Tools Reference

/ Directory Services 7

Latest update: 7.0.2

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 command-line tools.

Table of Contents

Overview

This reference covers Directory Services tools, which are bundled with the software. For the dsconfig command, also see the Configuration Reference.

ForgeRock Identity Platform™ serves as the basis for our simple and comprehensive Identity and Access Management solution. We help our customers deepen their relationships with their customers, and improve the productivity and connectivity of their employees and partners. For more information about ForgeRock and about the platform, see https://www.forgerock.com.

Chapter 1

addrate — measure add and delete throughput and response time

Synopsis

addrate {options} 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

`-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. To always generate the same data with the same command, use the same non-zero seed value. A value of zero (the default) results in different data each time the tool is run.

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 | --numConcurrentRequests {numConcurrentRequests}

Number of concurrent requests per connection.

Default: 1

-Y | --proxyAs {authzID}

Use the proxied authorization control with the given authorization ID.

LDAP 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:

-E | --reportAuthzId

Use the authorization identity control.

Default: false

-h | --hostname {host}

Fully-qualified server host name or IP address.

Default: localhost.localdomain

-N | --certNickname {nickname}

Nickname of the certificate that should be sent to the server for SSL client authentication.

-o | --saslOption {name=value}

SASL bind options.

-p | --port {port}

Directory server port number.

-q | --useStartTls

Use StartTLS to secure communication with the server.

Default: false

-T | --trustStorePassword {trustStorePassword}

Truststore cleartext password.

--useJavaKeyStore {keyStorePath}

JKS keystore containing the certificate which should be used for SSL client authentication.

--useJavaTrustStore {trustStorePath}

Use a JKS truststore file for validating server certificate.

--useJceKeyStore {keyStorePath}

JCEKS keystore containing the certificate which should be used for SSL client authentication.

--useJceTrustStore {trustStorePath}

Use a JCEKS truststore file for validating server certificate.

--useJvmTrustStore

Use the a JVM truststore for validating server certificate.

Default: false

--usePasswordPolicyControl

Use the password policy request control.

Default: false

--usePkcs11KeyStore

PKCS#11 keystore containing the certificate which should be used for SSL client authentication.

Default: false

--usePkcs12KeyStore {keyStorePath}

PKCS#12 keystore containing the certificate which should be used for SSL client authentication.

--usePkcs12TrustStore {trustStorePath}

Use a PKCS#12 truststore file for validating server certificate.

-w | --bindPassword {bindPassword}

Password to use to bind to the server. Omit this option while providing the bind DN to ensure that the command prompts for the password, rather than entering the password as a command argument.

-W | --keyStorePassword {keyStorePassword}

Keystore cleartext password.

-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:

`--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.

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

```
$ addrate \
 --hostname localhost \
 --port 1389 \
 --bindDn uid=admin \
 --bindPassword password \
 --numConnections 10 \
 --keepConnectionsOpen \
 --deleteMode random \
 --deleteSizeThreshold 10,000 \
 /path/to/opendj/config/MakeLDIF/addrate.template
--------------------------------------------------------------------------------------
|     Throughput    |                 Response Time                 |    Additional   |
|    (ops/second)   |                (milliseconds)                 |    Statistics   |
|   recent  average |   recent  average    99.9%   99.99%  99.999% |  err/sec   Add% |
--------------------------------------------------------------------------------------
|    275.3    275.3 |   12.057   12.057   102.76   113.25   113.25 |       0.0 100.00 |
|    329.1    302.2 |   10.181   11.036    93.85   113.25   113.25 |       0.0 100.00 |
|    339.2    314.5 |    9.719   10.563   113.25   122.16   122.16 |       0.0 100.00 |
|    365.6    327.3 |    7.616    9.740   105.91   120.59   122.16 |       0.0 100.00 |
|    385.8    339.0 |    7.312    9.187   102.76   120.59   122.16 |       0.0 100.00 |
|    366.8    343.6 |    7.776    8.936    99.61   120.59   122.16 |       0.0  91.44 |
|    337.2    342.7 |   10.746    9.191   119.01   505.41   509.61 |       0.0  49.61 |
^C|    372.8    343.8 |    7.662    9.130   119.01   505.41   509.61 |       0.0  50.30 |
Purge phase...
```

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

```
$ addrate \
 --hostname localhost \
 --port 1389 \
 --bindDn uid=admin \
 --bindPassword password \
 --numConnections 10 \
 --keepConnectionsOpen \
 --deleteMode fifo \
 --deleteAgeThreshold 10 \
 /path/to/opendj/config/MakeLDIF/addrate.template
--------------------------------------------------------------------------------------
|     Throughput    |                 Response Time                 |    Additional   |
|    (ops/second)   |                (milliseconds)                 |    Statistics   |
|   recent  average |   recent  average    99.9%   99.99%  99.999% |  err/sec   Add% |
--------------------------------------------------------------------------------------
|    377.8    377.8 |    7.258    7.258    18.87    20.71    20.71 |       0.0 100.00 |
|    393.2    385.5 |    7.069    7.161    18.09    23.20    23.20 |       0.0 100.00 |
|    387.8    386.3 |    7.226    7.183    28.18    36.44    37.49 |       0.0  50.05 |
|    396.8    388.9 |    6.957    7.125    23.20    36.44    37.49 |       0.0  50.18 |
|    400.6    391.2 |    6.906    7.080    19.27    36.44    37.49 |       0.0  49.73 |
^C|    397.6    391.4 |    7.083    7.080    19.27    36.44    37.49 |       0.0  50.00 |
Purge phase...
```

These examples use the following options:

`--hostname localhost`
`--port 1389`

Access the server running on the local system on port 1389.

`--bindDn uid=admin`
`--bindPassword password`

Authenticate as the directory root user `uid=admin` with the bind password that is literally `password`.

This user is not subject to access control, so rates may be higher than what you observe with a regular user.

`--numConnections 10`

Open 10 connections to the server.

`--keepConnectionsOpen`

Keep the connections open to reuse them during the operation.

`--deleteMode (random | fifo)`

After adding entries, delete them in random order, or in first-in-first-out order.

`--deleteSizeThreshold 10,000`

Add 10,000 entries before starting to delete them.

`--deleteAgeThreshold 10`

Begin to delete entries when they are 10 seconds old.

`/path/to/opendj/config/MakeLDIF/addrate.template`

When building entries to add, use this file as the template.

Notice the following characteristics of the output:

- The first two columns show the throughput in operations completed per second.

  The recent column shows the average rate for operations reflected in this row of output.

  The average column shows the average rate since the beginning of the run.

- The response time columns indicate characteristics of response latency in milliseconds.

  The recent column shows the average latency for operations reflected in this row of output.

  The average column shows the average latency since the beginning of the run.

The "99.9%" column shows the latency after which 99.9% of operations have completed. Only 1 operation in 1000 took longer than this.

The "99.99%" column shows the latency after which 99.99% of operations have completed. Only 1 operation in 10,000 took longer than this.

The "99.999%" column shows the latency after which 99.999% of operations have completed. Only 1 operation in 100,000 took longer than this.

- The additional statistics columns show information about what is happening during the run.

  The "err/sec" column shows the rate of error results per second for this row of output. Unless you have intentionally set up the command to generate errors, this column should indicate `0.0`. Check that this column matches your expectations before looking at any other columns.

  The "Add%" column shows the percentage of operations performed that were adds. The rest are delete operations. Notice that the percentage of add operations drops as the command begins to delete entries.

Chapter 2

authrate — measure bind throughput and response time

Synopsis

authrate {options} [filter template 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.

Template 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 template strings.

Example (bind only):

```
authrate -p 1389 -D 'uid=user.{},ou=people,dc=example,dc=com' \

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

Example (search then bind):

```
authrate -p 1389 -D '{2}' -w password -f -c 10 \

-b 'ou=people,dc=example,dc=com' -s one -g 'rand(0,2000)' '(uid=user.{1})'
```

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 template string.

`-B | --warmUpDuration {warmUpDuration}`

Warm up duration in seconds.

Default: 0

`-c | --numConnections {numConnections}`

Number of connections.

Default: 1

-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 template strings in program parameters (ie. Base DN, Search Filter). The set of all arguments provided form 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}

Calculate max response time for a percentile of operations.

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:

`--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:

`-E | --reportAuthzId`

Use the authorization identity control.

Default: false

`-h | --hostname {host}`

Fully-qualified server host name or IP address.

Default: localhost.localdomain

`-N | --certNickname {nickname}`

Nickname of the certificate that should be sent to the server for SSL client authentication.

`-o | --saslOption {name=value}`

SASL bind options.

-p | --port {port}

Directory server port number.

-q | --useStartTls

Use StartTLS to secure communication with the server.

Default: false

-T | --trustStorePassword {trustStorePassword}

Truststore cleartext password.

--useJavaKeyStore {keyStorePath}

JKS keystore containing the certificate which should be used for SSL client authentication.

--useJavaTrustStore {trustStorePath}

Use a JKS truststore file for validating server certificate.

--useJceKeyStore {keyStorePath}

JCEKS keystore containing the certificate which should be used for SSL client authentication.

--useJceTrustStore {trustStorePath}

Use a JCEKS truststore file for validating server certificate.

--useJvmTrustStore

Use the a JVM truststore for validating server certificate.

Default: false

--usePasswordPolicyControl

Use the password policy request control.

Default: false

--usePkcs11KeyStore

PKCS#11 keystore containing the certificate which should be used for SSL client authentication.

Default: false

--usePkcs12KeyStore {keyStorePath}

PKCS#12 keystore containing the certificate which should be used for SSL client authentication.

`--usePkcs12TrustStore {trustStorePath}`

Use a PKCS#12 truststore file for validating server certificate.

`-w | --bindPassword {bindPassword}`

Password to use to bind to the server. Omit this option while providing the bind DN to ensure that the command prompts for the password, rather than entering the password as a command argument.

`-W | --keyStorePassword {keyStorePassword}`

Keystore cleartext password.

`-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:

`-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.

`-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 \
 --hostname localhost \
 --port 1389 \
 --argument "rand(0,2000)" --bindDn "uid=user.{},ou=people,dc=example,dc=com" \
 --bindPassword password \
 --numConnections 10 \
 --keepConnectionsOpen
-------------------------------------------------------------------------------
|     Throughput    |                 Response Time                 |          |
|    (ops/second)   |                (milliseconds)                 |          |
|   recent  average |   recent  average    99.9%   99.99%  99.999% |  err/sec |
-------------------------------------------------------------------------------
|  26046.6  26046.6 |    0.377    0.377    10.62    20.71    36.44 |      0.0 |
|  45757.6  35902.1 |    0.214    0.273     7.21    15.93    26.08 |      0.0 |
|  47457.8  39754.0 |    0.206    0.247     5.70    13.57    25.30 |      0.0 |
|  47715.2  41744.3 |    0.205    0.235     4.98    12.32    24.77 |      0.0 |
|  48203.0  43036.0 |    0.203    0.228     4.59    11.80    20.84 |      0.0 |
|  49363.0  44090.5 |    0.198    0.222     4.33    11.27    20.71 |      0.0 |
^C
```

This example uses the following options:

`--hostname localhost`
`--port 1389`

Access the server running on the local system on port 1389.

