

Use quiet mode.

Default: false

General options:

-V | --version

Display Directory Server version information.

Default: false

-H | --help

Display this usage information.

Default: false

Exit Codes

0

The command completed successfully.

> 0

An error occurred.

Examples

The following command starts the server without displaying information about the startup process.

```
$ start-ds -Q
```

Name

status — display basic OpenDJ server information

Synopsis

`status` {options}

Description

This utility can be used to display basic server information.

Options

The **status** command takes the following options:

Command options:

--connectTimeout {timeout}

Maximum length of time (in milliseconds) that can be taken to establish a connection. Use '0' to specify no time out.

Default: 30000

LDAP connection options:

-D | --bindDn {bindDN}

DN to use to bind to the server.

Default: cn=Directory Manager

-j | --bindPasswordFile {bindPasswordFile}

Bind password file.

-K | --keyStorePath {keyStorePath}

Certificate key store path.

-N | --certNickname {nickname}

Nickname of the certificate that the server should use when accepting SSL-based connections or performing StartTLS negotiation.

-o | --saslOption {name=value}

SASL bind options.

-P | --trustStorePath {trustStorePath}

Certificate trust store path.

-T | --trustStorePassword {trustStorePassword}

Certificate trust store PIN.

-u | --keyStorePasswordFile {keyStorePasswordFile}

Certificate key store PIN file. A PIN is required when you specify to use an existing certificate as server certificate.

-U | --trustStorePasswordFile {path}

Certificate trust store PIN file.

-w | --bindPassword {bindPassword}

Password to use to bind to the server. Use -w - to ensure that the command prompts for the password, rather than entering the password as a command argument.

-W | --keyStorePassword {keyStorePassword}

Certificate key store PIN. A PIN is required when you specify to use an existing certificate as server certificate.

-X | --trustAll

Trust all server SSL certificates.

Default: false

Utility input/output options:

-n | --no-prompt

Use non-interactive mode. If data in the command is missing, the user is not prompted and the tool will fail.

Default: false

--noPropertiesFile

No properties file will be used to get default command line argument values.

Default: false

--propertiesFilePath {propertiesFilePath}

Path to the file containing default property values used for command line arguments.

-r | --refresh {period}

When this argument is specified, the status command will display its contents periodically. Used to specify the period (in seconds) between two displays of the status.

-s | --script-friendly

Use script-friendly mode.

Default: false

General options:

-V | --version

Display Directory Server version information.

Default: false

-H | --help

Display this usage information.

Default: false

Exit Codes

0

The command completed successfully.

> 0

An error occurred.

Examples

```
$ status -D "cn=Directory Manager" -w password

          --- Server Status ---
Server Run Status:        Started
Open Connections:         1

          --- Server Details ---
Host Name:                localhost.localdomain
Administrative Users:     cn=Directory Manager
Installation Path:        /path/to/opendj
Version:                  OpenDJ version
Java Version:             version
Administration Connector: Port 4444 (LDAPS)

          --- Connection Handlers ---
Address:Port : Protocol    : State
-------------:-------------:--------
-
--           : LDIF        : Disabled
8989         : Replication : Enabled
0.0.0.0:161  : SNMP        : Disabled
0.0.0.0:636  : LDAPS       : Disabled
0.0.0.0:1389 : LDAP        : Enabled
0.0.0.0:1689 : JMX         : Disabled

          --- Data Sources ---
```

```
Base DN:                       dc=example,dc=com
Backend ID:                    userRoot
Entries:                       160
Replication:                   Enabled
Missing Changes:               0
Age of Oldest Missing Change: <not available>

Base DN:     dc=myCompany,dc=com
Backend ID:  myCompanyRoot
Entries:     3
Replication: Disabled

Base DN:     o=myOrg
Backend ID:  myOrgRoot
Entries:     3
Replication: Disabled
```

Name

stop-ds — stop OpenDJ server

Synopsis

```
stop-ds
```

Description

This utility can be used to request that the Directory Server stop running or perform a restart. When run without connection options, this utility sends a signal to the OpenDJ process to stop the server. When run with connection options, this utility connects to the OpenDJ administration port and creates a shutdown task to stop the server.

Options

The **stop-ds** command takes the following options:

Command options:

-r | --stopReason {stopReason}

Reason the server is being stopped or restarted.

-R | --restart

Attempt to automatically restart the server once it has stopped.

Default: false

-t | --stopTime {stopTime}

Indicates the date/time at which the shutdown operation will begin as a server task expressed in format YYYYMMDDhhmmssZ for UTC time or YYYYMMDDhhmmss for local time. A value of '0' will cause the shutdown to be scheduled for immediate execution. When this option is specified the operation will be scheduled to start at the specified time after which this utility will exit immediately.

-Y | --proxyAs {authzID}

Use the proxied authorization control with the given authorization ID.

LDAP connection options:

-D | --bindDn {bindDN}

DN to use to bind to the server.

-h | --hostname {host}

Directory server hostname or IP address.

Default: localhost.localdomain

-j | --bindPasswordFile {bindPasswordFile}

Bind password file.

-K | --keyStorePath {keyStorePath}

Certificate key store path.

-N | --certNickname {nickname}

Nickname of certificate for SSL client authentication.

-o | --saslOption {name=value}

SASL bind options.

-p | --port {port}

Directory server administration port number.

Default: 4444

-P | --trustStorePath {trustStorePath}

Certificate trust store path.

-T | --trustStorePassword {trustStorePassword}

Certificate trust store PIN.

-u | --keyStorePasswordFile {keyStorePasswordFile}

Certificate key store PIN file.

-U | --trustStorePasswordFile {path}

Certificate trust store PIN file.

-w | --bindPassword {bindPassword}

Password to use to bind to the server.

-W | --keyStorePassword {keyStorePassword}

Certificate key store PIN.

-X | --trustAll

Trust all server SSL certificates.

Default: false

Utility input/output options:

--noPropertiesFile

No properties file will be used to get default command line argument values.

Default: false

--propertiesFilePath {propertiesFilePath}

Path to the file containing default property values used for command line arguments.

-Q | --quiet

Use quiet mode.

Default: false

General options:

-V | --version

Display Directory Server version information.

Default: false

-H | --help

Display this usage information.

Default: false

Exit Codes

0

The command completed successfully.

> 0

An error occurred.

Examples

The following example restarts OpenDJ directory server.

```
$ stop-ds --restart
Stopping Server...

...The Directory Server has started successfully
```

Name

uninstall — remove OpenDJ directory server software

Synopsis

`uninstall` {options}

Description

This utility can be used to uninstall the Directory Server.

Options

The **uninstall** command takes the following options:

Command options:

-a | --remove-all

Remove all components of the server (this option is not compatible with the rest of remove options).

Default: false

-b | --backup-files

Remove backup files.

Default: false

-c | --configuration-files

Remove configuration files.

Default: false

--connectTimeout {timeout}

Maximum length of time (in milliseconds) that can be taken to establish a connection. Use '0' to specify no time out.

Default: 30000

-d | --databases

Remove database contents.

Default: false

-e | --ldif-files

Remove LDIF files.

Default: false

-f | --forceOnError

Specifies whether the uninstall should continue if there is an error updating references to this server in remote server instances or not. This option can only be used with the --no-prompt no prompt option.

Default: false

-i | --cli

Use the command line install. If not specified the graphical interface will be launched. The rest of the options (excluding help and version) will only be taken into account if this option is specified.

Default: false

-l | --server-libraries

Remove Server Libraries and Administrative Tools.

Default: false

-L | --log-files

Remove log files.

Default: false

LDAP connection options:

-h | --referencedHostname {host}

The name of this host (or IP address) as it is referenced in remote servers for replication.

Default: localhost.localdomain

-I | --adminUid {adminUID}

User ID of the Global Administrator to use to bind to the server.

Default: admin

-j | --bindPasswordFile {bindPasswordFile}

Bind password file.

-K | --keyStorePath {keyStorePath}

Certificate key store path.

-N | --certNickname {nickname}

Nickname of the certificate that the server should use when accepting SSL-based connections or performing StartTLS negotiation.

-o | --saslOption {name=value}

SASL bind options.

-P | --trustStorePath {trustStorePath}

Certificate trust store path.

-T | --trustStorePassword {trustStorePassword}

Certificate trust store PIN.

-u | --keyStorePasswordFile {keyStorePasswordFile}

Certificate key store PIN file. A PIN is required when you specify to use an existing certificate as server certificate.

-U | --trustStorePasswordFile {path}

Certificate trust store PIN file.

-w | --bindPassword {bindPassword}

Password to use to bind to the server. Use -w - to ensure that the command prompts for the password, rather than entering the password as a command argument.

-W | --keyStorePassword {keyStorePassword}

Certificate key store PIN. A PIN is required when you specify to use an existing certificate as server certificate.

-X | --trustAll

Trust all server SSL certificates.

Default: false

Utility input/output options:

-n | --no-prompt

Use non-interactive mode. If data in the command is missing, the user is not prompted and the tool will fail.

Default: false

--noPropertiesFile

No properties file will be used to get default command line argument values.

Default: false

--propertiesFilePath {propertiesFilePath}

Path to the file containing default property values used for command line arguments.

-Q | --quiet

Use quiet mode.

Default: false

-v | --verbose

Use verbose mode.

Default: false

General options:

-V | --version

Display Directory Server version information.

Default: false

-H | --help

Display this usage information.

Default: false

Exit Codes

0

The command completed successfully.

> 0

An error occurred.

Examples

The following command removes OpenDJ directory server without interaction.

```
$ /path/to/opendj/uninstall -a --cli -I admin -w password -n

Stopping Directory Server ..... Done.
Deleting Files under the Installation Path ..... Done.

The Uninstall Completed Successfully.
To complete the uninstallation, you must delete manually the following files
and directories:
/path/to/opendj/lib
See /var/.../opends-uninstall-3...0.log for a detailed log of this operation.

$ rm -rf /path/to/opendj
```

Name

upgrade — upgrade OpenDJ configuration and application data

Synopsis

`upgrade` {options}

Description

Upgrades OpenDJ configuration and application data so that it is compatible with the installed binaries.

This tool should be run immediately after upgrading the OpenDJ binaries and before restarting the server.

NOTE: this tool does not provide backup or restore capabilities. Therefore, it is the responsibility of the OpenDJ administrator to take necessary precautions before performing the upgrade.

This utility thus performs only part of the upgrade process, which includes the following phases for a single server.

1. Get and unpack a newer version of OpenDJ directory server software.
2. Stop the current OpenDJ directory server.
3. Overwrite existing binary and script files with those of the newer version, and then run this utility before restarting OpenDJ.
4. Start the upgraded OpenDJ directory server.

Important

This utility *does not back up OpenDJ before you upgrade, nor does it restore OpenDJ if the utility fails.* In order to revert a failed upgrade, make sure you back up OpenDJ directory server before you overwrite existing binary and script files.

By default this utility requests confirmation before making important configuration changes. You can use the `--no-prompt` option to run the command non-interactively.

When using the `--no-prompt` option, if this utility cannot complete because it requires confirmation for a potentially very long or critical task, then it exits with an error and a message about how to finish making the changes. You can add the `--force` option to force a non-interactive upgrade to continue in this case, also performing long running and critical tasks.

After upgrading, see the resulting `upgrade.log` file for a full list of operations performed.

Options

The **upgrade** command takes the following options:

Command options:

--acceptLicense

Automatically accepts the product license (if present).

Default: false

--force

Forces a non-interactive upgrade to continue even if it requires user interaction. In particular, long running or critical upgrade tasks, such as re-indexing, which require user confirmation will be skipped. This option may only be used with the 'no-prompt' option.

Default: false

--ignoreErrors

Ignores any errors which occur during the upgrade. This option should be used with caution and may be useful in automated deployments where potential errors are known in advance and resolved after the upgrade has completed.

Default: false

Utility input/output options:

-n | --no-prompt

Use non-interactive mode. If data in the command is missing, the user is not prompted and the tool will fail.

Default: false

-Q | --quiet

Use quiet mode.

Default: false

-v | --verbose

Use verbose mode.

Default: false

General options:

-V | --version

Display Directory Server version information.

Default: false

-H | --help

Display this usage information.

Default: false

Exit Codes

0

The command completed successfully.

2

The command was run in non-interactive mode, but could not complete because confirmation was required to run a long or critical task.

See the error message or the log for details.

other

An error occurred.

See the *OpenDJ Installation Guide* for an example upgrade process for OpenDJ directory server installed from the cross-platform (.zip) delivery.

Native packages (.deb, .rpm) perform more of the upgrade process, stopping OpenDJ if it is running, overwriting older files with newer files, running this utility, and starting OpenDJ if it was running when you upgraded the package(s).

Name

verify-index — check index for consistency or errors

Synopsis

```
verify-index
```

Description

This utility can be used to ensure that index data is consistent within an indexed backend database.

Options

The **verify-index** command takes the following options:

Command options:

-b | --baseDn {baseDN}

Base DN of a backend supporting indexing. Verification is performed on indexes within the scope of the given base DN.

-c | --clean

Specifies that a single index should be verified to ensure it is clean. An index is clean if each index value references only entries containing that value. Only one index at a time may be verified in this way.

Default: false

--countErrors

Count the number of errors found during the verification and return that value as the exit code (values > 255 will be reduced to 255 due to exit code restrictions).

Default: false

-i | --index {index}

Name of an index to be verified. For an attribute index this is simply an attribute name. Multiple indexes may be verified for completeness, or all indexes if no indexes are specified. An index is complete if each index value references all entries containing that value.

General options:

-V | --version

Display Directory Server version information.

Default: false

-H | --help

Display this usage information.

Default: false

Exit Codes

0

The command completed successfully.

1

The command was run in non-interactive mode, but could not complete because confirmation was required to run a long or critical task.

See the error message or the log for details.

0-255

The number of errors in the index, as indicated for the `--countErrors` option.