`--argument "rand(0,2000)" --bindDn "uid=user.{},ou=people,dc=example,dc=com"`

Authenticate as a user with bind DN `uid=user.`*`number`*`,ou=people,dc=example,dc=com`, where *number* is a random number between 0 and 2000, inclusive.

`--bindPassword password`

Authenticate with the bind password that is literally `password`.

`--numConnections 10`

Open 10 connections to the server.

`--keepConnectionsOpen`

Keep the connections open to reuse them during the operation.

Notice the following characteristics of the output:

- The first two columns show the throughput in operations completed per second.

  The recent column shows the average rate for operations reflected in this row of output.

  The average column shows the average rate since the beginning of the run.

- The response time columns indicate characteristics of response latency in milliseconds.

  The recent column shows the average latency for operations reflected in this row of output.

  The average column shows the average latency since the beginning of the run.

  The "99.9%" column shows the latency after which 99.9% of operations have completed. Only 1 operation in 1000 took longer than this.

  The "99.99%" column shows the latency after which 99.99% of operations have completed. Only 1 operation in 10,000 took longer than this.

  The "99.999%" column shows the latency after which 99.999% of operations have completed. Only 1 operation in 100,000 took longer than this.

- The "err/sec" column show the rate of error results per second for this row of output.

  Unless you have intentionally set up the command to generate errors, this column should indicate `0.0`. Check that this column matches your expectations before looking at any other columns.

Chapter 3

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:

`-b | --baseDn {baseDN}`

The base DN within the backend.

`-i | --indexName {indexName}`

The name of the index.

`-k | --minKeyValue {minKeyValue}`

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

`-K | --maxKeyValue {maxKeyValue}`

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

`-n | --backendId {backendName}`

The backend ID of the backend.

`-p | --skipDecode`

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

Default: false

`-q | --statsOnly`

Do not display backend data, just statistics.

Default: false

`-s | --minDataSize {minDataSize}`

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

Default: -1

`-S | --maxDataSize {maxDataSize}`

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

Default: -1

`-x | --minHexKeyValue {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.

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:

`-d | --dbName {databaseName}`

The raw database name.

`-k | --minKeyValue {minKeyValue}`

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

`-K | --maxKeyValue {maxKeyValue}`

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

`-l | --singleLine`

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

Default: false

`-n | --backendId {backendName}`

The backend ID of the backend.

`-q | --statsOnly`

Do not display backend data, just statistics.

Default: false

-s | --minDataSize {minDataSize}

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

Default: -1

-S | --maxDataSize {maxDataSize}

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

Default: -1

-x | --minHexKeyValue {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.

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:

`-b | --baseDn {baseDN}`

The base DN within the backend.

`-n | --backendId {backendName}`

The backend ID of 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 show-index-status subcommand, 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 the directory server's internal use.

Example: `givenName.caseIgnoreSubstringsMatch:6`

Raw DB Name

The internal name of the database within the storage which the directory server is using for the index.

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 the 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 the 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:

`-b | --baseDn {baseDN}`

The base DN within the backend.

`-n | --backendId {backendName}`

The backend ID of the backend.

Exit Codes

0

The command completed successfully.

> 0

An error occurred.

Examples

The following example displays index information:

```
$ backendstat \
 dump-index \
 --backendId userData \
 --baseDn dc=example,dc=com \
 --indexName 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
```

Chapter 4

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 --encodedData YWRkOiBkZXNjcmlwdGlvbgpkZXNjcmlwdGlvbjogQSB0aGlyZCBjaGFuZ2UK\
LQpyZXBsYWNlOiBtb2RpZmllcnNOYW1lCm1vZGlmaWVyc05hbWU6IGNuPURpcmVjdG9yeSBNYW5hZ2V\
yLGNuPVJvb3QgRE5zLGNuPWNvbmZpZwotCnJlcGxhY2U6IG1vZGlmeVRpbWVzdGFtcAptb2RpZnlUaW\
1lc3RhbXA6IDIwMTEwNjEzMDcxMjEwWgotCg==
add: description
description: A third change
-
replace: modifiersName
modifiersName: uid=admin,cn=Root DNs,cn=config
-
replace: modifyTimestamp
modifyTimestamp: 20110613071210Z
-
```

Chapter 5
changelogstat — debug changelog and changenumber files

Synopsis

changelogstat {subcommand} {options}

Description

This utility can be used to debug changelog and changenumber files.

Options

The changelogstat command takes the following options:

`-V | --version`

Display Directory Server version information.

Default: false

`-H | --help`

Display this usage information.

Default: false

Subcommands

The changelogstat command supports the following subcommands:

changelogstat dump-change-number-db

Dump the change number DB.

Options

The changelogstat dump-change-number-db command takes the following options:

`--from {change number}`

The lower bound of the range of change numbers to dump.

`--outputDir {directory}`

The output directory for the dump files.

`--to {change number}`

The upper bound of the range of change numbers to dump.

changelogstat dump-replica-db

Dump the replica DB for a given domain and replica.

Options

The changelogstat dump-replica-db command takes the following options:

`--from {csn}`

The lower bound of the range of changes to dump.

`--outputDir {directory}`

The output directory for the dump files.

`--to {csn}`

The upper bound of the range of changes to dump.

changelogstat dump-replica-db-file

Dump a replica DB file.

Options

The changelogstat dump-replica-db-file command takes the following options:

`--baseDn {base dn}`

The base-dn of the changes contained in the provided replica DB file.

Default:

`--from {csn}`

The lower bound of the range of changes to dump.

`--to {csn}`

The upper bound of the range of changes to dump.

Exit Codes

0

The command completed successfully.

1

An error occurred.

Examples

To dump the change number DB from change number 10 to 15:

```
$ changelogstat dump-change-number-db --from 10 --to 15

 changeNumber=10 baseDN=dc=example,dc=com csn=01010166aaf2a3e3000002c61 (sid=1,tsd=Thu Oct 25 13:18:40
 CEST 2018,ts=1540466320355,seqnum=710)
 changeNumber=11 baseDN=dc=example,dc=com csn=01010166aaf2a3e3000002c71 (sid=1,tsd=Thu Oct 25 13:18:40
 CEST 2018,ts=1540466320355,seqnum=711)
 changeNumber=12 baseDN=dc=example,dc=com csn=01010166aaf2a3e3000002c81 (sid=1,tsd=Thu Oct 25 13:18:40
 CEST 2018,ts=1540466320355,seqnum=712)
 changeNumber=13 baseDN=dc=example,dc=com csn=01010166aaf2a3e3000002c91 (sid=1,tsd=Thu Oct 25 13:18:40
 CEST 2018,ts=1540466320355,seqnum=713)
 changeNumber=14 baseDN=dc=example,dc=com csn=01010166aaf2a3e3000002ca1 (sid=1,tsd=Thu Oct 25 13:18:40
 CEST 2018,ts=1540466320355,seqnum=714)
 changeNumber=15 baseDN=dc=example,dc=com csn=01010166aaf2a3e3000002cb1 (sid=1,tsd=Thu Oct 25 13:18:40
 CEST 2018,ts=1540466320355,seqnum=715)
```

To dump the replica DB for the domain dc=example,dc=com on the server 1:

```
$ changelogstat dump-replica-db --outputDir myOutputDir dc=example,dc=com 1
```

To dump a specific replica DB file:

```
$ changelogstat dump-replica-db-file changelogDb/2.dom/1.server/01010166aaf2a3e3000002bd1.log

 ModifyMsg content:  protocolVersion: 10 dn: uid=user.48,ou=people,dc=example,dc=com csn:
 01010166aaf2a3e3000002bd1 uniqueId: 55cf0798-774c-3d55-888b-c3833d57ba0e
 ModifyMsg content:  protocolVersion: 10 dn: uid=user.73,ou=people,dc=example,dc=com csn:
 01010166aaf2a3e3000002be1 uniqueId: 8977f8ac-1579-3538-accf-a6ce7f612076
 ModifyMsg content:  protocolVersion: 10 dn: uid=user.34,ou=people,dc=example,dc=com csn:
 01010166aaf2a3e3000002bf1 uniqueId: a1fa5d92-326a-3283-a040-114300fcc7e5
 ModifyMsg content:  protocolVersion: 10 dn: uid=user.69,ou=people,dc=example,dc=com csn:
 01010166aaf2a3e3000002c01 uniqueId: da34114f-b183-3ccd-b7d8-486791aa4651
 ...
 ModifyMsg content:  protocolVersion: 10 dn: uid=user.4,ou=people,dc=example,dc=com csn:
 01010166aaf2da2400008b7d1 uniqueId: 1539438e-ae81-36ce-aecf-dd4dc72a12f0
 ModifyMsg content:  protocolVersion: 10 dn: uid=user.27,ou=people,dc=example,dc=com csn:
 01010166aaf2da2400008b7e1 uniqueId: 950dd85c-9e53-3b12-8074-c2eb88582156
 ModifyMsg content:  protocolVersion: 10 dn: uid=user.13,ou=people,dc=example,dc=com csn:
 01010166aaf2da2400008b7f1 uniqueId: 120b8640-5295-36dc-9ea3-8b20735348ab
 ModifyMsg content:  protocolVersion: 10 dn: uid=user.91,ou=people,dc=example,dc=com csn:
 01010166aaf2da2400008b801 uniqueId: 6e9c2930-cc4f-3f7b-9dcf-d81aa46f57f9
```

Chapter 6

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

Synopsis

create-rc-script {options}

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.

`-s | --systemdService {path}`

The path to the systemd service file to create.

`-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 the server at boot time on a Debian-based system, and then updates the runlevel system to use the script:

```
$ sudo create-rc-script --outputFile /etc/init.d/opendj --userName opendj-user
$ sudo update-rc.d opendj
```

Chapter 7
dsbackup — Backup and restore backends

Synopsis

dsbackup {subcommand} {options}

Description

Backup and restore backends, manage backup files.

Options

The dsbackup command takes the following options:

Command options:

`--offline`

> Indicates that the command will operate independently of the server process. It will run regardless of whether the server is started or stopped. When using this option with the restore sub-command, the server must be stopped; also as the command will write to server files, you should run the command as a user having the same filesystem permissions as the user running the server. Using this option with the create sub-command when the server is running is possible and supported. With JE Backends, the integrity of the backup is ensured by the process. With LDIF backends, avoid simultaneous changes to the backends.
>
> 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.

`--description {description}`

Gives a description to the task.

`--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.

The schedule pattern for a recurring task supports only the following crontab features:

Time and Date Fields

Field	Allowed Values
minute	0-59
hour	0-23
day of month	1-31
month	1-12 (or names)
day of week	0-7 (0 or 7 is Sunday, or use names)

A field can contain an asterisk, `*`. An asterisk stands for *`first-last`*.

Fields can include ranges of numbers. A range is two numbers separated by a hyphen, and is inclusive. For example, `8-10` for an "hour" field means execution at hours 8, 9, and 10.

Fields can include lists. A list is a set of numbers or ranges separated by commas. For example, `4, 8-10` for an "hour" field means execution at hours 4, 8, 9, and 10.

When using names for in "month" or "day of week" fields, use the first three letters of the particular month or day of the week. Case does not matter. Ranges and lists of names are not supported.

`-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.

`--taskId {taskID}`

Gives an ID to the task.

LDAP 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: uid=admin

`-E | --reportAuthzId`

Use the authorization identity control.

Default: false

`-h | --hostname {host}`

Fully-qualified server host name or IP address.

Default: localhost.localdomain

`-N | --certNickname {nickname}`

Nickname of the certificate that should be sent to the server for SSL client authentication.

`-o | --saslOption {name=value}`

SASL bind options.

`-p | --port {port}`

Directory server administration port number.

`-T | --trustStorePassword {trustStorePassword}`

Truststore cleartext password.

`--useJavaKeyStore {keyStorePath}`

JKS keystore containing the certificate which should be used for SSL client authentication.

--useJavaTrustStore {trustStorePath}

Use a JKS truststore file for validating server certificate.

--useJceKeyStore {keyStorePath}

JCEKS keystore containing the certificate which should be used for SSL client authentication.

--useJceTrustStore {trustStorePath}

Use a JCEKS truststore file for validating server certificate.

--useJvmTrustStore

Use the a JVM truststore for validating server certificate.

Default: false

--usePasswordPolicyControl

Use the password policy request control.

Default: false

--usePkcs11KeyStore

PKCS#11 keystore containing the certificate which should be used for SSL client authentication.

Default: false

--usePkcs12KeyStore {keyStorePath}

PKCS#12 keystore containing the certificate which should be used for SSL client authentication.

--usePkcs12TrustStore {trustStorePath}

Use a PKCS#12 truststore file for validating server certificate.

-w | --bindPassword {bindPassword}

Password to use to bind to the server. Omit this option while providing the bind DN to ensure that the command prompts for the password, rather than entering the password as a command argument.

-W | --keyStorePassword {keyStorePassword}

Keystore cleartext password.

-X | --trustAll

Trust all server SSL certificates.

Default: false

Utility input/output options:

`--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

Subcommands

The dsbackup command supports the following subcommands:

dsbackup create

Take encrypted and signed backups of individual backends and send them to the desired location.

Options

The dsbackup create command takes the following options:

`-d | --backupLocation {backup location}`

Backup file-system path or URI for alternative storage mechanisms. File-system paths may be expressed as absolute or relative paths and are resolved relative to the current working directory when the tool is run in offline mode, or relative to the server instance directory when the tool is run in task mode. Read the documentation for further information regarding alternative backup storage mechanisms.

`-n | --backendName {backendName}`

The name of the backend to back up. Specify this option multiple times to backup multiple backends or skip this option to backup all the enabled backends that support backups.

`--storageProperty {PROP:VALUE}`

Assigns a value to a storage property where PROP is the name of the property and VALUE is the single value to be assigned. Specify the same property multiple times in order to assign more than one value to it.

dsbackup list

List the backups at the specified location.

Options

The dsbackup list command takes the following options:

`-d | --backupLocation {backup location}`

Location containing backups: file-system path or URI for alternative storage mechanisms. File-system paths may be expressed as absolute or relative paths and are resolved relative to the current working directory when the tool is run in offline mode, or relative to the server instance directory when the tool is run in task mode. Read the documentation for further information regarding alternative backup storage mechanisms.

`--last`

Show only the last backup for each backend.

Default: false

`-n | --backendName {backendName}`

Show only backups taken from the provided backend.

`--serverId {server ID}`

Show only backups taken from the provided server.

`--storageProperty {PROP:VALUE}`

Assigns a value to a storage property where PROP is the name of the property and VALUE is the single value to be assigned. Specify the same property multiple times in order to assign more than one value to it.

--verify

Verify backups completeness, integrity and whether they can be decrypted.

Default: false

dsbackup purge

Delete one or more backups.

Options

The dsbackup purge command takes the following options:

--backupId {backup ID}

The ID of the backup that should be deleted. Specify this option multiple times to purge multiple backups.

-d | --backupLocation {backup location}

Location containing backups: file-system path or URI for alternative storage mechanisms. File-system paths may be expressed as absolute or relative paths and are resolved relative to the current working directory when the tool is run in offline mode, or relative to the server instance directory when the tool is run in task mode. Read the documentation for further information regarding alternative backup storage mechanisms.

--force

Must be used with the '--olderThan' option, indicates that the last backup of each backend can be deleted if older than the provided duration.

Default: false

--keepCount {number of backups}

The number of backups to keep per backend. Use this option to keep the n latest backups of each backend and delete the others. If n=0, all the backups will be removed.

-n | --backendName {backend name}

Purge only backups of the specified backend. Specify this option multiple times to allow purging backups of different backends. Skip this option to allow purging backups of all backends. This can only be used with options '--keepCount' or '--olderThan'.

--olderThan {duration}

Delete backups that are older than the provided duration. The latest backup of each backend will always be kept unless the '--force' option is also provided. Duration examples: '12 hours', '3 days', '1y'.

`--storageProperty {PROP:VALUE}`

Assigns a value to a storage property where PROP is the name of the property and VALUE is the single value to be assigned. Specify the same property multiple times in order to assign more than one value to it.

dsbackup restore

Restore one or more backends. In order to decrypt and verify signatures on backup files, the server must have access to the master key pair used to encrypt and sign the files when they were created.

Options

The dsbackup restore command takes the following options:

`--backupId {backup ID}`

Restore the backup having the provided ID. Specify this option multiple times to restore multiple backends.

`-d | --backupLocation {backup location}`

Location containing backups: file-system path or URI for alternative storage mechanisms. File-system paths may be expressed as absolute or relative paths and are resolved relative to the current working directory when the tool is run in offline mode, or relative to the server instance directory when the tool is run in task mode. Read the documentation for further information regarding alternative backup storage mechanisms.

`-n | --backendName {backendName}`

Restore the last backup of the provided backend. Specify this option multiple times to restore multiple backends.

`--storageProperty {PROP:VALUE}`

Assigns a value to a storage property where PROP is the name of the property and VALUE is the single value to be assigned. Specify the same property multiple times in order to assign more than one value to it.

Exit Codes

0

The command completed successfully.

> 0

An error occurred.

Chapter 8

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 the server 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 the 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.

`--configFile {configFile}`

Path to the Directory Server configuration file.

Default: /mnt/scratch/workspaces/workspace/ds-release_sustaining_7.0.x/config/config.ldif

`--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

`--offline`

Indicates that the command must be run in offline mode.

Default: false

Configuration Options

`--advanced`

Allows the configuration of advanced components and properties.

Default: false

LDAP 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: uid=admin

-E | --reportAuthzId

Use the authorization identity control.

Default: false

-h | --hostname {host}

Fully-qualified server host name or IP address.

Default: localhost.localdomain

-N | --certNickname {nickname}

Nickname of the certificate that should be sent to the server for SSL client authentication.

-o | --saslOption {name=value}

SASL bind options.

-p | --port {port}

Directory server administration port number.

-T | --trustStorePassword {trustStorePassword}

Truststore cleartext password.

--useJavaKeyStore {keyStorePath}

JKS keystore containing the certificate which should be used for SSL client authentication.

--useJavaTrustStore {trustStorePath}

Use a JKS truststore file for validating server certificate.

--useJceKeyStore {keyStorePath}

JCEKS keystore containing the certificate which should be used for SSL client authentication.

--useJceTrustStore {trustStorePath}

Use a JCEKS truststore file for validating server certificate.

--useJvmTrustStore

Use the a JVM truststore for validating server certificate.

Default: false

--usePasswordPolicyControl

Use the password policy request control.

Default: false

--usePkcs11KeyStore

PKCS#11 keystore containing the certificate which should be used for SSL client authentication.

Default: false

--usePkcs12KeyStore {keyStorePath}

PKCS#12 keystore containing the certificate which should be used for SSL client authentication.

--usePkcs12TrustStore {trustStorePath}

Use a PKCS#12 truststore file for validating server certificate.

-w | --bindPassword {bindPassword}

Password to use to bind to the server. Omit this option while providing the bind DN to ensure that the command prompts for the password, rather than entering the password as a command argument.

-W | --keyStorePassword {keyStorePassword}

Keystore cleartext password.

-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 get and set component properties. Subcommands 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 "uid=admin" \
 --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 "uid=admin" \
 --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
- `y`: years

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

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

```
$ dsconfig --hostname opendj.example.com --port 4444 --bindDn "uid=admin" --bindPassword password

>>>> OpenDJ configuration console main menu

What do you want to configure?

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

    a)   show advanced components and properties
    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 "uid=admin" \
 --bindPassword password \
 --commandFilePath ~/enable-debug-log.batch

$ cat ~/enable-debug-log.batch
# dsconfig session start date: <date>

# Session operation number: 1
# Operation date: <date>
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 uid=admin \
          --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 uid=admin \
          --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
uid=admin
--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 the dsconfig command, 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.

Chapter 9
dskeymgr — manage public key infrastructure in private deployments

Synopsis

dskeymgr {subcommand} {options}

Description

This utility can be used for provisioning and managing TLS certificates for use in private deployments.

Subcommands easily allow to:

- Create a deployment CA certificate
- Distribute the CA certificate to all deployed applications
- Provision each application with a TLS key pair signed by the deployment CA
- Rotate the TLS key pairs

Subcommands take several seconds to run because the tool uses a computationally expensive algorithm for hashing the deployment key password.

Options

The dskeymgr command takes the following options:

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

General options:

-V | --version

Display Directory Server version information.

Default: false

-H | --help

Display this usage information.

Default: false

Subcommands

The dskeymgr command supports the following subcommands:

dskeymgr create-deployment-key

Creates a new deployment key.

Options

The dskeymgr create-deployment-key command takes the following options:

-f | --outputFile {outputFile}

Optional path to a file where the deployment key will be written, overwriting the file if it exists.

-v | --validity {validity}

The duration for which the CA certificate associated with the deployment key will be valid. Examples: '20years', '1days'.

Default: 10 y

-w | --deploymentKeyPassword {deploymentKeyPassword}

The deployment key password.

dskeymgr create-tls-key-pair

Creates a TLS key-pair signed by the CA associated with a deployment key and then adds it to a keystore with a given alias (default: 'ssl-key-pair'), overwriting any existing key-pair with the same alias.

Options

The dskeymgr create-tls-key-pair command takes the following options:

-a | --alias {alias}

The TLS key-pair alias, any entry with the same alias will be overwritten.

Default: ssl-key-pair

-h | --hostname {hostname}

The hostname(s) that will be added to the TLS certificate alternative name extension. Multiple hostnames may be given by providing this argument multiple times. Hostnames can start with a wildcard.

Default: localhost

-k | --deploymentKey {deploymentKey}

The deployment key.

-K | --keyStoreFile {keyStoreFile}

Path to an existing PKCS12 keystore file or a path indicating where a new keystore file should be created.

-s | --subjectDn {subjectDn}

The TLS certificate subject DN.

-v | --validity {validity}

The duration for which the TLS certificate will be valid. Examples: '1days', '12hours', '1d 12h'.

Default: 1 y

-w | --deploymentKeyPassword {deploymentKeyPassword}

The deployment key password.

-W | --keyStorePassword {keyStorePassword}

Keystore cleartext password.

dskeymgr export-ca-cert

Exports the CA certificate associated with a deployment key to a keystore or as a PEM file.

Options

The dskeymgr export-ca-cert command takes the following options:

-a | --alias {alias}

The CA certificate alias, must not already exist in the keystore.

Default: ca-cert

-f | --outputFile {outputFile}

Optional path to a file where the CA certificate will be written in the PEM format, overwriting the file if it exists.

-k | --deploymentKey {deploymentKey}

The deployment key.

-K | --keyStoreFile {keyStoreFile}

Path to an existing PKCS12 keystore file or a path indicating where a new keystore file should be created.

-w | --deploymentKeyPassword {deploymentKeyPassword}

The deployment key password.

-W | --keyStorePassword {keyStorePassword}

Keystore cleartext password.

dskeymgr export-master-key-pair

Exports the master key pair associated with a deployment key to a keystore.

Options

The dskeymgr export-master-key-pair command takes the following options:

-a | --alias {alias}

The master key pair alias, must not already exist in the keystore.

Default: master-key

-k | --deploymentKey {deploymentKey}

The deployment key.

-K | --keyStoreFile {keyStoreFile}

Path to an existing PKCS12 keystore file or a path indicating where a new keystore file should be created.

-w | --deploymentKeyPassword {deploymentKeyPassword}

The deployment key password.

-W | --keyStorePassword {keyStorePassword}

Keystore cleartext password.

Exit Codes

0

The command completed successfully.

> 0

An error occurred.

Examples

The following example shows how to create a deployment key for managing the public key infrastructure of a private deployment:

```
$ dskeymgr \
 create-deployment-key \
 --deploymentKeyPassword password \
 --validity "10 years"
AFPxL0RlmdMZHeVkkcC3GYFsAHNlNQ5CBVN1bkVDM7FyW2gWxnvQdQ
```

The following examples show how to use a deployment key to obtain the deployment CA certificate:

- Export the CA certificate to a file in PEM format:

  ```
  $ dskeymgr \
   export-ca-cert \
   --deploymentKey AFPxL0RlmdMZHeVkkcC3GYFsAHNlNQ5CBVN1bkVDM7FyW2gWxnvQdQ \
   --deploymentKeyPassword password \
   > ca.pem
  ```

- Export the CA certificate to a PKCS#12 truststore, creating the truststore if it does not exist:

  ```
  $ dskeymgr \
   export-ca-cert \
   --deploymentKey AFPxL0RlmdMZHeVkkcC3GYFsAHNlNQ5CBVN1bkVDM7FyW2gWxnvQdQ \
   --deploymentKeyPassword password \
   --keyStoreFile keystore \
   --keyStorePassword secret12 \
   --alias ca-cert
  ```

The following example shows how to use a deployment key to generate a TLS key pair signed by the deployment CA certificate and add it to a PKCS#12 keystore, creating the keystore if the keystore file does not exist. In this example, the key pair must be used by an application hosted on `*.example.com` and the application's entry has the DN `cn=test account,cn=service`.

```
$ dskeymgr \
 create-tls-key-pair \
 --deploymentKey AFPxL0RlmdMZHeVkkcC3GYFsAHNlNQ5CBVN1bkVDM7FyW2gWxnvQdQ \
 --deploymentKeyPassword password \
 --subjectDn "cn=test account,cn=service" \
 --hostname "*.example.com" \
 --validity "1 days" \
 --keyStoreFile keystore \
 --keyStorePassword secret12 \
 --alias tls-key-pair
```

In the example above, the key pair is only valid for one day. When it is about to expire, run the same command again to replace the old key pair having the alias `tls-key-pair` with a new one.

Chapter 10

dsrepl — Manages data synchronization between servers

Synopsis

dsrepl {subcommand} {options}

Description

This tool manages data synchronization between servers. For replication to work you must initialize the contents of one of the servers with the contents of the others using the 'initialize' subcommand.

Options

The dsrepl command takes the following options:

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.

`-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

Subcommands

The dsrepl command supports the following subcommands:

dsrepl add-local-server-to-pre-7-0-topology

Adds the local server (with version 7.0 or more) to a topology with older server versions (prior to 7.0).

Options

The dsrepl add-local-server-to-pre-7-0-topology command takes the following options:

SubCommand Options:

`-b | --baseDn {baseDN}`

Base DN(s) to replicate.

`--masterKeyPairCertAlias {masterKeyPairCertAlias}`

Alias of the shared master key to use for protecting secret keys.

Default: master-key

`--rootCaCertAlias {rootCaCertAlias}`

Alias for the root CA certificate.

Default: ca-cert

LDAP 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=admin,cn=Administrators,cn=admin data

-E | --reportAuthzId

Use the authorization identity control.

Default: false

-h | --hostname {host}

Fully-qualified server host name or IP address.

Default: localhost.localdomain

-N | --certNickname {nickname}

Nickname of the certificate that should be sent to the server for SSL client authentication.

-o | --saslOption {name=value}

SASL bind options.

-p | --port {port}

Directory server administration port number.

-T | --trustStorePassword {trustStorePassword}

Truststore cleartext password.

--useJavaKeyStore {keyStorePath}

JKS keystore containing the certificate which should be used for SSL client authentication.

--useJavaTrustStore {trustStorePath}

Use a JKS truststore file for validating server certificate.

--useJceKeyStore {keyStorePath}

JCEKS keystore containing the certificate which should be used for SSL client authentication.

`--useJceTrustStore {trustStorePath}`

Use a JCEKS truststore file for validating server certificate.

`--useJvmTrustStore`

Use the a JVM truststore for validating server certificate.

Default: false

`--usePasswordPolicyControl`

Use the password policy request control.

Default: false

`--usePkcs11KeyStore`

PKCS#11 keystore containing the certificate which should be used for SSL client authentication.

Default: false

`--usePkcs12KeyStore {keyStorePath}`

PKCS#12 keystore containing the certificate which should be used for SSL client authentication.

`--usePkcs12TrustStore {trustStorePath}`

Use a PKCS#12 truststore file for validating server certificate.

`-w | --bindPassword {bindPassword}`

Password to use to bind to the server. Omit this option while providing the bind DN to ensure that the command prompts for the password, rather than entering the password as a command argument.

`-W | --keyStorePassword {keyStorePassword}`

Keystore cleartext password.

`-X | --trustAll`

Trust all server SSL certificates.

Default: false

dsrepl cleanup-migrated-pre-7-0-topology

Clean all the servers (with version 7.0 or more) that have been migrated from a topology of older servers (version prior to 7.0).

Options

The dsrepl cleanup-migrated-pre-7-0-topology command takes the following options:

SubCommand Options:

`--bootstrapServer {serverSource}`

Server ID of the server containing the source data.

LDAP 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: uid=admin

`-E | --reportAuthzId`

Use the authorization identity control.

Default: false

`-h | --hostname {host}`

Fully-qualified server host name or IP address.

Default: localhost.localdomain

`-N | --certNickname {nickname}`

Nickname of the certificate that should be sent to the server for SSL client authentication.

`-o | --saslOption {name=value}`

SASL bind options.

`-p | --port {port}`

Directory server administration port number.

`-T | --trustStorePassword {trustStorePassword}`

Truststore cleartext password.

`--useJavaKeyStore {keyStorePath}`

JKS keystore containing the certificate which should be used for SSL client authentication.

`--useJavaTrustStore {trustStorePath}`

Use a JKS truststore file for validating server certificate.

`--useJceKeyStore {keyStorePath}`

JCEKS keystore containing the certificate which should be used for SSL client authentication.

`--useJceTrustStore {trustStorePath}`

Use a JCEKS truststore file for validating server certificate.

`--useJvmTrustStore`

Use the a JVM truststore for validating server certificate.

Default: false

`--usePasswordPolicyControl`

Use the password policy request control.

Default: false

`--usePkcs11KeyStore`

PKCS#11 keystore containing the certificate which should be used for SSL client authentication.

Default: false

`--usePkcs12KeyStore {keyStorePath}`

PKCS#12 keystore containing the certificate which should be used for SSL client authentication.

`--usePkcs12TrustStore {trustStorePath}`

Use a PKCS#12 truststore file for validating server certificate.

`-w | --bindPassword {bindPassword}`

Password to use to bind to the server. Omit this option while providing the bind DN to ensure that the command prompts for the password, rather than entering the password as a command argument.

`-W | --keyStorePassword {keyStorePassword}`

Keystore cleartext password.

-X | --trustAll

Trust all server SSL certificates.

Default: false

dsrepl clear-changelog

Clears all replication server changelog data for the offline local server; the other replication servers in the topology will transfer any needed data when the server restarts.

dsrepl end-disaster-recovery

End disaster recovery for all servers.

Options

The dsrepl end-disaster-recovery command takes the following options:

SubCommand Options:

-b | --baseDn {baseDN}

Base DN(s) to use. Multiple base DNs can be provided by using this option multiple times.

LDAP 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: uid=admin

-E | --reportAuthzId

Use the authorization identity control.

Default: false

-h | --hostname {host}

Fully-qualified server host name or IP address.

Default: localhost.localdomain

-N | --certNickname {nickname}

Nickname of the certificate that should be sent to the server for SSL client authentication.

-o | --saslOption {name=value}

SASL bind options.

-p | --port {port}

Directory server administration port number.

-T | --trustStorePassword {trustStorePassword}

Truststore cleartext password.

--useJavaKeyStore {keyStorePath}

JKS keystore containing the certificate which should be used for SSL client authentication.

--useJavaTrustStore {trustStorePath}

Use a JKS truststore file for validating server certificate.

--useJceKeyStore {keyStorePath}

JCEKS keystore containing the certificate which should be used for SSL client authentication.

--useJceTrustStore {trustStorePath}

Use a JCEKS truststore file for validating server certificate.

--useJvmTrustStore

Use the a JVM truststore for validating server certificate.

Default: false

--usePasswordPolicyControl

Use the password policy request control.

Default: false

--usePkcs11KeyStore

PKCS#11 keystore containing the certificate which should be used for SSL client authentication.

Default: false

`--usePkcs12KeyStore {keyStorePath}`

PKCS#12 keystore containing the certificate which should be used for SSL client authentication.

`--usePkcs12TrustStore {trustStorePath}`

Use a PKCS#12 truststore file for validating server certificate.

`-w | --bindPassword {bindPassword}`

Password to use to bind to the server. Omit this option while providing the bind DN to ensure that the command prompts for the password, rather than entering the password as a command argument.

`-W | --keyStorePassword {keyStorePassword}`

Keystore cleartext password.

`-X | --trustAll`

Trust all server SSL certificates.

Default: false

dsrepl initialize

Initialize replication data for the server.

Options

The dsrepl initialize command takes the following options:

SubCommand Options:

`-b | --baseDn {baseDN}`

Base DN(s) to use. Multiple base DNs can be provided by using this option multiple times.

`--fromServer {serverSource}`

Server ID of the server containing the source data.

`--toAllServers`

Initialize all the other servers in the topology.

Default: false

`--toServer {serverToInitialize}`

Server ID of the server to be initialized.

LDAP 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: uid=admin

-E | --reportAuthzId

Use the authorization identity control.

Default: false

-h | --hostname {host}

Fully-qualified server host name or IP address.

Default: localhost.localdomain

-N | --certNickname {nickname}

Nickname of the certificate that should be sent to the server for SSL client authentication.