Examples

The following example shows how to verify the `sn` (surname) index for completeness and for errors. The messages shown are for a backend of type `pdb`. The output is similar for other backend types:

```
$ verify-index -b dc=example,dc=com -i sn --clean --countErrors
[20/05/2015:14:24:18 +0200] category=...PDBStorage seq=0 severity=INFO
 msg=The PDB storage for backend 'userRoot' initialized
 to use 57528 buffers of 16384 bytes (total 920448kb)
[20/05/2015:14:24:18 +0200] category=...pluggable.VerifyJob seq=1 severity=INFO
 msg=Checked 478 records and found 0 error(s) in 0 seconds
 (average rate 3594.0/sec)
[20/05/2015:14:24:18 +0200] category=...pluggable.VerifyJob seq=2 severity=FINE
 msg=Number of records referencing more than one entry: 224
[20/05/2015:14:24:18 +0200] category=...pluggable.VerifyJob seq=3 severity=FINE
 msg=Number of records that exceed the entry limit: 0
[20/05/2015:14:24:18 +0200] category=...pluggable.VerifyJob seq=4 severity=FINE
 msg=Average number of entries referenced is 2.00/record
[20/05/2015:14:24:18 +0200] category=...pluggable.VerifyJob seq=5 severity=FINE
 msg=Maximum number of entries referenced by any record is 32
```

Name

windows-service — register OpenDJ as a Windows Service

Synopsis

`windows-service` {options}

Description

This utility can be used to run OpenDJ directory server as a Windows Service.

Service Options

`-c, --cleanupService` ***`serviceName`***

Disable the service and clean up the windows registry information associated with the provided service name

`-d, --disableService`

Disable the server as a Windows service and stop the server

`-e, --enableService`

Enable the server as a Windows service

`-s, --serviceState`

Provide information about the state of the server as a Windows service

General Options

`-V, --version`

Display version information

`-?, -H, --help`

Display usage information

Exit Codes

0

The command completed successfully.

> 0

An error occurred.

Example

The following command registers OpenDJ directory server as a Windows Service.

```
C:\path\to\opendj\bat> windows-service.bat --enableService
```

After running this command, you can manage the service using Windows administration tools.

Glossary

Abandon operation	LDAP operation to stop processing of a request in progress, after which the server drops the connection without a reply to the client application.
Access control	Control to grant or to deny access to a resource.
Access control instruction (ACI)	Instruction added as a directory entry attribute for fine-grained control over what a given user or group member is authorized to do in terms of LDAP operations and access to user data. ACIs are implemented independently from privileges, which apply to administrative operations. See Also Privilege.
Access control list (ACL)	An access control list connects a user or group of users to one or more security entitlements. For example, users in group sales are granted the entitlement read-only to some financial data.
`access` log	Server log tracing the operations the server processes including timestamps, connection information, and information about the operation itself.
Account lockout	The act of making an account temporarily or permanently inactive after successive authentication failures.
Active user	A user that has the ability to authenticate and use the services, having valid credentials.
Add operation	LDAP operation to add a new entry or entries to the directory.

Anonymous	A user that does not need to authenticate, and is unknown to the system.
Anonymous bind	A bind operation using simple authentication with an empty DN and an empty password, allowing anonymous access such as reading public information.
Approximate index	Index is used to match values that "sound like" those provided in the filter.
Attribute	Properties of a directory entry, stored as one or more key-value pairs. Typical examples include the common name (`cn`) to store the user's full name and variations of the name, user ID (`uid`) to store a unique identifier for the entry, and `mail` to store email addresses.
`audit` log	Type of access log that dumps changes in LDIF.
Authentication	The process of verifying who is requesting access to a resource; the act of confirming the identity of a principal.
Authorization	The process of determining whether access should be granted to an individual based on information about that individual; the act of determining whether to grant or to deny a principal access to a resource.
Backend	Repository that stores directory data. Different implementations with different capabilities exist.
Binary copy	Binary backup archive of one directory server that can be restored on another directory server.
Bind operation	LDAP authentication operation to determine the client's identity in LDAP terms, the identity which is later used by the server to authorize (or not) access to directory data that the client wants to lookup or change.
Branch	The distinguished name (DN) of a non-leaf entry in the Directory Information Tree (DIT), and also that entry and all its subordinates taken together. Some administrative operations allow you to include or exclude branches by specifying the DN of the branch. See also Suffix.
Collective attribute	A standard mechanism for defining attributes that appear on all the entries in a particular subtree.
Compare operation	LDAP operation to compare a specified attribute value with the value stored on an entry in the directory.

Control	Information added to an LDAP message to further specify how an LDAP operation should be processed. OpenDJ supports many LDAP controls.
Database cache	Memory space set aside to hold database content.
`debug` log	Server log tracing details needed to troubleshoot a problem in the server.
Delete operation	LDAP operation to remove an existing entry or entries from the directory.
Directory	A directory is a network service which lists participants in the network such as users, computers, printers, and groups. The directory provides a convenient, centralized, and robust mechanism for publishing and consuming information about network participants.
Directory hierarchy	A directory can be organized into a hierarchy in order to make it easier to browse or manage. Directory hierarchies normally represent something in the physical world, such as organizational hierarchies or physical locations. For example, the top level of a directory may represent a company, the next level down divisions, the next level down departments, and down the hierarchy. Alternately, the top level may represent the world, the next level down countries, next states or provinces, and next cities.
Directory Information Tree (DIT)	A set of directory entries organized hierarchically in a tree structure, where the vertices are the entries and the arcs between vertices define relationships between entries
Directory manager	Default Root DN who has privileges to do full administration of the OpenDJ server, including bypassing access control evaluation, changing access controls, and changing administrative privileges. See Also Root DN.
Directory object	A directory object is an item in a directory. Example objects include users, user groups, computers, and more. Objects may be organized into a hierarchy and contain identifying attributes. See Also Entry.
Directory proxy server	Server that forwards LDAP requests to remote directory servers. A standalone directory proxy server does not store user data. See Also Directory server.
Directory server	Server application for centralizing information about network participants. A highly available directory service consists of multiple directory servers configured to replicate directory data. See Also Directory, Replication.

Directory Services Markup Language (DSML)	Standard language to access directory services using XML. DMSL v1 defined an XML mapping of LDAP objects, while DSMLv2 maps the LDAP Protocol and data model to XML.
Distinguished name (DN)	Fully qualified name for a directory entry, such as `uid=bjensen,ou=People,dc=example,dc=com`, built by concatenating the entry RDN (`uid=bjensen`) with the DN of the parent entry (`ou=People,dc=example,dc=com`).
Dynamic group	Group that specifies members using LDAP URLs.
Entry	As generic and hierarchical data stores, directories always contain different kinds of entries, either nodes (or containers) or leaf entries. An entry is an object in the directory, defined by one of more object classes and their related attributes. At startup, OpenDJ reports the number of entries contained in each suffix.
Entry cache	Memory space set aside to hold frequently accessed, large entries, such as static groups.
Equality index	Index used to match values that correspond exactly (though generally without case sensitivity) to the value provided in the search filter.
`errors` log	Server log tracing server events, error conditions, and warnings, categorized and identified by severity.
Export	Save directory data in an LDIF file.
Extended operation	Additional LDAP operation not included in the original standards. OpenDJ servers support several standard LDAP extended operations.
Extensible match index	Index for a matching rule other than approximate, equality, ordering, presence, substring or VLV, such as an index for generalized time.
External user	An individual that accesses company resources or services but is not working for the company. Typically a customer or partner.
Filter	An LDAP search filter is an expression that the server uses to find entries that match a search request, such as `(mail=*@example.com)` to match all entries having an email address in the example.com domain.
Group	Entry identifying a set of members whose entries are also in the directory.
Idle time limit	Defines how long OpenDJ allows idle connections to remain open.
Import	Read in and index directory data from an LDIF file.
Inactive user	An entry in the directory that once represented a user but which is now no longer able to be authenticated.

Index	Directory server backend feature to allow quick lookup of entries based on their attribute values. See Also Approximate index, Equality index, Extensible match index, Ordering index, Presence index, Substring index, Virtual list view (VLV) index, Index entry limit.
Index entry limit	When the number of entries that an index key points to exceeds the index entry limit, OpenDJ stops maintaining the list of entries for that index key.
Internal user	An individual who works within the company either as an employee or as a contractor.
LDAP Data Interchange Format (LDIF)	Standard, portable, text-based representation of directory content. See RFC 2849.
LDAP URL	LDAP Uniform Resource Locator such as `ldap://directory.example.com:389/dc=example,dc=com??sub?(uid=bjensen)`. See RFC 2255.
LDAPS	LDAP over SSL.
Lightweight Directory Access Protocol (LDAP)	A simple and standardized network protocol used by applications to connect to a directory, search for objects and add, edit or remove objects. See RFC 4510.
Lookthrough limit	Defines the maximum number of candidate entries OpenDJ considers when processing a search.
Matching rule	Defines rules for performing matching operations against assertion values. Matching rules are frequently associated with an attribute syntax and are used to compare values according to that syntax. For example, the `distinguishedNameEqualityMatch` matching rule can be used to determine whether two DNs are equal and can ignore unnecessary spaces around commas and equal signs, differences in capitalization in attribute names, and other discrepancies.
Modify DN operation	LDAP modification operation to request that the server change the distinguished name of an entry.
Modify operation	LDAP modification operation to request that the server change one or more attributes of an entry.
Naming context	Base DN under which client applications can look for user data.
Object class	Identifies entries that share certain characteristics. Most commonly, an entry's object classes define the attributes that must and may be present on the entry. Object classes are stored on entries as values of the `objectClass` attribute. Object classes are defined in the directory schema, and can be abstract (defining characteristics for other object

	classes to inherit), structural (defining the basic structure of an entry, one structural inheritance per entry), or auxiliary (for decorating entries already having a structural object class with other required and optional attributes).
Object identifier (OID)	String that uniquely identifies an object, such as `0.9.2342.19200300.100.1.1` for the user ID attribute or `1.3.6.1.4.1.1466.115.121.1.15` for `DirectoryString` syntax.
Operational attribute	An attribute that has a special (operational) meaning for the server, such as `pwdPolicySubentry` or `modifyTimestamp`.
Ordering index	Index used to match values for a filter that specifies a range.
Password policy	A set of rules regarding what sequence of characters constitutes an acceptable password. Acceptable passwords are generally those that would be too difficult for another user or an automated program to guess and thereby defeat the password mechanism. Password policies may require a minimum length, a mixture of different types of characters (lowercase, uppercase, digits, punctuation marks, and other characters), avoiding dictionary words or passwords based on the user's name, and other attributes. Password policies may also require that users not reuse old passwords and that users change their passwords regularly.
Password reset	Password change performed by a user other than the user who owns the entry.
Password storage scheme	Mechanism for encoding user passwords stored on directory entries. OpenDJ implements a number of password storage schemes.
Password validator	Mechanism for determining whether a proposed password is acceptable for use. OpenDJ implements a number of password validators.
Plugin	Java library with accompanying configuration that implements a feature through processing that is not essential to the core operation of an OpenDJ server. As the name indicates, plugins can be plugged in to an installed server for immediate configuration and use without recompiling the server. OpenDJ servers invoke plugins at specific points in the lifecycle of a client request. The OpenDJ configuration framework lets directory administrators manage plugins with the same tools used to manage the server.
Presence index	Index used to match the fact that an attribute is present on the entry, regardless of the value.

Principal	Entity that can be authenticated, such as a user, a device, or an application.
Privilege	Server configuration settings controlling access to administrative operations such as exporting and importing data, restarting the server, performing password reset, and changing the server configuration. Privileges are implemented independently from access control instructions (ACI), which apply to LDAP operations and user data. See Also Access control instruction (ACI).
Referential integrity	Ensuring that group membership remains consistent following changes to member entries.
`referint` log	Server log tracing referential integrity events, with entries similar to the errors log.
Referral	Reference to another directory location, which can be another directory server running elsewhere or another container on the same server, where the current operation can be processed.
Relative distinguished name (RDN)	Initial portion of a DN that distinguishes the entry from all other entries at the same level, such as `uid=bjensen` in `uid=bjensen,ou=People,dc=example,dc=com`.
Replication	Data synchronization that ensures all directory servers participating eventually share a consistent set of directory data.
`replication` log	Server log tracing replication events, with entries similar to the errors log.
Replication server	Server dedicated to transmitting replication messages. A standalone replication server does not store user data.
Root DN	A directory superuser, whose account is specific to a server under `cn=Root DNs,cn=config`. The default Root DN is Directory Manager. You can create additional Root DN accounts, each with different administrative privileges. See Also Directory manager, Privilege.
Root DSE	The directory entry with distinguished name "" (empty string), where DSE is an acronym for DSA-Specific Entry. DSA is an acronym for Directory Server Agent, a single directory server. The root DSE serves to expose information over LDAP about what the directory server supports in terms of LDAP controls, auth password schemes, SASL mechanisms, LDAP protocol versions, naming contexts, features, LDAP extended operations, and other information.

Schema	LDAP schema defines the object classes, attributes types, attribute value syntaxes, matching rules and other constrains on entries held by the directory server.
Search filter	See Filter.
Search operation	LDAP lookup operation where a client requests that the server return entries based on an LDAP filter and a base DN under which to search.
Simple authentication	Bind operation performed with a user's entry DN and user's password. Use simple authentication only if the network connection is secure.
Size limit	Sets the maximum number of entries returned for a search.
Static group	Group that enumerates member entries.
Subentry	An entry, such as a password policy entry, that resides with the user data but holds operational data, and is not visible in search results unless explicitly requested.
Substring index	Index used to match values specified with wildcards in the filter.
Suffix	The distinguished name (DN) of a root entry in the Directory Information Tree (DIT), and also that entry and all its subordinates taken together as a single object of administrative tasks such as export, import, indexing, and replication.
Task	Mechanism to provide remote access to server administrative functions. OpenDJ software supports tasks to back up and restore backends, to import and export LDIF files, and to stop and restart the server.
Time limit	Defines the maximum processing time OpenDJ devotes to a search operation.
Unbind operation	LDAP operation to release resources at the end of a session.
Unindexed search	Search operation for which no matching index is available. If no indexes are applicable, then the directory server potentially has to go through all entries to look for candidate matches. For this reason, the `unindexed-search` privilege, which allows users to request searches for which no applicable index exists, is reserved for the directory manager by default.
User	An entry that represents an individual that can be authenticated through credentials contained or referenced by its attributes. A user may represent an internal user or an external user, and may be an active user or an inactive user.
User attribute	An attribute for storing user data on a directory entry such as `mail` or `givenname`.

Virtual attribute	An attribute with dynamically generated values that appear in entries but are not persistently stored in the backend.
Virtual directory	An application that exposes a consolidated view of multiple physical directories over an LDAP interface. Consumers of the directory information connect to the virtual directory's LDAP service. Behind the scenes, requests for information and updates to the directory are sent to one or more physical directories where the actual information resides. Virtual directories enable organizations to create a consolidated view of information that for legal or technical reasons cannot be consolidated into a single physical copy.
Virtual list view (VLV) index	Browsing index designed to help the directory server respond to client applications that need, for example, to browse through a long list of results a page at a time in a GUI.
Virtual static group	OpenDJ group that lets applications see dynamic groups as what appear to be static groups.
X.500	A family of standardized protocols for accessing, browsing and maintaining a directory. X.500 is functionally similar to LDAP, but is generally considered to be more complex, and has consequently not been widely adopted.

Appendix A. REST to LDAP Configuration

OpenDJ offers two alternatives for access to directory data over HTTP:

- OpenDJ servers have an HTTP connection handler that exposes RESTful APIs to directory data over HTTP (or HTTPS). You configure an OpenDJ HTTP connection handler, and the HTTP endpoints that it serves, by using the **dsconfig** command. For each HTTP endpoint served by an HTTP connection handler that exposes your directory data, you configure mappings between JSON resources and LDAP entries.
- The OpenDJ REST to LDAP gateway runs in a Servlet container independent from the directory service. You configure the gateway to access the directory service by editing configuration files for the gateway web application.

Interface stability: Evolving (See Section I.2, "ForgeRock Product Interface Stability")

The files for configuring the gateway and the JSON resource to LDAP entry mappings are in JSON format.

In an OpenDJ server installation, the default location for the configuration files is under `/path/to/opendj/config`.

In a REST to LDAP gateway Servlet, the configuration files are under `WEB-INF/classes`.

The format and relative locations of the mapping files are the same for OpenDJ servers and the OpenDJ REST to LDAP gateway. Only OpenDJ REST to LDAP gateway, however, has files for configuring how the gateway connects to LDAP servers, how user identities extracted from HTTP requests map to LDAP user identities, and what LDAP features the gateway uses. In OpenDJ servers these capabilities are part of the server configuration.

The following list describes the configuration files, indicated by relative location under the configuration directory:

`config.json` **(gateway only)**

This file defines how the gateway connects to LDAP servers, and how user identities extracted from HTTP requests map to LDAP user identities.

For details, see Section A.1, "Gateway Configuration File".

`rest2ldap/rest2ldap.json` **(gateway only)**

This file defines which LDAP features the gateway uses.

For details, see Section A.2, "Gateway REST2LDAP Configuration File".

`rest2ldap/endpoints/`*`base-path`*`/`*`root-resource`*`.json`

These files define JSON resource to LDAP entry mappings.

For details about the configuration fields, see Section A.3, "Mapping Configuration File".

A.1. Gateway Configuration File

The `config.json` file for the REST to LDAP gateway can hold the configuration objects described in this section.

The order of the settings in the JSON file is not meaningful. Here, the order shown is that of the default configuration file:

`security`

Configures security parameters for establishing secure connections between the gateway (as a client) and the servers it contacts, such as LDAP directory servers and OAuth 2.0 authorization servers.

This field has the following properties:

`trustManager` **(optional)**

This setting configures how the servers are trusted. This setting is ignored for connections to LDAP servers if `connectionSecurity` is set to `none`:

- `file` means trust server certificates signed by a CA that is trusted according to the file-based truststore configured with `fileBasedTrustManager*` settings described below.
- `jvm` (default) means trust server certificates signed by a CA trusted by the Java environment.
- `trustAll` means blindly trust all server certificates.

Caution

This setting is not secure and makes man-in-the-middle attacks possible.

`fileBasedTrustManagerType` (optional)

If `trustManager` is set to `file`, then this setting configures the format for the data in the truststore file specified by the `fileBasedTrustManagerFile` setting. Formats include the following, though other implementations might be supported as well, depending on the Java environment:

- `JKS` (default) specifies Java Keystore format.
- `PKCS12` specifies Public-Key Cryptography Standards 12 format.

`fileBasedTrustManagerFile`

If `trustManager` is set to `file`, then this setting must specify the location of the truststore file.

Example: `/path/to/truststore`

`fileBasedTrustManagerPasswordFile` (optional)

If `trustManager` is set to `file`, then this setting specifies the file containing the truststore password.

Example: `/path/to/pinfile`

`keyManager` (optional)

This setting configures how the keys are managed for the gateway when the gateway is acting as a client of an LDAP server or OAuth 2.0 authorization server. The client keys are used to establish a secure connection to a server when the server requires client authentication.

This field can take the following values:

- `jvm` (default) means look for client keys in the default keystore for the Java environment.
- `file` means look for client keys in the specified keystore file, configured with the `fileBasedKeyManager*` settings.
- `pkcs11` means look for client keys in a PKCS #11 cryptographic token, where the PIN file is configured with the `pkcs11KeyManagerPasswordFile` setting described below.

`fileBasedKeyManagerFile`

If `keyManager` is set to `file`, then this setting must specify the keystore file.

Example: `/path/to/keystore`

`fileBasedKeyManagerPasswordFile` **(optional)**

If `keyManager` is set to `file`, then this setting specifies the file containing the keystore password.

Example: `/path/to/pinfile`

`fileBasedKeyManagerType` **(optional)**

If `keyManager` is set to `file`, then this setting specifies the format of the keystore specified by the `fileBasedKeyManagerFile` setting. Formats include the following, though other implementations might be supported as well, depending on the Java environment:

- `JKS` (default) specifies Java Keystore format.
- `PKCS12` specifies Public-Key Cryptography Standards 12 format.

`pkcs11KeyManagerPasswordFile` **(optional)**

If `keyManager` is set to `pkcs11`, then this setting specifies the file containing the PKCS #11 token password.

Example: `/path/to/pinfile`

`ldapConnectionFactories`

Configures how the gateway connects to LDAP servers. This entire configuration object applies only to the REST to LDAP gateway.

Configures at least a connection factory for unauthenticated connections that are used for bind requests. By default, also configures a factory for authenticated connections that are used for searches during authentication and for proxied authorization operations.

The default configuration is set to connect to a local directory server listening for LDAP connections on port 1389, authenticating as the root DN user `cn=Directory Manager`, with the password `password`:

`bind`

Configures the unauthenticated connection factory for bind operations:

`connectionSecurity` **(optional)**

Whether connections to LDAP servers should be secured by using SSL or StartTLS. The following values are supported:

- `none` (default) means connections use plain LDAP and are not secured.
- `ssl` means connections are secured using LDAPS.
- `startTLS` means connections are secured using LDAP and StartTLS.

If you set `connectionSecurity`, also review the `trustManager` and `fileBasedTrustManager*` settings in the `security` field.

`sslCertAlias` **(optional)**

If secure connections to LDAP servers require client authentication, this identifies the alias of the certificate to use for client authentication when establishing a secure connection.

If you uses this setting because client authentication is required, make sure the `keyManager` settings in the `security` field are properly configured.

If this field is missing, then the certificate is chosen during the SSL handshake.

Example: `client-cert`

`connectionPoolSize` **(optional)**

The gateway creates connection pools to the primary and secondary LDAP servers. The connection pools maintain up to `connectionPoolSize` connections to the servers.

Default: 24

`heartBeatIntervalSeconds` **(optional)**

The gateway tests its connections every `heartBeatIntervalSeconds` to detect whether the connection is still alive. The first test is performed immediately when the gateway gets a connection. Subsequent tests follow every `heartBeatIntervalSeconds`.

Default: 30 (seconds)

`heartBeatTimeoutMilliSeconds` **(optional)**

When the gateway tests a connection, if the heartbeat does not come back after `heartBeatTimeoutMilliSeconds` the connection is marked as closed.

Default: 500 (milliseconds)

`primaryLdapServers` **(required)**

The gateway accesses this array of LDAP servers before failing over to the secondary LDAP servers. These might be LDAP servers in the same data center, for example:

```
{
    "primaryLdapServers": [
        {
            "hostname": "local1.example.com",
            "port": 1389
        },
        {
            "hostname": "local2.example.com",
            "port": 1389
        }
    ]
}
```

By default, the gateway connects to the directory server listening on port 1389 on the local host.

`secondaryLdapServers` **(optional)**

The gateway accesses this array of LDAP servers if primary LDAP servers cannot be contacted. These might be LDAP servers in the same remote data center, for example:

```
{
    "secondaryLdapServers": [
        {
            "hostname": "remote1.example.com",
            "port": 1389
        },
        {
            "hostname": "remote2.example.com",
            "port": 1389
        }
    ]
}
```

No secondary LDAP servers are configured by default.

`root`

Configures the authenticated connection factory:

`inheritFrom` **(optional)**

Identifies the unauthenticated connection factory to inherit the settings from. If this connection factory does not inherit from another configuration object, then you must specify the configuration here.

Default: `bind`

`authentication` **(required)**

The gateway authenticates by simple bind using the credentials specified:

```
{
    "authentication": {
        "bindDn": "cn=Directory Manager",
        "password": "password"
    }
}
```

If the OAuth 2.0 authorization policy is configured for the gateway, then the directory service must be configured to allow the user configured here to perform proxied authorization.

`authorization`

Configures how authorization is performed for REST operations. This entire configuration object applies only to the REST to LDAP gateway.

The default configuration handles authorization by mapping HTTP Basic authentication credentials to LDAP bind credentials. User entries are `inetOrgPerson` entries expected to have `uid=`*`username`*, and expected to be found under `ou=people,dc=example,dc=com`.

The default configuration also allows alternative, HTTP header-based authentication in the style of OpenIDM software.

To protect passwords, configure HTTPS for the container where the REST to LDAP gateway runs.

This object has the following configuration fields:

`policies`

Which authorization policies are allowed, where the supported policies include:

- `anonymous`
- `basic` (HTTP Basic)
- `oauth2`

When more than one policy is specified, policies are applied in the following order:

1. If the client request has an `Authorization` header, and policies include `oauth2`, the server attempts to apply the OAuth 2.0 policy.
2. If the client request has an `Authorization` header, or has the custom credentials headers specified in the configuration, and policies includes `basic`, the server attempts to apply the Basic Auth policy.
3. Otherwise, if policies includes `anonymous`, and none of the previous policies apply, the server attempts to apply the policy for anonymous requests.

Default: `[ "basic" ]`

`anonymous`

Configuration for authorization when the HTTP connection to the gateway is not authenticated.

Operations are performed using connections from the specified factory:

`ldapConnectionFactory`

Factor providing LDAP connections to use for anonymous HTTP requests.

In effect, you add `"anonymous"` to the array of policies allowed without otherwise changing the default configuration, anonymous HTTP requests result in LDAP requests performed by Directory Manager. Take care to adjust this setting appropriately when allowing anonymous requests.

Default: `root`

`basic`

Configuration for authorization using HTTP Basic credentials.

The HTTP Basic credentials are mapped to LDAP credentials. The LDAP credentials are then used to bind to the directory service.

This object has the following configuration fields:

supportAltAuthentication

Whether to allow alternative, HTTP header-based authentication. If this is set to `true`, then the headers containing credentials are specified as the values for `altAuthenticationUsernameHeader` and `altAuthenticationPasswordHeader`, and the bind DN is resolved using a template.

Default: `true`

altAuthenticationUsernameHeader

The HTTP header containing the username for authentication when alternative, HTTP header-based authentication is allowed.

Default: `X-OpenIDM-Username`

altAuthenticationPasswordHeader

The HTTP header containing the password for authentication when alternative, HTTP header-based authentication is allowed.

Default: `X-OpenIDM-Password`

bind

How HTTP Basic credentials are mapped to LDAP credentials used to bind to the directory service.

The following values are supported:

- `search` (default) means the gateway performs a search based on the HTTP Basic user name to obtain the bind DN.
- `sasl-plain` means the gateway transforms the HTTP Basic user name to an authorization ID (authzid) using a template.
- `simple` means the HTTP Basic user name is the LDAP bind DN.

simple

How to reuse HTTP Basic credentials for an LDAP simple bind.

This object has the following configuration fields:

ldapConnectionFactory

The factory providing LDAP connections to the directory service.

Default: `bind`

bindDnTemplate

The template to produce the bind DN from the HTTP Basic user name.

A single occurrence of the string `{username}` is replaced in the template with the HTTP Basic user name.

For example, if the user name is also the UID of the LDAP entry, use `uid={username},ou=People,dc=example,dc=com`.

Default: `{username}`

`sasl-plain`

How to reuse HTTP Basic credentials for an LDAP SASL plain bind.

This object has the following configuration fields:

`ldapConnectionFactory`

The factory providing LDAP connections to the directory service.

Default: `bind`

`authzIdTemplate`

The template to produce the authorization ID from the HTTP Basic user name.

A single occurrence of the string `{username}` is replaced in the template with the HTTP Basic user name.

If the user name is also the authorization ID, use `u:{username}`.

If the user name is the LDAP bind DN, use `dn:{username}`.

`search`

How to reuse HTTP Basic credentials to find the bind DN for an LDAP simple bind.

This object has the following configuration fields:

`searchLdapConnectionFactory`

The factory providing LDAP connections to the directory service for the LDAP search operation.

Default: `root`

`bindLdapConnectionFactory`

The factory providing LDAP connections to the directory service for the LDAP bind operation that uses the bind DN returned by the search.

Default: `bind`

`baseDn`

The base DN for the LDAP search.

Example: `ou=People,dc=example,dc=com`.

`scope`

The scope for the LDAP search.

Use `sub` for a subtree search, `one` for a one-level search.

`filterTemplate`

The template for the filter of the LDAP search.

A single occurrence of the string `{username}` is replaced in the template with the HTTP Basic user name.

If the user name is also the UID, use `(&(uid={username})(objectClass=inetOrgPerson))`.