-o | --saslOption {name=value}

SASL bind options.

-p | --port {port}

Directory server administration port number.

-T | --trustStorePassword {trustStorePassword}

Truststore cleartext password.

--useJavaKeyStore {keyStorePath}

JKS keystore containing the certificate which should be used for SSL client authentication.

--useJavaTrustStore {trustStorePath}

Use a JKS truststore file for validating server certificate.

--useJceKeyStore {keyStorePath}

JCEKS keystore containing the certificate which should be used for SSL client authentication.

--useJceTrustStore {trustStorePath}

Use a JCEKS truststore file for validating server certificate.

--useJvmTrustStore

Use the a JVM truststore for validating server certificate.

Default: false

--usePasswordPolicyControl

Use the password policy request control.

Default: false

--usePkcs11KeyStore

PKCS#11 keystore containing the certificate which should be used for SSL client authentication.

Default: false

--usePkcs12KeyStore {keyStorePath}

PKCS#12 keystore containing the certificate which should be used for SSL client authentication.

--usePkcs12TrustStore {trustStorePath}

Use a PKCS#12 truststore file for validating server certificate.

-w | --bindPassword {bindPassword}

Password to use to bind to the server. Omit this option while providing the bind DN to ensure that the command prompts for the password, rather than entering the password as a command argument.

-W | --keyStorePassword {keyStorePassword}

Keystore cleartext password.

-X | --trustAll

Trust all server SSL certificates.

Default: false

dsrepl purge-meta-data

Purges old replication meta-data from application data.

Options

The dsrepl purge-meta-data command takes the following options:

SubCommand Options:

`-b | --baseDn {baseDN}`

Base DN(s) to use. Multiple base DNs can be provided by using this option multiple times.

`--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.

`--description {description}`

Gives a description to the task.

`--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.

`--maximumDuration {maximum duration in seconds}`

Maximum duration of the command in seconds.

Default: 3600

`--recurringTask {schedulePattern}`

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

The schedule pattern for a recurring task supports only the following crontab features:

Time and Date Fields

Field	Allowed Values
minute	0-59

Field	Allowed Values
hour	0-23
day of month	1-31
month	1-12 (or names)
day of week	0-7 (0 or 7 is Sunday, or use names)

A field can contain an asterisk, `*`. An asterisk stands for *`first-last`*.

Fields can include ranges of numbers. A range is two numbers separated by a hyphen, and is inclusive. For example, `8-10` for an "hour" field means execution at hours 8, 9, and 10.

Fields can include lists. A list is a set of numbers or ranges separated by commas. For example, `4, 8-10` for an "hour" field means execution at hours 4, 8, 9, and 10.

When using names for in "month" or "day of week" fields, use the first three letters of the particular month or day of the week. Case does not matter. Ranges and lists of names are not supported.

`-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.

`--taskId {taskID}`

Gives an ID to the task.

LDAP 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: uid=admin

`-E | --reportAuthzId`

Use the authorization identity control.

Default: false

-h | --hostname {host}

Fully-qualified server host name or IP address.

Default: localhost.localdomain

-N | --certNickname {nickname}

Nickname of the certificate that should be sent to the server for SSL client authentication.

-o | --saslOption {name=value}

SASL bind options.

-p | --port {port}

Directory server administration port number.

-T | --trustStorePassword {trustStorePassword}

Truststore cleartext password.

--useJavaKeyStore {keyStorePath}

JKS keystore containing the certificate which should be used for SSL client authentication.

--useJavaTrustStore {trustStorePath}

Use a JKS truststore file for validating server certificate.

--useJceKeyStore {keyStorePath}

JCEKS keystore containing the certificate which should be used for SSL client authentication.

--useJceTrustStore {trustStorePath}

Use a JCEKS truststore file for validating server certificate.

--useJvmTrustStore

Use the a JVM truststore for validating server certificate.

Default: false

--usePasswordPolicyControl

Use the password policy request control.

Default: false

--usePkcs11KeyStore

PKCS#11 keystore containing the certificate which should be used for SSL client authentication.

Default: false

--usePkcs12KeyStore {keyStorePath}

PKCS#12 keystore containing the certificate which should be used for SSL client authentication.

--usePkcs12TrustStore {trustStorePath}

Use a PKCS#12 truststore file for validating server certificate.

-w | --bindPassword {bindPassword}

Password to use to bind to the server. Omit this option while providing the bind DN to ensure that the command prompts for the password, rather than entering the password as a command argument.

-W | --keyStorePassword {keyStorePassword}

Keystore cleartext password.

-X | --trustAll

Trust all server SSL certificates.

Default: false

dsrepl reset-change-number

Re-synchronizes the change-log change number of the target server with the change-log change number of the source server.

Options

The dsrepl reset-change-number command takes the following options:

SubCommand Options:

--change-number {change number}

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

--sourceBindDn {bindDN}

DN to use to bind to the server.

Default: uid=admin

`--sourceBindPassword {bindPassword}`

Password to use to bind to the server. Omit this option while providing the bind DN to ensure that the command prompts for the password, rather than entering the password as a command argument.

`--sourceHostname {host}`

Directory server hostname or IP address.

Default: localhost.localdomain

`--sourcePort {port}`

Directory server administration port number.

`--targetBindDn {bindDN}`

DN to use to bind to the server.

Default: uid=admin

`--targetBindPassword {bindPassword}`

Password to use to bind to the server. Omit this option while providing the bind DN to ensure that the command prompts for the password, rather than entering the password as a command argument.

`--targetHostname {host}`

Directory server hostname or IP address.

Default: localhost.localdomain

`--targetPort {port}`

Directory server administration port number.

LDAP 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

`-E | --reportAuthzId`

Use the authorization identity control.

Default: false

`-N | --certNickname {nickname}`

Nickname of the certificate that should be sent to the server for SSL client authentication.

`-o | --saslOption {name=value}`

SASL bind options.

`-T | --trustStorePassword {trustStorePassword}`

Truststore cleartext password.

`--useJavaKeyStore {keyStorePath}`

JKS keystore containing the certificate which should be used for SSL client authentication.

`--useJavaTrustStore {trustStorePath}`

Use a JKS truststore file for validating server certificate.

`--useJceKeyStore {keyStorePath}`

JCEKS keystore containing the certificate which should be used for SSL client authentication.

`--useJceTrustStore {trustStorePath}`

Use a JCEKS truststore file for validating server certificate.

`--useJvmTrustStore`

Use the a JVM truststore for validating server certificate.

Default: false

`--usePasswordPolicyControl`

Use the password policy request control.

Default: false

`--usePkcs11KeyStore`

PKCS#11 keystore containing the certificate which should be used for SSL client authentication.

Default: false

`--usePkcs12KeyStore {keyStorePath}`

PKCS#12 keystore containing the certificate which should be used for SSL client authentication.

`--usePkcs12TrustStore {trustStorePath}`

Use a PKCS#12 truststore file for validating server certificate.

`-W | --keyStorePassword {keyStorePassword}`

Keystore cleartext password.

`-X | --trustAll`

Trust all server SSL certificates.

Default: false

dsrepl start-disaster-recovery

Start disaster recovery for all servers.

Options

The dsrepl start-disaster-recovery command takes the following options:

SubCommand Options:

`-b | --baseDn {baseDN}`

Base DN(s) to use. Multiple base DNs can be provided by using this option multiple times.

LDAP 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: uid=admin

`-E | --reportAuthzId`

Use the authorization identity control.

Default: false

`-h | --hostname {host}`

Fully-qualified server host name or IP address.

Default: localhost.localdomain

-N | --certNickname {nickname}

Nickname of the certificate that should be sent to the server for SSL client authentication.

-o | --saslOption {name=value}

SASL bind options.

-p | --port {port}

Directory server administration port number.

-T | --trustStorePassword {trustStorePassword}

Truststore cleartext password.

--useJavaKeyStore {keyStorePath}

JKS keystore containing the certificate which should be used for SSL client authentication.

--useJavaTrustStore {trustStorePath}

Use a JKS truststore file for validating server certificate.

--useJceKeyStore {keyStorePath}

JCEKS keystore containing the certificate which should be used for SSL client authentication.

--useJceTrustStore {trustStorePath}

Use a JCEKS truststore file for validating server certificate.

--useJvmTrustStore

Use the a JVM truststore for validating server certificate.

Default: false

--usePasswordPolicyControl

Use the password policy request control.

Default: false

--usePkcs11KeyStore

PKCS#11 keystore containing the certificate which should be used for SSL client authentication.

Default: false

--usePkcs12KeyStore {keyStorePath}

PKCS#12 keystore containing the certificate which should be used for SSL client authentication.

--usePkcs12TrustStore {trustStorePath}

Use a PKCS#12 truststore file for validating server certificate.

-w | --bindPassword {bindPassword}

Password to use to bind to the server. Omit this option while providing the bind DN to ensure that the command prompts for the password, rather than entering the password as a command argument.

-W | --keyStorePassword {keyStorePassword}

Keystore cleartext password.

-X | --trustAll

Trust all server SSL certificates.

Default: false

dsrepl status

Displays the status of the replication service and various diagnostics about it. The information is derived from reading cn=monitor on all the servers in the replication topology. A server receives a LATE status when its replay delay exceeds five seconds.

Options

The dsrepl status command takes the following options:

SubCommand Options:

-b | --baseDn {baseDN}

Base DN(s) to display. Multiple base DNs can be provided by using this option multiple times. If no base DNs are provided, then all the base DNs will be displayed.

--showChangelogs

Displays individual changelog servers in the output.

Default: false

--showGroups

Display replication group information in the output.

Default: false

--showReplicas

Displays individual replicas in the output.

Default: false

LDAP 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: uid=monitor

-E | --reportAuthzId

Use the authorization identity control.

Default: false

-h | --hostname {host}

Fully-qualified server host name or IP address.

Default: localhost.localdomain

-N | --certNickname {nickname}

Nickname of the certificate that should be sent to the server for SSL client authentication.

-o | --saslOption {name=value}

SASL bind options.

-p | --port {port}

Directory server administration port number.

-T | --trustStorePassword {trustStorePassword}

Truststore cleartext password.

--useJavaKeyStore {keyStorePath}

JKS keystore containing the certificate which should be used for SSL client authentication.

--useJavaTrustStore {trustStorePath}

Use a JKS truststore file for validating server certificate.

--useJceKeyStore {keyStorePath}

JCEKS keystore containing the certificate which should be used for SSL client authentication.

--useJceTrustStore {trustStorePath}

Use a JCEKS truststore file for validating server certificate.

--useJvmTrustStore

Use the a JVM truststore for validating server certificate.

Default: false

--usePasswordPolicyControl

Use the password policy request control.

Default: false

--usePkcs11KeyStore

PKCS#11 keystore containing the certificate which should be used for SSL client authentication.

Default: false

--usePkcs12KeyStore {keyStorePath}

PKCS#12 keystore containing the certificate which should be used for SSL client authentication.

--usePkcs12TrustStore {trustStorePath}

Use a PKCS#12 truststore file for validating server certificate.

-w | --bindPassword {bindPassword}

Password to use to bind to the server. Omit this option while providing the bind DN to ensure that the command prompts for the password, rather than entering the password as a command argument.

-W | --keyStorePassword {keyStorePassword}

Keystore cleartext password.