`oauth2`

Configuration for authorization based on OAuth 2.0, where the gateway plays the role of resource server.

This object has the following configuration fields:

`realm`

Realm associated with access tokens presented to the gateway.

`requiredScopes`

Array of OAuth 2.0 scopes that are required to allow access.

This array must not be empty.

Example: `[ "read", "write", "uid" ]`

`resolver`

How to resolve OAuth 2.0 access tokens presented to the gateway.

Supported values include the following:

- `cts` to resolve tokens in a directory service acting as a Core Token Service (CTS) store for OpenAM
- `openam` to send requests for token resolution to an OpenAM server
- `rfc7662` to send requests for token resolution to an RFC 7622-compliant server

Each access token resolution mechanism has its own configuration.

`accessTokenCache`

How to cache OAuth 2.0 token information to avoid repeating calls for access token resolution.

This object has the following configuration fields:

`enabled`

Whether to cache access token information obtained from the resolver.

Default: `false`

`cacheExpiration`

How long to cache information for a particular token if caching is enabled.

Default: `5 minutes`

`openam`

Configuration for resolving OAuth 2.0 tokens by a request to OpenAM.

This object has the following configuration fields:

`endpointUrl`

OpenAM URL for requests for token information, which depends on OpenAM's OAuth 2.0 authorization server configuration.

Example: `https://openam.example.com:8443/openam/oauth2/tokeninfo`

`sslCertAlias` **(optional)**

If secure connections to the authorization server require client authentication, this identifies the alias of the certificate to use for client authentication when establishing a secure connection.

If you uses this setting because client authentication is required, make sure the `keyManager` settings in the `security` field are properly configured.

If this field is missing, then the certificate is chosen during the SSL handshake.

Example: `client-cert`

`authzIdTemplate`

The template to produce the authorization ID from OAuth 2.0 token information.

A JSON pointer value in braces is replaced in the template with a field value from the JSON returned during token resolution.

This template must start with `u:` or `dn:`.

For example, if token resolution returns a JSON document where the value of the `uid` field is the UID of the user entry in the directory, you might use `u:{uid}` or `dn:{uid},ou=People,dc=example,dc=com`.

`rfc7662`

Configuration for resolving OAuth 2.0 tokens by a request to an RFC 7662-compliant authorization server.

RFC 7662, *OAuth 2.0 Token Introspection*, defines a standard method for resolving access tokens.

This object has the following configuration fields:

`endpointUrl`

Authorization server URL for requests for token information with HTTP Basic authentication for OAuth 2.0 clients.

Example: `https://as.example.com/introspect`

`sslCertAlias` **(optional)**

If secure connections to the authorization server require client authentication, this identifies the alias of the certificate to use for client authentication when establishing a secure connection.

If you uses this setting because client authentication is required, make sure the `keyManager` settings in the `security` field are properly configured.

If this field is missing, then the certificate is chosen during the SSL handshake.

Example: `client-cert`

`clientId`

OAuth 2.0 client identifier defined during registration with the authorization server.

`clientSecret`

OAuth 2.0 client secret defined during registration with the authorization server.

`authzIdTemplate`

The template to produce the authorization ID from OAuth 2.0 token information.

A JSON pointer value in braces is replaced in the template with a field value from the JSON returned during token resolution.

This template must start with `u:` or `dn:`.

For example, if token resolution returns a JSON document where the value of the `username` field is the UID of the user entry in the directory, you might use `u:{username}` or `dn:{username},ou=People,dc=example,dc=com`.

`cts`

Configuration for resolving OAuth 2.0 tokens when the directory service acts as OpenAM's CTS store.

OpenAM's CTS store is constrained to a specific layout. The `authzIdTemplate` must therefore use `{userName/0}` for the user identifier.

This mechanism makes it possible to resolve access tokens by making a request to the CTS directory service, without making a request to OpenAM. *This mechanism does*

not, however, ensure that the token requested will have already been replicated to the directory server where the request is routed.

This object has the following configuration fields:

`ldapConnectionFactory`

The factory providing LDAP connections used to obtain token information from the CTS directory service.

Default: `root`

`baseDn`

The base DN in the CTS directory service where tokens are found.

If the base DN configured for CTS in OpenAM is `dc=cts,dc=example,dc=com`, then use `ou=famrecords,ou=openam-session,ou=tokens,dc=cts,dc=example,dc=com`.

`authzIdTemplate`

The template to produce the authorization ID from OAuth 2.0 token information.

A JSON pointer value in braces is replaced in the template with a field value from the JSON returned during token resolution.

This template must start with `u:` or `dn:`.

In OpenAM CTS, the user name field is an array. For example, if the user name is the UID of the user entry, the use `u:{userName/0}` or `dn:{userName/0},ou=People,dc=example,dc=com`.

A.2. Gateway REST2LDAP Configuration File

The `rest2ldap/rest2ldap.json` for the REST to LDAP gateway can hold the configuration objects described in this section.

The order of the settings in the JSON file is not meaningful. Here, the order shown is that of the default configuration file:

`useMvcc`

Whether the gateway supports multi-version concurrency control (MVCC). If true, also specify an `mvccAttribute` to use for MVCC.

Default: `true`

`mvccAttribute`

The LDAP attribute whose value is used for MVCC. Before performing a write operation, the client application can check, for example, whether it is modifying the correct version of a resource by matching the value of the header `If-Match: value`.

Default: `etag`

`readOnUpdatePolicy`

The policy used to read an entry before it is deleted, or to read an entry after it is added or modified. One of the following:

- `controls`: (default) use RFC 4527 read-entry controls to reflect the state of the resource at the time the update was performed.

  The directory service must support RFC 4527.

- `disabled`: do not read the entry or return the resource on update.

- `search`: perform an LDAP search to retrieve the entry before deletion or after it is added or modified.

  The JSON resource returned might differ from the LDAP entry that was updated.

`useSubtreeDelete`

Whether to use the LDAP Subtree Delete request control (OID: `1.2.840.113556.1.4.805`) for LDAP delete operations resulting from delete operations on resources. Clients applications that request deletes for resources with children must have access to use the control.

If this setting is `true`, REST to LDAP attempts to use the control, but falls back to searching for and deleting children if the server rejects the request, because the control is not supported, for example.

Default: `true`

Set this to `false` if the directory server does not support the control.

`usePermissiveModify`

Whether to use the LDAP Permissive Modify request control (OID: `1.2.840.113556.1.4.1413`) for LDAP modify operations resulting from patch and update operations on resources.

Default: `true`

Set this to `false` when using the gateway if the directory server does not support the control.

A.3. Mapping Configuration File

The `rest2ldap/endpoints/`*`base-path`*`/`*`root-resource`*`.json` files define how JSON resources map to LDAP entries.

For each base path exposing a REST API, a *`base-path`* directory holds one or more *`root-resource`*.json files. In the OpenDJ server configuration, the Rest2ldap endpoint `base-path` must match the *`base-path`* directory name.

Each *root-resource*.json file defines mappings for a specific version of the API. The *root-resource* in the file name must match the name of the root resource defined in the file.

If there is more than one version of the API, then client applications must select the version by setting a version header:

```
Accept-API-Version: resource=version
```

If more than one version of the API is available, and the client application does not select the version by setting a version header, then the latest version is returned.

Here, *version* is the value of the `version` field in the mapping configuration file.

The file `rest2ldap/endpoints/api/example-v1.json` is delivered as an example mapping. This file has the following basic structure:

```
{
  "version": "1.0",         // Version for this API.
  "resourceTypes": {        // Resources for this API.
    "example-v1": {         // Root resource type. Name matches file basename.
      "subResources": {     // The base resource, at /api, is not defined.
        "users": {},        // The subresources at /api/users/ and
        "groups": {}        // /api/groups are defined, however.
      }
    },

    // In addition to the root resource type,
    // the example defines a number of other resource type schemas.
    // These are used to describe the resources exposed under the root resource.
    // In the example file, you can see how these are used for inheritance.
    "frapi:opendj:rest2ldap:object:1.0": {},    // Parent type of all objects.
    "frapi:opendj:rest2ldap:user:1.0": {},      // Basic user type, parent of
    "frapi:opendj:rest2ldap:posixUser:1.0": {}, // user with uid, gid, home dir.
    "frapi:opendj:rest2ldap:group:1.0": {}      // Basic group type.
  }
}
```

The following list describes the individual fields in more detail.

The order of the settings in the JSON file is not meaningful. Here, the order shown is that of the default example configuration file:

`version` (optional)

The version string for the root resource of this API.

Valid values are `*`, *`integer`*, and *`integer.integer`*, where *`integer`* is a positive decimal integer.

If the version is set, and the client application sets the request header `Accept-API-Version: resource=version`, The mapping with the matching *`version`* value is selected.

If more than one version of the API is available, and the client application does not select the version by setting a version header, then the latest version is returned.

Default: `*` (no version specified)

`resourceTypes` **(required)**

The map of resource type names to resource type definitions for this API.

One of the resource type name must match the basename of the mapping file. This resource is referred to as the *root resource* for this version of the API.

The value of a resource type is an object whose properties are described in Table A.1, "Resource Type Properties".

Table A.1. Resource Type Properties

Property	Description
`resourceTypeProperty` (string, required for inheritance)	Name of the resource type property that specifies the type of this resource. REST to LDAP uses this to determine the resource subtype when creating a resource. This points the mapper to the type of the resource. The specified property must be of type `resourceType`.
`properties` (map, optional)	Map of property names to property definitions. Unlike LDAP entries, JSON resources are not necessarily flat. You can define nested properties of type `object` that have their own properties. For details on properties configuration, see Table A.2, "Properties of Resource Type Properties Objects".
`subResources` (map, optional)	Map of subresource names to subresource definitions. The subresource names are URL templates. A URL template sets the relative URL template beneath which the subresources are located. If empty, the subresources are located directly beneath the parent resource. URL templates can set variables in braces `{}`. Any URL template variables will be substituted into the DN template. For example, suppose LDAP entries for devices are located under the following base DNs: • `ou=others,ou=devices,dc=example,dc=com` • `ou=pcs,ou=devices,dc=example,dc=com` • `ou=phones,ou=devices,dc=example,dc=com` • `ou=tablets,ou=devices,dc=example,dc=com` The subresource name `/{type}` would be substituted in actual paths with `/others`, `/pcs`, `/phones`, and `/tablets`. The DN template for the subresource would specify `ou={type},ou=devices,dc=example,dc=com` in

Property	Description
	order to locate the entries in the correct LDAP organizational unit. In the example, REST to LDAP substitutes `{type}` in the DN template with the type defined in the request URL path. For details on subresource configuration, see Table A.3, "Sub-Resource Properties".
`isAbstract` (boolean, optional)	Whether this is an abstract resource type used only for inheritance. Default: `false`
`superType` (string, optional)	Name of the resource type that this resource type extends. Resource types that extend another type inherit properties of the extended type, and inherit subresource definitions. Default: none. This resource type does not extend another type.
`objectClasses` (array, optional)	Names of the LDAP object classes that this type corresponds to. When an object of this type is created, these object class names are added to the list of object classes on the LDAP entry. The LDAP object classes are not shown in the JSON resource. Default: none.
`supportedActions` (array, optional)	Names of the common REST actions that this resource type supports. The names must match actions allowed on the resource in the underlying implementation. Default: none.
`includeAllUserAttributesByDefault` (boolean, optional)	Whether to include all LDAP user attributes as properties of the JSON resource. If `true`, the property names in the JSON resource match the attribute names in the LDAP entries. Default: `false`
`excludedDefaultUserAttributes` (array, optional)	Names of the LDAP user attributes to exclude from the JSON resource when `includeAllUserAttributesByDefault` is `true`. Default: none.

Table A.2. Properties of Resource Type Properties Objects

Property	Description
`type` (string, required)	Determines the type of the mapping property, and therefore which other properties the object has. The type must be one of the following: **`constant`** The property maps the JSON resource property to a fixed value specified by the `value` property.

Property	Description
	`json` The property value maps the JSON resource property to a `Json` syntax LDAP attribute. When the type is `json`, the mapping must specify an `ldapAttribute` property that specifies the `Json` syntax LDAP attribute. The mapping may have the following optional properties: • `defaultJsonValue` • `isMultiValued` • `isRequired` • `schema` • `writability` **`object`** The property value is a JSON object with its own type and mapping specified by the object's `properties`. **`reference`** The property maps a JSON field to an LDAP entry found by reference. This is useful for LDAP attributes that reference other entries, such as `manager`, and (group) `member`. When the type is `reference`, the mapping must have the following required properties. • `baseDn` • `ldapAttribute` • `mapper` • `primaryKey` The mapping may have the following optional properties. • `isMultiValued` • `isRequired` • `searchFilter` • `writability`

Property	Description
	`resourceType` The property value is the name of a resource type defined in this mapping file. The name of the property with this type should match the `resourceTypeProperty` name. For example, if `"resourceTypeProperty": "_schema"` then the following should be specified or inherited: `"_schema": { "type": "resourceType" }`. **`simple`** The property maps a JSON property to an LDAP attribute. Use simple mappings where the correspondence between JSON properties and LDAP attributes is one-to-one. When the type is `simple`, the mapping must specify an `ldapAttribute` property. The mapping may have the following optional properties. • `defaultJsonValue` • `isBinary` • `isMultiValued` • `isRequired` • `writability`
`baseDn`	Indicates the base LDAP DN under which to find entries referenced by the JSON resource. Base DN values can be literal values, such as `dc=example,dc=com`, and can also use the following notation: ***`{url-template}`*** The *`{url-template}`* used in the description of the URL to the resource is replaced with the literal value used in the request. For example, suppose the path defined for the resources is `/{tenant}/users` and the base DN is `ou=people,dc={tenant},dc=com`. For a request to `/example/users`, the base DN is `ou=people,dc=example,dc=com`. **`..`** The `..` refers to the relative parent RDN. This is like `..` in a file system path, where `..` refers to the parent directory. Keep in mind that file system paths are big endian, whereas DNs are little endian. You write `../../file-in-grandparent-directory`, but `cn=Child of Grandparent Entry,..,...`.

Property	Description

	The following excerpt from the default example configuration shows how this could be used to reference a manager's entry (the mapper configuration is not shown):
	`{ "manager": { "type": "reference", "ldapAttribute": "manager", "baseDn": "..", "primaryKey": "uid", "mapper": {} } }`
	In this case, if the current LDAP entry for the resource `uid=bjensen,ou=people,dc=example,dc=com`, then the base DN is `ou=people,dc=example,dc=com`.
	Another excerpt from the default example configuration shows a reference to group member entries (again, the mapper configuration is not shown):
	`{ "members": { "type": "reference", "ldapAttribute": "uniqueMember", "baseDn": "ou=people,..,..", "primaryKey": "uid", "isMultiValued": true, "mapper": {} } }`
	In this case, if the current LDAP entry for the resource `cn=Directory Administrators,ou=groups,dc=example,dc=com`, then the base DN is `ou=people,dc=example,dc=com`.
	Notice a limitation in this reference to group member entries: all group members must be people; the configuration does not handle nested groups and other types of members.

`defaultJsonValue`	Sets the JSON value if no corresponding LDAP attribute is present. No default is set if this is omitted.
`isBinary`	Whether the underlying LDAP attribute holds a binary value, such as a JPEG photo or a digital certificate. If `true`, the JSON property takes the base64-encoded value. Binary values can also be handled directly as described in Section 3.11, "Working With Alternative Content Types" in the *Developer's Guide*.

Property	Description
	Default: `false`.
`isMultiValued`	Whether the JSON resource property can take an array value. Most LDAP attributes can take multiple values. A literal-minded mapping from LDAP to JSON would therefore be full of array properties, many with only one value. To minimize inconvenience, REST to LDAP generally returns single value scalars, even when the underlying LDAP attribute is multi-valued. If this property is omitted or set to `false`, then the JSON resource contains the first value returned for multi-valued LDAP attributes with more than value. If this property is `true`, then if the LDAP attribute only has one value, it is returned as a scalar. If the LDAP attribute has more than one value, the values are returned in an array. Default: `false`
`isRequired`	`true` means the LDAP attribute is mandatory and must be provided to create the resource; `false` means it is optional. Default: `false`.
`ldapAttribute`	Specifies the LDAP attribute in the entry underlying the JSON resource whose value points to the referenced entry. For example, a `manager` attribute value is the DN of the manager's entry. Default: use the name of the JSON property. For example, the JSON property `description` maps to the LDAP attribute `description` by default.
`mapper`	Describes how the referenced entry content maps to the content of this JSON property. A mapper object is a properties object of its own.
`primaryKey`	Indicates which LDAP attribute in the mapper holds the primary key to the referenced entry.
`schema`	Specifies a JSON Schema that applies values of type `json`. Default: No schema is specified; values may be arbitrary JSON.
`searchFilter`	Specifies the LDAP filter to use to search for the referenced entry. Default: `"(objectClass=*)"`
`value`	Use with `"type": "constant"` to specify the constant value.
`writability`	Indicates whether the mapping supports updates. The `writability` property takes one of the following values:

Property	Description
	• `createOnly`: This attribute can be set only when the entry is created. Attempts to update this attribute thereafter result in errors. • `createOnlyDiscardWrites`: This attribute can be set only when the entry is created. Attempts to update this attribute thereafter do not result in errors. Instead the update value is discarded. • `readOnly`: This attribute cannot be written. Attempts to write this attribute result in errors. • `readOnlyDiscardWrites`: This attribute cannot be written. Attempts to write this attribute do not result in errors. Instead the value to write is discarded. • `readWrite`: (default) This attribute can be set at creation and updated thereafter.

Table A.3. Sub-Resource Properties

Property	Description
`type` (string, required)	The type of this subresource, either `collection` or `singleton`. A collection subresource is a container for other resources, which can be created, read, updated, deleted, patched, and queried. A collection definition has the following required properties: • `namingStrategy` • `resource` A collection definition has the following optional properties: • `dnTemplate` • `glueObjectClasses` • `isReadOnly` A singleton subresource is a resource with no children. A singleton definition has the following required properties: • `resource` A singleton definition has the following optional properties: • `dnTemplate` • `isReadOnly`
`dnTemplate` (string, optional)	Sets the relative DN template beneath which the subresource LDAP entries are located.

Property	Description
	If this is an empty string, the LDAP entries are located directly beneath the parent LDAP entry. DN templates can use variables in braces `{}`. DN template variables are substituted using values extracted from the URL template. Default: empty string
`glueObjectClasses` (array, required if the DN template contains one or more RDNs)	Specifies one or more LDAP object class names associated with any intermediate "glue" entries forming the DN template. Default: no object classes are specified
`isReadOnly` (boolean, optional)	Whether this resource is read-only. Default: `false`

`namingStrategy` (object, required)	Specifies the approach used to map LDAP entry names to JSON resources.
	
	LDAP entries mapped to JSON resources must be immediate subordinates of the mapping's `baseDn`.
	
	The following naming strategies are supported:
	
	• RDN and resource ID are both derived from a single user attribute in the LDAP entry, as in the following example, where the `uid` attribute is the RDN and its value is the JSON resource ID:
	
	`{ "namingStrategy": { "type": "clientDnNaming", "dnAttribute": "uid" } }`
	
	• RDN and resource ID are derived from separate user attributes in the LDAP entry, as in the following example, where the RDN attribute is `uid`, but the JSON resource ID is the value of the `mail` attribute:
	
	`{ "namingStrategy": { "type": "clientNaming", "dnAttribute": "uid", "idAttribute": "mail" } }`
	
	• RDN is derived from a user attribute and the resource ID from an operational attribute in the LDAP entry, as in the following example, where the RDN attribute is `uid`, but the JSON resource ID is the value of the `entryUUID` operational attribute:

Property	Description

	<pre>{
	    "namingStrategy": {
	        "type": "serverNaming",
	        "dnAttribute": "uid",
	        "idAttribute": "entryUUID"
	    }
	}</pre>

`resource` (string, required)	Specifies the resource type name of the subresource. A collection can contain objects with different subresource types as long as all types inherit from the same super type. In that case, set `resource` to the super type name.

Appendix B. LDAP Result Codes

An operation result code as defined in RFC 4511 section 4.1.9 is used to indicate the final status of an operation. If a server detects multiple errors for an operation, only one result code is returned. The server should return the result code that best indicates the nature of the error encountered. Servers may return substituted result codes to prevent unauthorized disclosures.

Table B.1. OpenDJ LDAP Result Codes

Result Code	Name	Description
-1	Undefined	The result code that should only be used if the actual result code has not yet been determined. Despite not being a standard result code, it is an implementation of the null object design pattern for this type.
0	Success	The result code that indicates that the operation completed successfully.
1	Operations Error	The result code that indicates that the operation is not properly sequenced with relation to other operations (of same or different type). For example, this code is returned if the client attempts to StartTLS [RFC4346] while there are other uncompleted operations or if a TLS layer was already installed.
2	Protocol Error	The result code that indicates that the client sent a malformed or illegal request to the server.
3	Time Limit Exceeded	The result code that indicates that a time limit was exceeded while attempting to process the request.
4	Size Limit Exceeded	The result code that indicates that a size limit was exceeded while attempting to process the request.

Result Code	Name	Description
5	Compare False	The result code that indicates that the attribute value assertion included in a compare request did not match the targeted entry.
6	Compare True	The result code that indicates that the attribute value assertion included in a compare request did match the targeted entry.
7	Authentication Method Not Supported	The result code that indicates that the requested authentication attempt failed because it referenced an invalid SASL mechanism.
8	Strong Authentication Required	The result code that indicates that the requested operation could not be processed because it requires that the client has completed a strong form of authentication.
10	Referral	The result code that indicates that a referral was encountered. Strictly speaking this result code should not be exceptional since it is considered as a "success" response. However, referrals should occur rarely in practice and, when they do occur, should not be ignored since the application may believe that a request has succeeded when, in fact, nothing was done.
11	Administrative Limit Exceeded	The result code that indicates that processing on the requested operation could not continue because an administrative limit was exceeded.
12	Unavailable Critical Extension	The result code that indicates that the requested operation failed because it included a critical extension that is unsupported or inappropriate for that request.
13	Confidentiality Required	The result code that indicates that the requested operation could not be processed because it requires confidentiality for the communication between the client and the server.
14	SASL Bind in Progress	The result code that should be used for intermediate responses in multi-stage SASL bind operations.
16	No Such Attribute	The result code that indicates that the requested operation failed because it targeted an attribute or attribute value that did not exist in the specified entry.
17	Undefined Attribute Type	The result code that indicates that the requested operation failed because it referenced an attribute that is not defined in the server schema.
18	Inappropriate Matching	The result code that indicates that the requested operation failed because it attempted to perform an inappropriate type of matching against an attribute.

Result Code	Name	Description
19	Constraint Violation	The result code that indicates that the requested operation failed because it would have violated some constraint defined in the server.
20	Attribute or Value Exists	The result code that indicates that the requested operation failed because it would have resulted in a conflict with an existing attribute or attribute value in the target entry.
21	Invalid Attribute Syntax	The result code that indicates that the requested operation failed because it violated the syntax for a specified attribute.
32	No Such Entry	The result code that indicates that the requested operation failed because it referenced an entry that does not exist.
33	Alias Problem	The result code that indicates that the requested operation failed because it attempted to perform an illegal operation on an alias.
34	Invalid DN Syntax	The result code that indicates that the requested operation failed because it would have resulted in an entry with an invalid or malformed DN.
36	Alias Dereferencing Problem	The result code that indicates that a problem was encountered while attempting to dereference an alias for a search operation.
48	Inappropriate Authentication	The result code that indicates that an authentication attempt failed because the requested type of authentication was not appropriate for the targeted entry.
49	Invalid Credentials	The result code that indicates that an authentication attempt failed because the user did not provide a valid set of credentials.
50	Insufficient Access Rights	The result code that indicates that the client does not have sufficient permission to perform the requested operation.
51	Busy	The result code that indicates that the server is too busy to process the requested operation.
52	Unavailable	The result code that indicates that either the entire server or one or more required resources were not available for use in processing the request.
53	Unwilling to Perform	The result code that indicates that the server is unwilling to perform the requested operation.
54	Loop Detected	The result code that indicates that a referral or chaining loop was detected while processing the request.

Result Code	Name	Description
60	Sort Control Missing	The result code that indicates that a search request included a VLV request control without a server-side sort control.
61	Offset Range Error	The result code that indicates that a search request included a VLV request control with an invalid offset.
64	Naming Violation	The result code that indicates that the requested operation failed because it would have violated the server's naming configuration.
65	Object Class Violation	The result code that indicates that the requested operation failed because it would have resulted in an entry that violated the server schema.
66	Not Allowed on Non-Leaf	The result code that indicates that the requested operation is not allowed for non-leaf entries.
67	Not Allowed on RDN	The result code that indicates that the requested operation is not allowed on an RDN attribute.
68	Entry Already Exists	The result code that indicates that the requested operation failed because it would have resulted in an entry that conflicts with an entry that already exists.
69	Object Class Modifications Prohibited	The result code that indicates that the operation could not be processed because it would have modified the objectclasses associated with an entry in an illegal manner.
71	Affects Multiple DSAs	The result code that indicates that the operation could not be processed because it would impact multiple DSAs or other repositories.
76	Virtual List View Error	The result code that indicates that the operation could not be processed because there was an error while processing the virtual list view control.
80	Other	The result code that should be used if no other result code is appropriate.
81	Server Connection Closed	The client-side result code that indicates that the server is down. This is for client-side use only and should never be transferred over protocol.
82	Local Error	The client-side result code that indicates that a local error occurred that had nothing to do with interaction with the server. This is for client-side use only and should never be transferred over protocol.
83	Encoding Error	The client-side result code that indicates that an error occurred while encoding a request to send to the server. This is for client-side use only and should never be transferred over protocol.
84	Decoding Error	The client-side result code that indicates that an error occurred while decoding a response from the server.

Result Code	Name	Description
		This is for client-side use only and should never be transferred over protocol.
85	Client-Side Timeout	The client-side result code that indicates that the client did not receive an expected response in a timely manner. This is for client-side use only and should never be transferred over protocol.
86	Unknown Authentication Mechanism	The client-side result code that indicates that the user requested an unknown or unsupported authentication mechanism. This is for client-side use only and should never be transferred over protocol.
87	Filter Error	The client-side result code that indicates that the filter provided by the user was malformed and could not be parsed. This is for client-side use only and should never be transferred over protocol.
88	Cancelled by User	The client-side result code that indicates that the user cancelled an operation. This is for client-side use only and should never be transferred over protocol.
89	Parameter Error	The client-side result code that indicates that there was a problem with one or more of the parameters provided by the user. This is for client-side use only and should never be transferred over protocol.
90	Out of Memory	The client-side result code that indicates that the client application was not able to allocate enough memory for the requested operation. This is for client-side use only and should never be transferred over protocol.
91	Connect Error	The client-side result code that indicates that the client was not able to establish a connection to the server. This is for client-side use only and should never be transferred over protocol.
92	Operation Not Supported	The client-side result code that indicates that the user requested an operation that is not supported. This is for client-side use only and should never be transferred over protocol.
93	Control Not Found	The client-side result code that indicates that the client expected a control to be present in the response from the server but it was not included. This is for client-side use only and should never be transferred over protocol.
94	No Results Returned	The client-side result code that indicates that the requested single entry search operation or read operation failed because the Directory Server did not return any matching entries. This is for client-side use only and should never be transferred over protocol.
95	Unexpected Results Returned	The client-side result code that the requested single entry search operation or read operation failed

Result Code	Name	Description
		because the Directory Server returned multiple matching entries (or search references) when only a single matching entry was expected. This is for client-side use only and should never be transferred over protocol.
96	Referral Loop Detected	The client-side result code that indicates that the client detected a referral loop caused by servers referencing each other in a circular manner. This is for client-side use only and should never be transferred over protocol.
97	Referral Hop Limit Exceeded	The client-side result code that indicates that the client reached the maximum number of hops allowed when attempting to follow a referral (i.e., following one referral resulted in another referral which resulted in another referral and so on). This is for client-side use only and should never be transferred over protocol.
118	Canceled	The result code that indicates that a cancel request was successful, or that the specified operation was canceled.
119	No Such Operation	The result code that indicates that a cancel request was unsuccessful because the targeted operation did not exist or had already completed.
120	Too Late	The result code that indicates that a cancel request was unsuccessful because processing on the targeted operation had already reached a point at which it could not be canceled.
121	Cannot Cancel	The result code that indicates that a cancel request was unsuccessful because the targeted operation was one that could not be canceled.
122	Assertion Failed	The result code that indicates that the filter contained in an assertion control failed to match the target entry.
123	Authorization Denied	The result code that should be used if the server will not allow the client to use the requested authorization.
16,654	No Operation	The result code that should be used if the server did not actually complete processing on the associated operation because the request included the LDAP No-Op control.