-X | --trustAll

Trust all server SSL certificates.

Default: false

Exit Codes

0

The command completed successfully.

> 0

An error occurred.

Examples:

To be completed

Chapter 11

encode-password — encode a password with a storage scheme

Synopsis

encode-password {options}

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 compares a password with the encoded value:

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

$ encode-password --clearPassword secret12 --storageScheme CRYPT
{CRYPT}ZulJ6Dy3TFnrE

$ encode-password \
 --clearPassword secret12 \
 --storageScheme CRYPT \
 --encodedPassword "{CRYPT}ZulJ6Dy3TFnrE" \
 --useCompareResultCode
The provided clear-text and encoded passwords match

$ echo $?
6
```

Chapter 12
export-ldif — export directory data in LDIF

Synopsis

export-ldif {options}

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.

--excludeFilter {filter}

Filter to identify entries to exclude from the LDIF export.

-i | --includeAttribute {attribute}

Attribute to include in the LDIF export.

--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 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.

--description {description}

Gives a description to the task.

--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.

The schedule pattern for a recurring task supports only the following crontab features:

Time and Date Fields

Field	Allowed Values
minute	0-59
hour	0-23
day of month	1-31
month	1-12 (or names)
day of week	0-7 (0 or 7 is Sunday, or use names)

A field can contain an asterisk, `*`. An asterisk stands for *`first-last`*.

Fields can include ranges of numbers. A range is two numbers separated by a hyphen, and is inclusive. For example, `8-10` for an "hour" field means execution at hours 8, 9, and 10.

Fields can include lists. A list is a set of numbers or ranges separated by commas. For example, `4, 8-10` for an "hour" field means execution at hours 4, 8, 9, and 10.

When using names for in "month" or "day of week" fields, use the first three letters of the particular month or day of the week. Case does not matter. Ranges and lists of names are not supported.

`-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.

`--taskId {taskID}`

Gives an ID to the task.

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: uid=admin

-E | --reportAuthzId

Use the authorization identity control.

Default: false

-h | --hostname {host}

Fully-qualified server host name or IP address.

Default: localhost.localdomain

-N | --certNickname {nickname}

Nickname of the certificate that should be sent to the server for SSL client authentication.

-o | --saslOption {name=value}

SASL bind options.

-p | --port {port}

Directory server administration port number.

-T | --trustStorePassword {trustStorePassword}

Truststore cleartext password.

--useJavaKeyStore {keyStorePath}

JKS keystore containing the certificate which should be used for SSL client authentication.

--useJavaTrustStore {trustStorePath}

Use a JKS truststore file for validating server certificate.

--useJceKeyStore {keyStorePath}

JCEKS keystore containing the certificate which should be used for SSL client authentication.

`--useJceTrustStore {trustStorePath}`

Use a JCEKS truststore file for validating server certificate.

`--useJvmTrustStore`

Use the a JVM truststore for validating server certificate.

Default: false

`--usePasswordPolicyControl`

Use the password policy request control.

Default: false

`--usePkcs11KeyStore`

PKCS#11 keystore containing the certificate which should be used for SSL client authentication.

Default: false

`--usePkcs12KeyStore {keyStorePath}`

PKCS#12 keystore containing the certificate which should be used for SSL client authentication.

`--usePkcs12TrustStore {trustStorePath}`

Use a PKCS#12 truststore file for validating server certificate.

`-w | --bindPassword {bindPassword}`

Password to use to bind to the server. Omit this option while providing the bind DN to ensure that the command prompts for the password, rather than entering the password as a command argument.

`-W | --keyStorePassword {keyStorePassword}`

Keystore cleartext password.

`-X | --trustAll`

Trust all server SSL certificates.

Default: false

Utility input/output options:

`--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.

`--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 \
 --includeBranch dc=example,dc=com \
 --backendId userData \
 --ldifFile /path/to/opendj/ldif/Example.ldif \
 --offline
... category=BACKEND severity=INFORMATION ...
...Exported 160 entries and skipped 0 in 0 seconds (average rate 1428.6/sec)
```

Chapter 13

import-ldif — import directory data from LDIF

Synopsis

import-ldif {options}

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.

`--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.

`--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. When using this option, the command writes to server files. Run the command as a user having the same filesystem permissions as the user running the server.

Default: false

`-R | --rejectFile {rejectFile}`

Write rejected entries to the specified file.

`-s | --randomSeed {seed}`

Seed for the MakeLDIF random number generator. To always generate the same data with the same command, use the same non-zero seed value. A value of zero (the default) results in different data each time the tool is run.

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 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.

`--description {description}`

Gives a description to the task.

`--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.

The schedule pattern for a recurring task supports only the following crontab features:

Time and Date Fields

Field	Allowed Values
minute	0-59
hour	0-23
day of month	1-31
month	1-12 (or names)
day of week	0-7 (0 or 7 is Sunday, or use names)

A field can contain an asterisk, `*`. An asterisk stands for *`first-last`*.

Fields can include ranges of numbers. A range is two numbers separated by a hyphen, and is inclusive. For example, `8-10` for an "hour" field means execution at hours 8, 9, and 10.

Fields can include lists. A list is a set of numbers or ranges separated by commas. For example, `4, 8-10` for an "hour" field means execution at hours 4, 8, 9, and 10.

When using names for in "month" or "day of week" fields, use the first three letters of the particular month or day of the week. Case does not matter. Ranges and lists of names are not supported.

`-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.

`--taskId {taskID}`

Gives an ID to the task.

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: uid=admin

-E | --reportAuthzId

Use the authorization identity control.

Default: false

-h | --hostname {host}

Fully-qualified server host name or IP address.

Default: localhost.localdomain

-N | --certNickname {nickname}

Nickname of the certificate that should be sent to the server for SSL client authentication.

-o | --saslOption {name=value}

SASL bind options.

-p | --port {port}

Directory server administration port number.

-T | --trustStorePassword {trustStorePassword}

Truststore cleartext password.

--useJavaKeyStore {keyStorePath}

JKS keystore containing the certificate which should be used for SSL client authentication.

--useJavaTrustStore {trustStorePath}

Use a JKS truststore file for validating server certificate.

--useJceKeyStore {keyStorePath}

JCEKS keystore containing the certificate which should be used for SSL client authentication.

--useJceTrustStore {trustStorePath}

Use a JCEKS truststore file for validating server certificate.

--useJvmTrustStore

Use the a JVM truststore for validating server certificate.

Default: false

`--usePasswordPolicyControl`

Use the password policy request control.

Default: false

`--usePkcs11KeyStore`

PKCS#11 keystore containing the certificate which should be used for SSL client authentication.

Default: false

`--usePkcs12KeyStore {keyStorePath}`

PKCS#12 keystore containing the certificate which should be used for SSL client authentication.

`--usePkcs12TrustStore {trustStorePath}`

Use a PKCS#12 truststore file for validating server certificate.

`-w | --bindPassword {bindPassword}`

Password to use to bind to the server. Omit this option while providing the bind DN to ensure that the command prompts for the password, rather than entering the password as a command argument.

`-W | --keyStorePassword {keyStorePassword}`

Keystore cleartext password.

`-X | --trustAll`

Trust all server SSL certificates.

Default: false

Utility input/output options:

`--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 (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 in the current directory, `Example.ldif`, with the server offline:

```
$ import-ldif --includeBranch dc=example,dc=com --backendId userData --ldifFile Example.ldif --offline
... category=RUNTIME_INFORMATION severity=NOTICE...
... msg=Import LDIF environment close took 0 seconds
```

Chapter 14
ldapcompare — perform LDAP compare operations

Synopsis

ldapcompare {options} 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.

`-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 values are not case-sensitive:

`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 DS server 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

`PwdPolicy`
`PasswordPolicy`

Password Policy Request Control, Object Identifier: 1.3.6.1.4.1.42.2.27.8.5.1

`PasswordQualityAdvice`

Password Quality Advice Request Control, Object Identifier: 1.3.6.1.4.1.36733.2.1.5.5

`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

`RelaxRules`

Relax Rules Request Control, Object Identifier: 1.3.6.1.4.1.4203.666.5.12

`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 and do not contact the server.

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:

`--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:

`-E | --reportAuthzId`

Use the authorization identity control.

Default: false

`-h | --hostname {host}`

Fully-qualified server host name or IP address.

Default: localhost.localdomain

`-N | --certNickname {nickname}`

Nickname of the certificate that should be sent to the server for SSL client authentication.

-o | --saslOption {name=value}

SASL bind options.

-p | --port {port}

Directory server port number.

-q | --useStartTls

Use StartTLS to secure communication with the server.

Default: false

-T | --trustStorePassword {trustStorePassword}

Truststore cleartext password.

--useJavaKeyStore {keyStorePath}

JKS keystore containing the certificate which should be used for SSL client authentication.

--useJavaTrustStore {trustStorePath}

Use a JKS truststore file for validating server certificate.

--useJceKeyStore {keyStorePath}

JCEKS keystore containing the certificate which should be used for SSL client authentication.

--useJceTrustStore {trustStorePath}

Use a JCEKS truststore file for validating server certificate.

--useJvmTrustStore

Use the a JVM truststore for validating server certificate.

Default: false

--usePasswordPolicyControl

Use the password policy request control.

Default: false

--usePkcs11KeyStore

PKCS#11 keystore containing the certificate which should be used for SSL client authentication.

Default: false

`--usePkcs12KeyStore {keyStorePath}`

PKCS#12 keystore containing the certificate which should be used for SSL client authentication.

`--usePkcs12TrustStore {trustStorePath}`

Use a PKCS#12 truststore file for validating server certificate.

`-w | --bindPassword {bindPassword}`

Password to use to bind to the server. Omit this option while providing the bind DN to ensure that the command prompts for the password, rather than entering the password as a command argument.

`-W | --keyStorePassword {keyStorePassword}`

Keystore cleartext password.

`-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:

`--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.