Appendix C. File Layout

OpenDJ software installs and creates the following files and directories. The following list is not meant to be exhaustive:

`legal-notices`

License information

`Uninstall.app`

Mac OS X GUI for removing server software

`bak`

Directory for saving backup files

`bat`

Windows command-line tools and control panel

`bin`

UNIX/Linux/Mac OS X command-line tools and control panel

`changelogDb`

Backend data for the external change log when using replication

`classes`

Directory added to the server `CLASSPATH`, permitting individual classes to be patched

`config`

OpenDJ server configuration and schema, PKI stores, LDIF generation templates, resources for upgrade

`config/MakeLDIF`

Templates for use with the **makeldif** LDIF generation tool

`config/audit-handlers`

Templates for configuring external Common Audit event handlers

`config/config.ldif`

LDIF representation of current OpenDJ server configuration

Use the **dsconfig** command to edit OpenDJ server configuration.

`common-passwords.txt`

List of common passwords used to check password strength

`config/java.properties`

JVM settings for OpenDJ server and tools

`config/schema`

LDAP schema definition files

`config/tasks.ldif`

Data used by task scheduler backend so that scheduled tasks and recurring tasks persist after server restart

`config/tools.properties`

Default settings for command-line tools

Use as a template when creating an `~/.opendj/tools.properties` file.

`config/upgrade`

Resources used by the upgrade command to move to the next server version

`config/wordlist.txt`

List of words used to check password strength

`db`

Backend database files for persistent, indexed backends that hold user data

`example-plugin.zip`

Sample OpenDJ plugin code. Custom plugins are meant to be installed in `lib/extensions`.

`extlib`

Directory where you put optional additional .jar files that are required for your deployment and that are not delivered with the server

If the instance path is not the same as the binaries, make sure you place the files in *instance-path/extlib/*.

import-tmp

Used when importing data into OpenDJ

instance.loc

Pointer to the server on the file system, provided for package installations where the program files are separate from the server instance files

ldif

Directory for saving LDIF export files

lib

Scripts and libraries provided by the server

For additional .jar files, use `extlib`.

lib/extensions

Directory to hold server plugins

locks

Directory to hold lock files used when the server is running to prevent backends from accidentally being used by more than one server process

logs

Access, errors, audit, and replication logs

logs/server.pid

Contains the process ID for a running server

setup

UNIX setup utility

setup.bat

Windows setup utility

template

Template files for a server instance

upgrade

UNIX utility for upgrading OpenDJ servers

upgrade.bat

Windows utility for upgrading OpenDJ servers

Appendix D. Ports Used

OpenDJ server software uses the TCP/IP ports described in Table D.1, "Server Ports".

Table D.1. Server Ports

Protocols	Conventional Ports	Active by Default?	Description
LDAP	389	No	Port for cleartext LDAP requests; also used to request StartTLS for a secure connection. The reserved LDAP port number is 389. Interactive setup initially suggests this port number. If the initially suggested port is not free or cannot be used due to lack of privileges, interactive setup adds 1000 to the port number and tries again, repeatedly adding 1000 until a free port is found. If LDAP is used, leave this port open to client applications.
LDAPS	636	No	Port for secure LDAPS requests. The standard LDAPS port number is 636. Interactive setup initially suggests this port number. If the initially suggested port is not free or cannot be used due to lack of privileges, interactive setup adds 1000 to the port number and tries again, repeatedly adding 1000 until a free port is found.

Protocols	Conventional Ports	Active by Default?	Description
			If LDAPS is used, leave this port open to client applications.
HTTP, HTTPS	80, 443	No	Port for HTTP client requests, such as RESTful API calls. The standard HTTP port number is 80. The standard HTTPS port number is 443. Interactive setup initially suggests 8080 and 8443 instead. If an initially suggested port is not free or cannot be used due to lack of privileges, interactive setup adds 1000 to the port number and tries again, repeatedly adding 1000 until a free port is found. If HTTP or HTTPS is used, leave this port open to client applications. For production deployments, use HTTPS instead of HTTP.
Server administration	4444	Yes	Port for administrative requests, such as requests from the **dsconfig** command. Interactive setup initially suggests 4444. If an initially suggested port is not free or cannot be used due to lack of privileges, interactive setup adds 1000 to the port number and tries again, repeatedly adding 1000 until a free port is found. Initial setup secures access to this port.
Directory data replication	8989	No	Port for replication requests, using the OpenDJ-specific replication protocol. Interactive setup initially suggests 8989. If an initially suggested port is not free or cannot be used due to lack of privileges, interactive setup adds 1000 to the port number and tries again, repeatedly adding 1000 until a free port is found. If replication is used, leave this port open to other replicas. For production deployments, secure access to this port.
JMX	1689	No	Port for Java Management eXtension requests (1689), and JMX RMI requests.

Protocols	Conventional Ports	Active by Default?	Description
			The default setting for the JMX RMI port is `0`, meaning the service chooses a port of its own. This can be configured using the JMX connection handler `rmi-port` setting If used in production deployments, secure access to this port.
SNMP	161, 162	No	Reserved ports are 161 for regular SNMP requests and 162 for traps. If used in production deployments, secure access to these ports.

Appendix E. Standards, RFCs, & Internet-Drafts

OpenDJ 5 software implements the following RFCs, Internet-Drafts, and standards:

RFC 1274: The COSINE and Internet X.500 Schema

X.500 Directory Schema, or Naming Architecture, for use in the COSINE and Internet X.500 pilots.

RFC 1321: The MD5 Message-Digest Algorithm

MD5 message-digest algorithm that takes as input a message of arbitrary length and produces as output a 128-bit "fingerprint" or "message digest" of the input.

RFC 1777: Lightweight Directory Access Protocol (LDAPv2)

Provide access to the X.500 Directory while not incurring the resource requirements of the Directory Access Protocol.

Classified as an Historic document.

RFC 1778: The String Representation of Standard Attribute Syntaxes

Defines the requirements that must be satisfied by encoding rules used to render X.500 Directory attribute syntaxes into a form suitable for use in the LDAP, then defines the encoding rules for the standard set of attribute syntaxes.

Classified as an Historic document.

RFC 1779: A String Representation of Distinguished Names

Defines a string format for representing names, which is designed to give a clean representation of commonly used names, whilst being able to represent any distinguished name.

Classified as an Historic document.

RFC 2079: Definition of an X.500 Attribute Type and an Object Class to Hold Uniform Resource Identifiers (URIs)

Defines a new attribute type and an auxiliary object class to allow URIs, including URLs, to be stored in directory entries in a standard way.

RFC 2222: Simple Authentication and Security Layer (SASL)

Describes a method for adding authentication support to connection-based protocols.

RFC 2246: The TLS Protocol Version 1.0

Specifies Version 1.0 of the Transport Layer Security protocol.

RFC 2247: Using Domains in LDAP/X.500 Distinguished Names

Defines an algorithm by which a name registered with the Internet Domain Name Service can be represented as an LDAP distinguished name.

RFC 2251: Lightweight Directory Access Protocol (v3)

Describes a directory access protocol designed to provide access to directories supporting the X.500 models, while not incurring the resource requirements of the X.500 Directory Access Protocol.

RFC 2252: Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions

Defines a set of syntaxes for LDAPv3, and the rules by which attribute values of these syntaxes are represented as octet strings for transmission in the LDAP protocol.

RFC 2253: Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names

Defines a common UTF-8 format to represent distinguished names unambiguously.

RFC 2254: The String Representation of LDAP Search Filters

Defines the string format for representing names, which is designed to give a clean representation of commonly used distinguished names, while being able to represent any distinguished name.

RFC 2255: The LDAP URL Format

Describes a format for an LDAP Uniform Resource Locator.

RFC 2256: A Summary of the X.500(96) User Schema for use with LDAPv3

Provides an overview of the attribute types and object classes defined by the ISO and ITU-T committees in the X.500 documents, in particular those intended for use by directory clients.

RFC 2307: An Approach for Using LDAP as a Network Information Service

Describes an experimental mechanism for mapping entities related to TCP/IP and the UNIX system into X.500 entries so that they may be resolved with the Lightweight Directory Access Protocol.

RFC 2377: Naming Plan for Internet Directory-Enabled Applications

Proposes a new directory naming plan that leverages the strengths of the most popular and successful Internet naming schemes for naming objects in a hierarchical directory.

RFC 2696: LDAP Control Extension for Simple Paged Results Manipulation

Allows a client to control the rate at which an LDAP server returns the results of an LDAP search operation.

RFC 2713: Schema for Representing Java(tm) Objects in an LDAP Directory

Defines a common way for applications to store and retrieve Java objects from the directory.

RFC 2714: Schema for Representing CORBA Object References in an LDAP Directory

Define a common way for applications to store and retrieve CORBA object references from the directory.

RFC 2739: Calendar Attributes for vCard and LDAP

Defines a mechanism to locate a user calendar and free/busy time using the LDAP protocol.

RFC 2798: Definition of the inetOrgPerson LDAP Object Class

Define an object class called inetOrgPerson for use in LDAP and X.500 directory services that extends the X.521 standard organizationalPerson class.

RFC 2829: Authentication Methods for LDAP

Specifies particular combinations of security mechanisms which are required and recommended in LDAP implementations.

RFC 2830: Lightweight Directory Access Protocol (v3): Extension for Transport Layer Security

Defines the "Start Transport Layer Security (TLS) Operation" for LDAP.

RFC 2849: The LDAP Data Interchange Format (LDIF) - Technical Specification

Describes a file format suitable for describing directory information or modifications made to directory information.

RFC 2891: LDAP Control Extension for Server Side Sorting of Search Results

Describes two LDAPv3 control extensions for server-side sorting of search results.

RFC 2926: Conversion of LDAP Schemas to and from SLP Templates

Describes a procedure for mapping between Service Location Protocol service advertisements and lightweight directory access protocol descriptions of services.

RFC 3045: Storing Vendor Information in the LDAP root DSE

Specifies two Lightweight Directory Access Protocol attributes, vendorName and vendorVersion that MAY be included in the root DSA-specific Entry (DSE) to advertise vendor-specific information.

RFC 3062: LDAP Password Modify Extended Operation

Describes an LDAP extended operation to allow modification of user passwords which is not dependent upon the form of the authentication identity nor the password storage mechanism used.

RFC 3112: LDAP Authentication Password Schema

Describes schema in support of user/password authentication in a LDAP directory including the authPassword attribute type. This attribute type holds values derived from the user's password(s) (commonly using cryptographic strength one-way hash).

RFC 3296: Named Subordinate References in Lightweight Directory Access Protocol (LDAP) Directories

Details schema and protocol elements for representing and managing named subordinate references in Lightweight Directory Access Protocol (LDAP) Directories.

RFC 3377: Lightweight Directory Access Protocol (v3): Technical Specification

Specifies the set of RFCs comprising the Lightweight Directory Access Protocol Version 3 (LDAPv3), and addresses the "IESG Note" attached to RFCs 2251 through 2256.

RFC 3383: Internet Assigned Numbers Authority (IANA) Considerations for the Lightweight Directory Access Protocol (LDAP)

Provides procedures for registering extensible elements of the Lightweight Directory Access Protocol (LDAP).

RFC 3546: Transport Layer Security (TLS) Extensions

Describes extensions that may be used to add functionality to Transport Layer Security.

RFC 3671: Collective Attributes in the Lightweight Directory Access Protocol (LDAP)

Summarizes the X.500 information model for collective attributes and describes use of collective attributes in LDAP.

RFC 3672: Subentries in the Lightweight Directory Access Protocol (LDAP)

Adapts X.500 subentries mechanisms for use with the Lightweight Directory Access Protocol (LDAP).

RFC 3673: Lightweight Directory Access Protocol version 3 (LDAPv3): All Operational Attributes

Describes an LDAP extension which clients may use to request the return of all operational attributes.

RFC 3674: Feature Discovery in Lightweight Directory Access Protocol (LDAP)

Introduces a general mechanism for discovery of elective features and extensions which cannot be discovered using existing mechanisms.

RFC 3712: Lightweight Directory Access Protocol (LDAP): Schema for Printer Services

Defines a schema, object classes and attributes, for printers and printer services, for use with directories that support Lightweight Directory Access Protocol v3 (LDAP).

RFC 3771: Lightweight Directory Access Protocol (LDAP) Intermediate Response Message

Defines and describes the IntermediateResponse message, a general mechanism for defining single-request/multiple-response operations in Lightweight Directory Access Protocol.

RFC 3829: Lightweight Directory Access Protocol (LDAP) Authorization Identity Request and Response Controls

Extends the Lightweight Directory Access Protocol bind operation with a mechanism for requesting and returning the authorization identity it establishes.

RFC 3876: Returning Matched Values with the Lightweight Directory Access Protocol version 3 (LDAPv3)

Describes a control for the Lightweight Directory Access Protocol version 3 that is used to return a subset of attribute values from an entry.

RFC 3909: Lightweight Directory Access Protocol (LDAP) Cancel Operation

Describes a Lightweight Directory Access Protocol extended operation to cancel (or abandon) an outstanding operation, with a response to indicate the outcome of the operation.

RFC 4346: The Transport Layer Security (TLS) Protocol Version 1.1

Specifies Version 1.1 of the Transport Layer Security protocol.

RFC 4370: Lightweight Directory Access Protocol (LDAP) Proxied Authorization Control

Defines the Proxy Authorization Control, that allows a client to request that an operation be processed under a provided authorization identity instead of under the current authorization identity associated with the connection.

RFC 4403: Lightweight Directory Access Protocol (LDAP) Schema for Universal Description, Discovery, and Integration version 3 (UDDIv3)

Defines the Lightweight Directory Access Protocol schema for representing Universal Description, Discovery, and Integration data types in an LDAP directory.

RFC 4422: Simple Authentication and Security Layer (SASL)

Describes a framework for providing authentication and data security services in connection-oriented protocols via replaceable mechanisms.

RFC 4505: Anonymous Simple Authentication and Security Layer (SASL) Mechanism

Describes a new way to provide anonymous login is needed within the context of the Simple Authentication and Security Layer framework.

RFC 4510: Lightweight Directory Access Protocol (LDAP): Technical Specification Road Map

Provides a road map of the LDAP Technical Specification.

RFC 4511: Lightweight Directory Access Protocol (LDAP): The Protocol

Describes the protocol elements, along with their semantics and encodings, of the Lightweight Directory Access Protocol.

RFC 4512: Lightweight Directory Access Protocol (LDAP): Directory Information Models

Describes the X.500 Directory Information Models as used in LDAP.

RFC 4513: Lightweight Directory Access Protocol (LDAP): Authentication Methods and Security Mechanisms

Describes authentication methods and security mechanisms of the Lightweight Directory Access Protocol.

RFC 4514: Lightweight Directory Access Protocol (LDAP): String Representation of Distinguished Names

Defines the string representation used in the Lightweight Directory Access Protocol to transfer distinguished names.

RFC 4515: Lightweight Directory Access Protocol (LDAP): String Representation of Search Filters

Defines a human-readable string representation of LDAP search filters that is appropriate for use in LDAP URLs and in other applications.

RFC 4516: Lightweight Directory Access Protocol (LDAP): Uniform Resource Locator

Describes a format for a Lightweight Directory Access Protocol Uniform Resource Locator.

RFC 4517: Lightweight Directory Access Protocol (LDAP): Syntaxes and Matching Rules

Defines a base set of syntaxes and matching rules for use in defining attributes for LDAP directories.

RFC 4518: Lightweight Directory Access Protocol (LDAP): Internationalized String Preparation

Defines string preparation algorithms for character-based matching rules defined for use in LDAP.

RFC 4519: Lightweight Directory Access Protocol (LDAP): Schema for User Applications

Provides a technical specification of attribute types and object classes intended for use by LDAP directory clients for many directory services, such as White Pages.

RFC 4523: Lightweight Directory Access Protocol (LDAP) Schema Definitions for X.509 Certificates

Describes schema for representing X.509 certificates, X.521 security information, and related elements in directories accessible using the Lightweight Directory Access Protocol (LDAP).

RFC 4524: COSINE LDAP/X.500 Schema

Provides a collection of schema elements for use with the Lightweight Directory Access Protocol from the COSINE and Internet X.500 pilot projects.

RFC 4525: Lightweight Directory Access Protocol (LDAP) Modify-Increment Extension

Describes an extension to the Lightweight Directory Access Protocol Modify operation to support an increment capability.

RFC 4526: Lightweight Directory Access Protocol (LDAP) Absolute True and False Filters

Extends the Lightweight Directory Access Protocol to support absolute True and False filters based upon similar capabilities found in X.500 directory systems.

RFC 4527: Lightweight Directory Access Protocol (LDAP) Read Entry Controls

Specifies an extension to the Lightweight Directory Access Protocol to allow the client to read the target entry of an update operation.

RFC 4528: Lightweight Directory Access Protocol (LDAP) Assertion Control

Defines the Lightweight Directory Access Protocol Assertion Control, which allows a client to specify that a directory operation should only be processed if an assertion applied to the target entry of the operation is true.

RFC 4529: Requesting Attributes by Object Class in the Lightweight Directory Access Protocol (LDAP)

Extends LDAP to support a mechanism that LDAP clients may use to request the return of all attributes of an object class.

RFC 4530: Lightweight Directory Access Protocol (LDAP) entryUUID Operational Attribute

Describes the LDAP/X.500 'entryUUID' operational attribute and associated matching rules and syntax.

RFC 4532: Lightweight Directory Access Protocol (LDAP) "Who am I?" Operation

Provides a mechanism for Lightweight Directory Access Protocol clients to obtain the authorization identity the server has associated with the user or application entity.

RFC 4616: The PLAIN Simple Authentication and Security Layer (SASL) Mechanism

Defines a simple cleartext user/password Simple Authentication and Security Layer mechanism called the PLAIN mechanism.

RFC 4634: US Secure Hash Algorithms (SHA and HMAC-SHA)

Specifies Secure Hash Algorithms, SHA-256, SHA-384, and SHA-512, for computing a condensed representation of a message or a data file.

RFC 4752: The Kerberos V5 ("GSSAPI") Simple Authentication and Security Layer (SASL) Mechanism

Describes the method for using the Generic Security Service Application Program Interface (GSS-API) Kerberos V5 in the Simple Authentication and Security Layer, called the GSSAPI mechanism.

RFC 4876: A Configuration Profile Schema for Lightweight Directory Access Protocol (LDAP)-Based Agents

Defines a schema for storing a profile for agents that make use of the Lightweight Directory Access protocol (LDAP).

RFC 5020: The Lightweight Directory Access Protocol (LDAP) entryDN Operational Attribute

Describes the Lightweight Directory Access Protocol (LDAP) / X.500 'entryDN' operational attribute, that provides a copy of the entry's distinguished name for use in attribute value assertions.

FIPS 180-1: Secure Hash Standard (SHA-1)

Specifies a Secure Hash Algorithm, SHA-1, for computing a condensed representation of a message or a data file.

FIPS 180-2: Secure Hash Standard (SHA-1, SHA-256, SHA-384, SHA-512)

Specifies four Secure Hash Algorithms for computing a condensed representation of electronic data.

DSMLv2: Directory Service Markup Language

Provides a method for expressing directory queries and updates as XML documents.

JavaScript Object Notation

A data-interchange format that aims to be both "easy for humans to read and write," and also "easy for machines to parse and generate."

Simple Cloud Identity Management: Core Schema 1.0

Platform neutral schema and extension model for representing users and groups in JSON and XML formats. OpenDJ supports the JSON formats.

Appendix F. LDAP Controls

Controls provide a mechanism whereby the semantics and arguments of existing LDAP operations may be extended. One or more controls may be attached to a single LDAP message. A control only affects the semantics of the message it is attached to. Controls sent by clients are termed *request controls*, and those sent by servers are termed *response controls*.

OpenDJ software supports the following LDAP controls:

Account Usability Control

Object Identifier: 1.3.6.1.4.1.42.2.27.9.5.8

Control originally provided by Sun Microsystems, used to determine whether a user account can be used to authenticate to the directory.

Assertion request control

Object Identifier: 1.3.6.1.1.12

RFC: RFC 4528 - Lightweight Directory Access Protocol (LDAP) Assertion Control

Authorization Identity request control

Object Identifier: 2.16.840.1.113730.3.4.16

RFC: RFC 3829 - Lightweight Directory Access Protocol (LDAP) Authorization Identity Request and Response Controls

Authorization Identity response control

Object Identifier: 2.16.840.1.113730.3.4.15

RFC: RFC 3829 - Lightweight Directory Access Protocol (LDAP) Authorization Identity Request and Response Controls

Entry Change Notification response control

Object Identifier: 2.16.840.1.113730.3.4.7

Internet-Draft: draft-ietf-ldapext-psearch - Persistent Search: A Simple LDAP Change Notification Mechanism

Get Effective Rights request control

Object Identifier: 1.3.6.1.4.1.42.2.27.9.5.2

Internet-Draft: draft-ietf-ldapext-acl-model - Access Control Model for LDAPv3

Manage DSAIT request control

Object Identifier: 2.16.840.1.113730.3.4.2

RFC: RFC 3296 - Named Subordinate References in Lightweight Directory Access Protocol (LDAP) Directories

Matched Values request control

Object Identifier: 1.2.826.0.1.3344810.2.3

RFC: RFC 3876 - Returning Matched Values with the Lightweight Directory Access Protocol version 3 (LDAPv3)

No-Op Control

Object Identifier: 1.3.6.1.4.1.4203.1.10.2

Internet-Draft: draft-zeilenga-ldap-noop - LDAP No-Op Control

Password Expired response control

Object Identifier: 2.16.840.1.113730.3.4.4

Internet-Draft: draft-vchu-ldap-pwd-policy - Password Policy for LDAP Directories

Password Expiring response control

Object Identifier: 2.16.840.1.113730.3.4.5

Internet-Draft: draft-vchu-ldap-pwd-policy - Password Policy for LDAP Directories

Password Policy response control

Object Identifier: 1.3.6.1.4.1.42.2.27.8.5.1

Internet-Draft: draft-behera-ldap-password-policy - Password Policy for LDAP Directories

Permissive Modify request control

Object Identifier: 1.2.840.113556.1.4.1413

Microsoft defined this control that, "Allows an LDAP modify to work under less restrictive conditions. Without it, a delete will fail if an attribute done not exist, and an add will fail if an attribute already exists. No data is needed in this control." (source of quote)

Persistent Search request control

Object Identifier: 2.16.840.1.113730.3.4.3

Internet-Draft: draft-ietf-ldapext-psearch - Persistent Search: A Simple LDAP Change Notification Mechanism

Post-Read request control

Object Identifier: 1.3.6.1.1.13.2

RFC: RFC 4527 - Lightweight Directory Access Protocol (LDAP) Read Entry Controls

Post-Read response control

Object Identifier: 1.3.6.1.1.13.2

RFC: RFC 4527 - Lightweight Directory Access Protocol (LDAP) Read Entry Controls

Pre-Read request control

Object Identifier: 1.3.6.1.1.13.1

RFC: RFC 4527 - Lightweight Directory Access Protocol (LDAP) Read Entry Controls

Pre-Read response control

Object Identifier: 1.3.6.1.1.13.1

RFC: RFC 4527 - Lightweight Directory Access Protocol (LDAP) Read Entry Controls

Proxied Authorization v1 request control

Object Identifier: 2.16.840.1.113730.3.4.12

Internet-Draft: draft-weltman-ldapv3-proxy-04 - LDAP Proxied Authorization Control

Proxied Authorization v2 request control

Object Identifier: 2.16.840.1.113730.3.4.18

RFC: RFC 4370 - Lightweight Directory Access Protocol (LDAP) Proxied Authorization Control

Public Changelog Exchange Control

Object Identifier: 1.3.6.1.4.1.26027.1.5.4

OpenDJ specific, for using the bookmark cookie when reading the external change log.

Server-Side Sort request control

Object Identifier: 1.2.840.113556.1.4.473

RFC: RFC 2891 - LDAP Control Extension for Server Side Sorting of Search Results

Server-Side Sort response control

Object Identifier: 1.2.840.113556.1.4.474

RFC: RFC 2891 - LDAP Control Extension for Server Side Sorting of Search Results

Simple Paged Results Control

Object Identifier: 1.2.840.113556.1.4.319

RFC: RFC 2696 - LDAP Control Extension for Simple Paged Results Manipulation

Subentries request controls

Object Identifier: 1.3.6.1.4.1.4203.1.10.1

RFC: Subentries in the Lightweight Directory Access Protocol (LDAP)

Object Identifier: 1.3.6.1.4.1.7628.5.101.1

Internet-Draft: draft-ietf-ldup-subentry - LDAP Subentry Schema

Subtree Delete request control

Object Identifier: 1.2.840.113556.1.4.805

Internet-Draft: draft-armijo-ldap-treedelete - Tree Delete Control

Virtual List View request control

Object Identifier: 2.16.840.1.113730.3.4.9

Internet-Draft: draft-ietf-ldapext-ldapv3-vlv - LDAP Extensions for Scrolling View Browsing of Search Results

Virtual List View response control

Object Identifier: 2.16.840.1.113730.3.4.10

Internet-Draft: draft-ietf-ldapext-ldapv3-vlv - LDAP Extensions for Scrolling View Browsing of Search Results

Appendix G. LDAP Extended Operations

Extended operations allow additional operations to be defined for services not already available in the protocol

OpenDJ software supports the following LDAP extended operations:

Cancel Extended Request

Object Identifier: 1.3.6.1.1.8

RFC: RFC 3909 - Lightweight Directory Access Protocol (LDAP) Cancel Operation

Get Connection ID Extended Request

Object Identifier: 1.3.6.1.4.1.26027.1.6.2

OpenDJ extended operation to return the connection ID of the associated client connection. This extended operation is intended for OpenDJ internal use.

Password Modify Extended Request

Object Identifier: 1.3.6.1.4.1.4203.1.11.1

RFC: RFC 3062 - LDAP Password Modify Extended Operation

Password Policy State Extended Operation

Object Identifier: 1.3.6.1.4.1.26027.1.6.1

OpenDJ extended operation to query and update password policy state for a given user entry. This extended operation is intended for OpenDJ internal use.

Start Transport Layer Security Extended Request

Object Identifier: 1.3.6.1.4.1.1466.20037

RFC: RFC 4511 - Lightweight Directory Access Protocol (LDAP): The Protocol

Who am I? Extended Request

Object Identifier: 1.3.6.1.4.1.4203.1.11.3

RFC: RFC 4532 - Lightweight Directory Access Protocol (LDAP) "Who am I?" Operation

Appendix H. Localization

OpenDJ software stores data in UTF-8 format. It enables you to store and to search for attribute values according to a variety of language specific locales. OpenDJ software is also itself localized for a smaller variety of languages.

H.1. OpenDJ Languages

OpenDJ 5 software is localized in the following languages:

- French
- German
- Japanese
- Simplified Chinese
- Spanish

Note

Certain messages have also been translated into Catalan, Korean, Polish, and Traditional Chinese. Some error messages including messages labeled ERROR are provided only in English.

H.2. Directory Support For Locales and Language Subtypes

OpenDJ software supports the following locales with their associated language and country codes and their collation order object identifiers. Locale support depends on the Java Virtual Machine used at run time. The following list reflects all supported locales.