`-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 --port 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 --port 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
```

Chapter 15

ldapdelete — perform LDAP delete operations

Synopsis

ldapdelete {options} [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

`-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 values are not case-sensitive:

`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 DS server 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

`PwdPolicy`
`PasswordPolicy`

Password Policy Request Control, Object Identifier: 1.3.6.1.4.1.42.2.27.8.5.1

`PasswordQualityAdvice`

Password Quality Advice Request Control, Object Identifier: 1.3.6.1.4.1.36733.2.1.5.5

`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

`RelaxRules`

Relax Rules Request Control, Object Identifier: 1.3.6.1.4.1.4203.666.5.12

`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 and do not contact the server.

Default: false

`--numConnections {numConnections}`

Number of connections.

Default: 1

`-x | --deleteSubtree`

Delete the specified entry and all entries below it.

Default: false

`-Y | --proxyAs {authzID}`

Use the proxied authorization control with the given authorization ID.

LDAP 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:

`-E | --reportAuthzId`

Use the authorization identity control.

Default: false

`-h | --hostname {host}`

Fully-qualified server host name or IP address.

Default: localhost.localdomain

`-N | --certNickname {nickname}`

Nickname of the certificate that should be sent to the server for SSL client authentication.

-o | --saslOption {name=value}

SASL bind options.

-p | --port {port}

Directory server port number.

-q | --useStartTls

Use StartTLS to secure communication with the server.

Default: false

-T | --trustStorePassword {trustStorePassword}

Truststore cleartext password.

--useJavaKeyStore {keyStorePath}

JKS keystore containing the certificate which should be used for SSL client authentication.

--useJavaTrustStore {trustStorePath}

Use a JKS truststore file for validating server certificate.

--useJceKeyStore {keyStorePath}

JCEKS keystore containing the certificate which should be used for SSL client authentication.

--useJceTrustStore {trustStorePath}

Use a JCEKS truststore file for validating server certificate.

--useJvmTrustStore

Use the a JVM truststore for validating server certificate.

Default: false

--usePasswordPolicyControl

Use the password policy request control.

Default: false

--usePkcs11KeyStore

PKCS#11 keystore containing the certificate which should be used for SSL client authentication.

Default: false

`--usePkcs12KeyStore {keyStorePath}`

PKCS#12 keystore containing the certificate which should be used for SSL client authentication.

`--usePkcs12TrustStore {trustStorePath}`

Use a PKCS#12 truststore file for validating server certificate.

`-w | --bindPassword {bindPassword}`

Password to use to bind to the server. Omit this option while providing the bind DN to ensure that the command prompts for the password, rather than entering the password as a command argument.

`-W | --keyStorePassword {keyStorePassword}`

Keystore cleartext password.

`-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:

`--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.

-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 \
 --port 1389 \
 --bindDn uid=admin \
 --bindPassword 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 \
 --port 1389 \
 --bindDn uid=admin \
 --bindPassword password \
 --deleteSubtree \
 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
```

Chapter 16

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

Synopsis

ldapmodify {options} [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

`-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 values are not case-sensitive:

`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 DS server 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

`PwdPolicy`
`PasswordPolicy`

Password Policy Request Control, Object Identifier: 1.3.6.1.4.1.42.2.27.8.5.1

`PasswordQualityAdvice`

Password Quality Advice Request Control, Object Identifier: 1.3.6.1.4.1.36733.2.1.5.5

`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

`RelaxRules`

Relax Rules Request Control, Object Identifier: 1.3.6.1.4.1.4203.666.5.12

`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 and do not contact the server.

Default: false

`--numConnections {numConnections}`

Number of connections.

Default: 1

`--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:

`--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:

-E | --reportAuthzId

Use the authorization identity control.

Default: false

-h | --hostname {host}

Fully-qualified server host name or IP address.

Default: localhost.localdomain

-N | --certNickname {nickname}

Nickname of the certificate that should be sent to the server for SSL client authentication.

-o | --saslOption {name=value}

SASL bind options.

-p | --port {port}

Directory server port number.

-q | --useStartTls

Use StartTLS to secure communication with the server.

Default: false

-T | --trustStorePassword {trustStorePassword}

Truststore cleartext password.

--useJavaKeyStore {keyStorePath}

JKS keystore containing the certificate which should be used for SSL client authentication.

--useJavaTrustStore {trustStorePath}

Use a JKS truststore file for validating server certificate.

--useJceKeyStore {keyStorePath}

JCEKS keystore containing the certificate which should be used for SSL client authentication.

--useJceTrustStore {trustStorePath}

Use a JCEKS truststore file for validating server certificate.

--useJvmTrustStore

Use the a JVM truststore for validating server certificate.

Default: false

--usePasswordPolicyControl

Use the password policy request control.

Default: false

--usePkcs11KeyStore

PKCS#11 keystore containing the certificate which should be used for SSL client authentication.

Default: false

--usePkcs12KeyStore {keyStorePath}

PKCS#12 keystore containing the certificate which should be used for SSL client authentication.

--usePkcs12TrustStore {trustStorePath}

Use a PKCS#12 truststore file for validating server certificate.

-w | --bindPassword {bindPassword}

Password to use to bind to the server. Omit this option while providing the bind DN to ensure that the command prompts for the password, rather than entering the password as a command argument.

-W | --keyStorePassword {keyStorePassword}

Keystore cleartext password.

-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:

--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.

`-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 \
 --port 1389 \
 --bindDn uid=kvaughan,ou=people,dc=example,dc=com \
 --bindPassword 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 --hostname $HOST --port $PORT --bindDn $ADMIN --bindPassword $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 \
 --port 1389 \
 --bindDn uid=kvaughan,ou=people,dc=example,dc=com \
 --bindPassword 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 \
 --port 1389 \
 --bindDn uid=kvaughan,ou=people,dc=example,dc=com \
 --bindPassword 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 \
 --port 1389 \
 --bindDn uid=kvaughan,ou=people,dc=example,dc=com \
 --bindPassword 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
```

Chapter 17

ldappasswordmodify — perform LDAP password modifications

Synopsis

ldappasswordmodify {options}

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.

`-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 values are not case-sensitive:

`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 DS server 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

`PwdPolicy`
`PasswordPolicy`

Password Policy Request Control, Object Identifier: 1.3.6.1.4.1.42.2.27.8.5.1

`PasswordQualityAdvice`

Password Quality Advice Request Control, Object Identifier: 1.3.6.1.4.1.36733.2.1.5.5

`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

`RelaxRules`

Relax Rules Request Control, Object Identifier: 1.3.6.1.4.1.4203.666.5.12

`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.

`-Y | --proxyAs {authzID}`

Use the proxied authorization control with the given authorization ID.

LDAP 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:

`-E | --reportAuthzId`

Use the authorization identity control.

Default: false

`-h | --hostname {host}`

Fully-qualified server host name or IP address.

Default: localhost.localdomain

-N | --certNickname {nickname}

Nickname of the certificate that should be sent to the server for SSL client authentication.

-o | --saslOption {name=value}

SASL bind options.

-p | --port {port}

Directory server port number.

-q | --useStartTls

Use StartTLS to secure communication with the server.

Default: false

-T | --trustStorePassword {trustStorePassword}

Truststore cleartext password.

--useJavaKeyStore {keyStorePath}

JKS keystore containing the certificate which should be used for SSL client authentication.

--useJavaTrustStore {trustStorePath}

Use a JKS truststore file for validating server certificate.

--useJceKeyStore {keyStorePath}

JCEKS keystore containing the certificate which should be used for SSL client authentication.

--useJceTrustStore {trustStorePath}

Use a JCEKS truststore file for validating server certificate.

--useJvmTrustStore

Use the a JVM truststore for validating server certificate.

Default: false

--usePasswordPolicyControl

Use the password policy request control.

Default: false

--usePkcs11KeyStore

PKCS#11 keystore containing the certificate which should be used for SSL client authentication.

Default: false

`--usePkcs12KeyStore {keyStorePath}`

PKCS#12 keystore containing the certificate which should be used for SSL client authentication.

`--usePkcs12TrustStore {trustStorePath}`

Use a PKCS#12 truststore file for validating server certificate.

`-w | --bindPassword {bindPassword}`

Password to use to bind to the server. Omit this option while providing the bind DN to ensure that the command prompts for the password, rather than entering the password as a command argument.

`-W | --keyStorePassword {keyStorePassword}`

Keystore cleartext password.

`-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:

`--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.

-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 \
 --port 1389 \
 --currentPasswordFile /tmp/currpwd.txt \
 --newPasswordFile /tmp/newpwd.txt \
 --bindDn uid=kvaughan,ou=people,dc=example,dc=com \
 --bindPassword bribery
The LDAP password modify operation was successful
```

Chapter 18

ldapsearch — perform LDAP search operations

Synopsis

ldapsearch {options} 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.

`--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 values are not case-sensitive:

`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 DS server 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

`PwdPolicy`
`PasswordPolicy`

Password Policy Request Control, Object Identifier: 1.3.6.1.4.1.42.2.27.8.5.1

`PasswordQualityAdvice`

Password Quality Advice Request Control, Object Identifier: 1.3.6.1.4.1.36733.2.1.5.5

`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

`RelaxRules`

Relax Rules Request Control, Object Identifier: 1.3.6.1.4.1.4203.666.5.12

`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 and do not contact the server.

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}

Use the server side sort control to have the server sort the results using the provided sort order. You can provide multiple comma separated sort keys. Sort key must respect the following pattern: "[-] attributeType [:OrderingRuleNameOrOID]". Minus character represent a descending 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:

`--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:

`-E | --reportAuthzId`

Use the authorization identity control.

Default: false

`-h | --hostname {host}`

Fully-qualified server host name or IP address.

Default: localhost.localdomain

`-N | --certNickname {nickname}`

Nickname of the certificate that should be sent to the server for SSL client authentication.

`-o | --saslOption {name=value}`

SASL bind options.

`-p | --port {port}`

Directory server port number.

`-q | --useStartTls`

Use StartTLS to secure communication with the server.

Default: false

`-T | --trustStorePassword {trustStorePassword}`

Truststore cleartext password.

--useJavaKeyStore {keyStorePath}

JKS keystore containing the certificate which should be used for SSL client authentication.

--useJavaTrustStore {trustStorePath}

Use a JKS truststore file for validating server certificate.

--useJceKeyStore {keyStorePath}

JCEKS keystore containing the certificate which should be used for SSL client authentication.

--useJceTrustStore {trustStorePath}

Use a JCEKS truststore file for validating server certificate.

--useJvmTrustStore

Use the a JVM truststore for validating server certificate.

Default: false

--usePasswordPolicyControl

Use the password policy request control.

Default: false

--usePkcs11KeyStore

PKCS#11 keystore containing the certificate which should be used for SSL client authentication.

Default: false

--usePkcs12KeyStore {keyStorePath}

PKCS#12 keystore containing the certificate which should be used for SSL client authentication.

--usePkcs12TrustStore {trustStorePath}

Use a PKCS#12 truststore file for validating server certificate.

-w | --bindPassword {bindPassword}

Password to use to bind to the server. Omit this option while providing the bind DN to ensure that the command prompts for the password, rather than entering the password as a command argument.

-W | --keyStorePassword {keyStorePassword}

Keystore cleartext password.

-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:

--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.

-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 DNs and UIDs:

```

$ ldapsearch \
 --port 1389 \
 --baseDn 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 \
 --port 1389 \
 --baseDn 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 \
 --port 1389 \
 --baseDn 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
```

Chapter 19

ldifdiff — compare small LDIF files

Synopsis

ldifdiff {options} 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

`-x | --exactMatch`

Match values byte-for-byte instead of using equality matching rules, which can be useful when comparing schema files.