Supported Locales

Afrikaans

Code tag: af

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.1.1

Albanian

Code tag: sq

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.127.1

Amharic

Code tag: am

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.2.1

Arabic

Code tag: ar

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.3.1

Arabic (Algeria)

Code tag: ar-DZ

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.6.1

Arabic (Bahrain)

Code tag: ar-BH

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.5.1

Arabic (Egypt)

Code tag: ar-EG

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.7.1

Arabic (India)

Code tag: ar-IN

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.8.1

Arabic (Iraq)

Code tag: ar-IQ

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.9.1

Arabic (Jordan)

Code tag: ar-JO

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.10.1

Arabic (Kuwait)

Code tag: ar-KW

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.11.1

Arabic (Lebanon)

Code tag: ar-LB

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.12.1

Arabic (Libya)

Code tag: ar-LY

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.13.1

Arabic (Morocco)

Code tag: ar-MA

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.14.1

Arabic (Oman)

Code tag: ar-OM

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.15.1

Arabic (Qatar)

Code tag: ar-QA

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.16.1

Arabic (Saudi Arabia)

Code tag: ar-SA

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.17.1

Arabic (Sudan)

Code tag: ar-SD

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.18.1

Arabic (Syria)

Code tag: ar-SY

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.19.1

Arabic (Tunisia)

Code tag: ar-TN

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.20.1

Arabic (United Arab Emirates)

Code tag: ar-AE

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.4.1

Arabic (Yemen)

Code tag: ar-YE

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.21.1

Armenian

Code tag: hy

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.89.1

Basque

Code tag: eu

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.70.1

Belarusian

Code tag: be

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.22.1

Bengali

Code tag: bn

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.24.1

Bulgarian

Code tag: bg

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.23.1

Catalan

Code tag: ca

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.25.1

Chinese

Code tag: zh

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.143.1

Chinese (China)

Code tag: zh-CN

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.144.1

Chinese (Hong Kong)

Code tag: zh-HK

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.145.1

Chinese (Macao)

Code tag: zh-MO

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.146.1

Chinese (Singapore)

Code tag: zh-SG

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.147.1

Chinese (Taiwan)

Code tag: zh-TW

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.148.1

Cornish

Code tag: kw

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.99.1

Croatian

Code tag: hr

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.87.1

Czech

Code tag: cs

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.26.1

Danish

Code tag: da

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.27.1

Dutch

Code tag: nl

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.105.1

Dutch (Belgium)

Code tag: nl-BE

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.106.1

Dutch (Netherlands)

Code tag: nl-NL

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.105.1

English

Code tag: en

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.34.1

English (Australia)

Code tag: en-AU

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.35.1

English (Canada)

Code tag: en-CA

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.36.1

English (Hong Kong)

Code tag: en-HK

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.38.1

English (India)

Code tag: en-IN

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.40.1

English (Ireland)

Code tag: en-IE

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.39.1

English (Malta)

Code tag: en-MT

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.41.1

English (New Zealand)

Code tag: en-NZ

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.42.1

English (Philippines)

Code tag: en-PH

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.43.1

English (Singapore)

Code tag: en-SG

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.44.1

English (South Africa)

Code tag: en-ZA

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.46.1

English (U.S. Virgin Islands)

Code tag: en-VI

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.45.1

English (United Kingdom)

Code tag: en-GB

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.37.1

English (United States)

Code tag: en-US

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.34.1

English (Zimbabwe)

Code tag: en-ZW

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.47.1

Esperanto

Code tag: eo

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.48.1

Estonian

Code tag: et

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.69.1

Faroese

Code tag: fo

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.75.1

Finnish

Code tag: fi

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.74.1

French

Code tag: fr

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.76.1

French (Belgium)

Code tag: fr-BE

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.77.1

French (Canada)

Code tag: fr-CA

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.78.1

French (France)

Code tag: fr-FR

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.76.1

French (Luxembourg)

Code tag: fr-LU

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.80.1

French (Switzerland)

Code tag: fr-CH

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.79.1

Gallegan

Code tag: gl

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.82.1

German

Code tag: de

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.28.1

German (Austria)

Code tag: de-AT

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.29.1

German (Belgium)

Code tag: de-BE

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.30.1

German (Germany)

Code tag: de-DE

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.28.1

German (Luxembourg)

Code tag: de-LU

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.32.1

German (Switzerland)

Code tag: de-CH

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.31.1

Greek

Code tag: el

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.33.1

Greenlandic

Code tag: kl

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.95.1

Gujarati

Code tag: gu

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.83.1

Hebrew

Code tag: iw

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.85.1

Hindi

Code tag: hi

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.86.1

Hungarian

Code tag: hu

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.88.1

Icelandic

Code tag: is

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.91.1

Indonesian

Code tag: in

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.90.1

Irish

Code tag: ga

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.81.1

Italian

Code tag: it

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.92.1

Italian (Switzerland)

Code tag: it-CH

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.93.1

Japanese

Code tag: ja

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.94.1

Kannada

Code tag: kn

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.96.1

Konkani

Code tag: kok

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.98.1

Korean

Code tag: ko

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.97.1

Latvian

Code tag: lv

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.101.1

Lithuanian

Code tag: lt

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.100.1

Macedonian

Code tag: mk

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.102.1

Maltese

Code tag: mt

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.104.1

Manx

Code tag: gv

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.84.1

Marathi

Code tag: mr

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.103.1

Norwegian

Code tag: no

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.107.1

Norwegian (Norway)

Code tag: no-NO-B

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.110.1

Norwegian Bokm?l

Code tag: nb

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.110.1

Norwegian Nynorsk

Code tag: nn

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.109.1

Oromo

Code tag: om

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.111.1

Oromo (Ethiopia)

Code tag: om-ET

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.112.1

Oromo (Kenya)

Code tag: om-KE

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.113.1

Persian

Code tag: fa

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.71.1

Persian (India)

Code tag: fa-IN

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.72.1

Persian (Iran)

Code tag: fa-IR

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.73.1

Polish

Code tag: pl

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.114.1

Portuguese

Code tag: pt

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.115.1

Portuguese (Brazil)

Code tag: pt-BR

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.116.1

Portuguese (Portugal)

Code tag: pt-PT

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.115.1

Romanian

Code tag: ro

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.117.1

Russian

Code tag: ru

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.118.1

Russian (Russia)

Code tag: ru-RU

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.118.1

Russian (Ukraine)

Code tag: ru-UA

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.119.1

Serbian

Code tag: sr

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.128.1

Serbo-Croatian

Code tag: sh

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.120.1

Slovak

Code tag: sk

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.121.1

Slovenian

Code tag: sl

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.122.1

Somali

Code tag: so

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.123.1

Somali (Djibouti)

Code tag: so-DJ

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.124.1

Somali (Ethiopia)

Code tag: so-ET

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.125.1

Somali (Kenya)

Code tag: so-KE

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.126.1

Somali (Somalia)

Code tag: so-SO

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.123.1

Spanish

Code tag: es

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.49.1

Spanish (Argentina)

Code tag: es-AR

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.50.1

Spanish (Bolivia)

Code tag: es-BO

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.51.1

Spanish (Chile)

Code tag: es-CL

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.52.1

Spanish (Colombia)

Code tag: es-CO

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.53.1

Spanish (Costa Rica)

Code tag: es-CR

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.54.1

Spanish (Dominican Republic)

Code tag: es-DO

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.55.1

Spanish (Ecuador)

Code tag: es-EC

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.56.1

Spanish (El Salvador)

Code tag: es-SV

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.65.1

Spanish (Guatemala)

Code tag: es-GT

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.57.1

Spanish (Honduras)

Code tag: es-HN

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.58.1

Spanish (Mexico)

Code tag: es-MX

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.59.1

Spanish (Nicaragua)

Code tag: es-NI

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.60.1

Spanish (Panama)

Code tag: es-PA

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.61.1

Spanish (Paraguay)

Code tag: es-PY

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.64.1

Spanish (Peru)

Code tag: es-PE

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.62.1

Spanish (Puerto Rico)

Code tag: es-PR

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.63.1

Spanish (Spain)

Code tag: es-ES

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.49.1

Spanish (United States)

Code tag: es-US

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.66.1

Spanish (Uruguay)

Code tag: es-UY

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.67.1

Spanish (Venezuela)

Code tag: es-VE

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.68.1

Swahili

Code tag: sw

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.131.1

Swahili (Kenya)

Code tag: sw-KE

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.132.1

Swahili (Tanzania)

Code tag: sw-TZ

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.133.1

Swedish

Code tag: sv

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.129.1

Swedish (Finland)

Code tag: sv-FI

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.130.1

Swedish (Sweden)

Code tag: sv-SE

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.129.1

Tamil

Code tag: ta

Collation order object identifier: 1 3 1.3.6.1.4.1.42.2.27.9.4.134.1

Telugu

Code tag: te

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.135.1

Thai

Code tag: th

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.136.1

Tigrinya

Code tag: ti

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.137.1

Tigrinya (Eritrea)

Code tag: ti-ER

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.138.1

Tigrinya (Ethiopia)

Code tag: ti-ET

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.139.1

Turkish

Code tag: tr

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.140.1

Ukrainian

Code tag: uk

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.141.1

Vietnamese

Code tag: vi

Collation order object identifier: 1.3.6.1.4.1.42.2.27.9.4.142.1

Supported Language Subtypes

- Afrikaans, af
- Albanian, sq
- Amharic, am

- Arabic, ar
- Armenian, hy
- Basque, eu
- Belarusian, be
- Bengali, bn
- Bulgarian, bg
- Catalan, ca
- Chinese, zh
- Cornish, kw
- Croatian, hr
- Czech, cs
- Danish, da
- Dutch, nl
- English, en
- Esperanto, eo
- Estonian, et
- Faroese, fo
- Finnish, fi
- French, fr
- Gallegan, gl
- German, de
- Greek, el
- Greenlandic, kl
- Gujarati, gu
- Hebrew, iw
- Hindi, hi

- Hungarian, hu
- Icelandic, is
- Indonesian, in
- Irish, ga
- Italian, it
- Japanese, ja
- Kannada, kn
- Konkani, kok
- Korean, ko
- Latvian, lv
- Lithuanian, lt
- Macedonian, mk
- Maltese, mt
- Manx, gv
- Marathi, mr
- Norwegian, no
- Norwegian Bokm?l, nb
- Norwegian Nynorsk, nn
- Oromo, om
- Persian, fa
- Polish, pl
- Portuguese, pt
- Romanian, ro
- Russian, ru
- Serbian, sr
- Serbo-Croatian, sh

- Slovak, sk
- Slovenian, sl
- Somali, so
- Spanish, es
- Swahili, sw
- Swedish, sv
- Tamil, ta
- Telugu, te
- Thai, th
- Tigrinya, ti
- Turkish, tr
- Ukrainian, uk
- Vietnamese, vi

Appendix I. Release Levels and Interface Stability

This appendix includes ForgeRock definitions for product release levels and interface stability.

Some interfaces are labelled as Evolving in the body of the documentation. In addition, the following rules apply:

- All Java APIs are Evolving, except `com.*` packages, which are Internal/Undocumented.
- The class `org.forgerock.opendj.ldap.CoreMessages` is Internal.
- The configuration, user, and application programming interfaces for RESTful access over HTTP to directory data are Evolving. This includes interfaces exposed for the HTTP connection handler, its access log, and also the REST to LDAP gateway.
- Text in log messages should be considered Internal. Log message IDs are Evolving.
- The default content of `cn=schema` (LDAP schema) is Evolving.
- The monitoring interface `cn=monitor` for LDAP and the monitoring interface exposed by the JMX connection handler are Evolving.
- Newly Deprecated and Removed interfaces are identified in Chapter 3, "*Compatibility*" in the *Release Notes*.
- Interfaces that are not described in released product documentation should be considered Internal/Undocumented. For example, the LDIF representation of the server configuration, `config.ldif`, should be considered Internal.

I.1. ForgeRock Product Release Levels

ForgeRock defines Major, Minor, and Maintenance product release levels. The release level is reflected in the version number. The release level tells you what sort of compatibility changes to expect.

Table I.1. Release Level Definitions

Release Label	Version Numbers	Characteristics
Major	Version: x[.0.0] (trailing 0s are optional)	• Bring major new features, minor features, and bug fixes • Can include changes even to Stable interfaces • Can remove previously Deprecated functionality, and in rare cases remove Evolving functionality that has not been explicitly Deprecated • Include changes present in previous Minor and Maintenance releases
Minor	Version: x.y[.0] (trailing 0s are optional)	• Bring minor features, and bug fixes • Can include backwards-compatible changes to Stable interfaces in the same Major release, and incompatible changes to Evolving interfaces • Can remove previously Deprecated functionality • Include changes present in previous Minor and Maintenance releases
Maintenance	Version: x.y.z	• Bring bug fixes • Are intended to be fully compatible with previous versions from the same Minor release

I.2. ForgeRock Product Interface Stability

ForgeRock products support many protocols, APIs, GUIs, and command-line interfaces. Some of these interfaces are standard and very stable. Others offer new functionality that is continuing to evolve.

ForgeRock acknowledges that you invest in these interfaces, and therefore must know when and how ForgeRock expects them to change. For that reason, ForgeRock defines interface stability labels and uses these definitions in ForgeRock products.

Table I.2. Interface Stability Definitions

Stability Label	Definition
Stable	This documented interface is expected to undergo backwards-compatible changes only for major releases. Changes may be announced at least one minor release before they take effect.
Evolving	This documented interface is continuing to evolve and so is expected to change, potentially in backwards-incompatible ways even in a minor release. Changes are documented at the time of product release. While new protocols and APIs are still in the process of standardization, they are Evolving. This applies for example to recent Internet-Draft implementations, and also to newly developed functionality.
Deprecated	This interface is deprecated and likely to be removed in a future release. For previously stable interfaces, the change was likely announced in a previous release. Deprecated interfaces will be removed from ForgeRock products.
Removed	This interface was deprecated in a previous release and has now been removed from the product.
Internal/Undocumented	Internal and undocumented interfaces can change without notice. If you depend on one of these interfaces, contact ForgeRock support or email info@forgerock.com to discuss your needs.

Index