Default: false

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 /path/to/newuser.ldif /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.
```

Chapter 20

ldifmodify — apply LDIF changes to LDIF

Synopsis

ldifmodify {options} 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 --outputLdif 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.
```

Chapter 21

ldifsearch — search LDIF with LDAP filters

Synopsis

ldifsearch {options} 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 --baseDn 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
```

Chapter 22

makeldif — generate test LDIF

Synopsis

makeldif {options} 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, such as `example.template`, or `people_and_groups.template`.

The following default template and data files are provided:

`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. To always generate the same data with the same command, use the same non-zero seed value. A value of zero (the default) results in different data each time the tool is run.

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 --outputLdif ../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 — template file for the makeldif command*"

Chapter 23

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:

<DateTime>

The DateTime tag is replaced by a timestamp.

The DateTime tag takes the form `<DateTime[:offsetInSeconds[:formatString]]>`, where:

- *offsetInSeconds* is the offset in seconds from the current time.

  The offset may be a positive or negative integer.

  Default: `0` (seconds).

- *formatString* is a date time pattern string. For details, see the Javadoc for the `DateTimeFormat` class.

  Default: `yyyyMMddHHmmss.SSS'Z'`.

<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 — generate test LDIF*", the server template file `config/MakeLDIF/example.template`

Chapter 24

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:

`--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: uid=admin

`-E | --reportAuthzId`

Use the authorization identity control.

Default: false

`-h | --hostname {host}`

Fully-qualified server host name or IP address.

Default: localhost.localdomain

`-N | --certNickname {nickname}`

Nickname of the certificate that should be sent to the server for SSL client authentication.

`-o | --saslOption {name=value}`

SASL bind options.

`-p | --port {port}`

Directory server administration port number.

`-T | --trustStorePassword {trustStorePassword}`

Truststore cleartext password.

`--useJavaKeyStore {keyStorePath}`

JKS keystore containing the certificate which should be used for SSL client authentication.

`--useJavaTrustStore {trustStorePath}`

Use a JKS truststore file for validating server certificate.

`--useJceKeyStore {keyStorePath}`

JCEKS keystore containing the certificate which should be used for SSL client authentication.

`--useJceTrustStore {trustStorePath}`

Use a JCEKS truststore file for validating server certificate.

`--useJvmTrustStore`

Use the a JVM truststore for validating server certificate.

Default: false

`--usePasswordPolicyControl`

Use the password policy request control.

Default: false

`--usePkcs11KeyStore`

PKCS#11 keystore containing the certificate which should be used for SSL client authentication.

Default: false

`--usePkcs12KeyStore {keyStorePath}`

PKCS#12 keystore containing the certificate which should be used for SSL client authentication.

`--usePkcs12TrustStore {trustStorePath}`

Use a PKCS#12 truststore file for validating server certificate.

`-w | --bindPassword {bindPassword}`

Password to use to bind to the server. Omit this option while providing the bind DN to ensure that the command prompts for the password, rather than entering the password as a command argument.

`-W | --keyStorePassword {keyStorePassword}`

Keystore cleartext password.

`-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

`-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 add-authentication-failure-time

Add an authentication failure time to the user account. This should be used only for testing purposes.

Options

The manage-account add-authentication-failure-time command takes the following options:

`-O | --operationValue {time}`

> A timestamp value using the generalized time syntax. Multiple timestamp values may be given by providing this argument multiple times.

manage-account add-grace-login-use-time

Add a grace login use time to the user account. This should be used only for testing purposes.

Options

The manage-account add-grace-login-use-time command takes the following options:

`-O | --operationValue {time}`

> A timestamp value using the generalized time syntax. Multiple timestamp values may be given by providing this argument multiple times.

manage-account clear-account-expiration-time

Clear account expiration time information from the user account.

manage-account clear-account-is-disabled

Clear account disabled state information from the user account.

manage-account clear-authentication-failure-times

Clear authentication failure time information from the user's account. This should be used only for testing purposes.

manage-account clear-grace-login-use-times

Clear the set of grace login use times for the user. This should be used only for testing purposes.

manage-account clear-last-login-time

Clear the time that the user last authenticated to the server. This should be used only for testing purposes.

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

Clear information about the required password change time with which the user last complied. This should be used only for testing purposes.

manage-account clear-password-changed-time

Clear information about the time that the user's password was last changed. This should be used only for testing purposes.

manage-account clear-password-expiration-warned-time

Clear information about the time that the user first received an expiration warning notice. This should be used only for testing purposes.

manage-account clear-password-history

Clear password history state values for the user. This should be used only for testing purposes.

manage-account clear-password-is-reset

Clear information about whether the user will be required to change his or her password on the next successful authentication. This should be used only for testing purposes.

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-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-expiration-time

Specify when the user account will expire.

Options

The manage-account set-account-expiration-time command takes the following options:

-O | --operationValue {time}

A timestamp value using the generalized time syntax.

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.

manage-account set-authentication-failure-times

Specify the authentication failure times for the user. This should be used only for testing purposes.

Options

The manage-account set-authentication-failure-times command takes the following options:

-O | --operationValue {time}

A timestamp value using the generalized time syntax. Multiple timestamp values may be given by providing this argument multiple times.

manage-account set-grace-login-use-times

Specify the grace login use times for the user. This should be used only for testing purposes.

Options

The manage-account set-grace-login-use-times command takes the following options:

-O | --operationValue {time}

A timestamp value using the generalized time syntax. Multiple timestamp values may be given by providing this argument multiple times.

manage-account set-last-login-time

Specify the time that the user last authenticated to the server. This should be used only for testing purposes.

Options

The manage-account set-last-login-time command takes the following options:

`-O | --operationValue {time}`

A timestamp value using the generalized time syntax.

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

Specify the required password change time with which the user last complied. This should be used only for testing purposes.

Options

The manage-account set-password-changed-by-required-time command takes the following options:

`-O | --operationValue {time}`

A timestamp value using the generalized time syntax.

manage-account set-password-changed-time

Specify the time that the user's password was last changed. This should be used only for testing purposes.

Options

The manage-account set-password-changed-time command takes the following options:

`-O | --operationValue {time}`

A timestamp value using the generalized time syntax.

manage-account set-password-expiration-warned-time

Specify the time that the user first received an expiration warning notice. This should be used only for testing purposes.

Options

The manage-account set-password-expiration-warned-time command takes the following options:

`-O | --operationValue {time}`

A timestamp value using the generalized time syntax.

manage-account set-password-is-reset

Specify whether the user will be required to change his or her password on the next successful authentication. This should be used only for testing purposes.

Options

The manage-account set-password-is-reset 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 administrator, 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 have all access"; allow(all)
  groupdn = "ldap:///cn=Directory Administrators,ou=Groups,dc=example,dc=com";)
```

The following command disables a user account:

```
$ manage-account \
 set-account-is-disabled \
 --port 4444 \
 --bindDn "uid=kvaughan,ou=people,dc=example,dc=com" \
 --bindPassword bribery \
 --operationValue true \
 --targetDn uid=bjensen,ou=people,dc=example,dc=com
 --trustAll
Account Is Disabled:  true
```

The following command enables a disabled user account:

```
$ manage-account \
 clear-account-is-disabled \
 --port 4444 \
 --bindDn "uid=kvaughan,ou=people,dc=example,dc=com" \
 --bindPassword bribery \
 --targetDn uid=bjensen,ou=people,dc=example,dc=com \
 --trustAll
Account Is Disabled:  false
```

Chapter 25

manage-tasks — manage server administration tasks

Synopsis

manage-tasks {options}

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.

`-i | --info {taskID}`

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

`-s | --summary`

Print a summary of tasks.

Default: false

`--status {taskStatus}`

Show only tasks with this status.

`-t | --type {taskType}`

Show only tasks of this type.

LDAP 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: uid=admin

`-E | --reportAuthzId`

Use the authorization identity control.

Default: false

`-h | --hostname {host}`

Fully-qualified server host name or IP address.

Default: localhost.localdomain

`-N | --certNickname {nickname}`

Nickname of the certificate that should be sent to the server for SSL client authentication.

`-o | --saslOption {name=value}`

SASL bind options.

`-p | --port {port}`

Directory server administration port number.

`-T | --trustStorePassword {trustStorePassword}`

Truststore cleartext password.

`--useJavaKeyStore {keyStorePath}`

JKS keystore containing the certificate which should be used for SSL client authentication.

`--useJavaTrustStore {trustStorePath}`

Use a JKS truststore file for validating server certificate.

--useJceKeyStore {keyStorePath}

JCEKS keystore containing the certificate which should be used for SSL client authentication.

--useJceTrustStore {trustStorePath}

Use a JCEKS truststore file for validating server certificate.

--useJvmTrustStore

Use the a JVM truststore for validating server certificate.

Default: false

--usePasswordPolicyControl

Use the password policy request control.

Default: false

--usePkcs11KeyStore

PKCS#11 keystore containing the certificate which should be used for SSL client authentication.

Default: false

--usePkcs12KeyStore {keyStorePath}

PKCS#12 keystore containing the certificate which should be used for SSL client authentication.

--usePkcs12TrustStore {trustStorePath}

Use a PKCS#12 truststore file for validating server certificate.

-w | --bindPassword {bindPassword}

Password to use to bind to the server. Omit this option while providing the bind DN to ensure that the command prompts for the password, rather than entering the password as a command argument.

-W | --keyStorePassword {keyStorePassword}

Keystore cleartext password.

-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 \
 --hostname opendj.example.com \
 --port 4444  \
 --bindDn uid=admin \
 --bindPassword password \
 --summary

  ID                                  Type    Status
  ---------------------------------------------------------------
  example-backup                      Backup  Recurring
  example-backup-<backupId>           Backup  Waiting on start time
```

Chapter 26

modrate — measure modification throughput and response time

Synopsis

modrate {options} [(attribute:value template 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 'uid=admin' -w password \

-F -c 4 -t 4 -b 'uid=user.{1},ou=people,dc=example,dc=com' \

-g 'rand(0,2000)' -g 'randstr(16)' 'description:{2}'

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 template string.

-B | --warmUpDuration {warmUpDuration}

Warm up duration in seconds.

Default: 0

-c | --numConnections {numConnections}

Number of connections.

Default: 1

-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 template strings in program parameters (ie. Base DN, Search Filter). The set of all arguments provided form 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 | --numConcurrentRequests {numConcurrentRequests}

Number of concurrent requests per connection.

Default: 1

-Y | --proxyAs {authzID}

Use the proxied authorization control with the given authorization ID.

LDAP 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:

`-E | --reportAuthzId`

Use the authorization identity control.

Default: false

`-h | --hostname {host}`

Fully-qualified server host name or IP address.

Default: localhost.localdomain

`-N | --certNickname {nickname}`

Nickname of the certificate that should be sent to the server for SSL client authentication.

`-o | --saslOption {name=value}`

SASL bind options.

`-p | --port {port}`

Directory server port number.

`-q | --useStartTls`

Use StartTLS to secure communication with the server.

Default: false

`-T | --trustStorePassword {trustStorePassword}`

Truststore cleartext password.

`--useJavaKeyStore {keyStorePath}`

JKS keystore containing the certificate which should be used for SSL client authentication.

`--useJavaTrustStore {trustStorePath}`

Use a JKS truststore file for validating server certificate.

`--useJceKeyStore {keyStorePath}`

JCEKS keystore containing the certificate which should be used for SSL client authentication.

`--useJceTrustStore {trustStorePath}`

Use a JCEKS truststore file for validating server certificate.

`--useJvmTrustStore`

Use the a JVM truststore for validating server certificate.

Default: false

`--usePasswordPolicyControl`

Use the password policy request control.

Default: false

`--usePkcs11KeyStore`

PKCS#11 keystore containing the certificate which should be used for SSL client authentication.

Default: false

`--usePkcs12KeyStore {keyStorePath}`

PKCS#12 keystore containing the certificate which should be used for SSL client authentication.

`--usePkcs12TrustStore {trustStorePath}`

Use a PKCS#12 truststore file for validating server certificate.

`-w | --bindPassword {bindPassword}`

Password to use to bind to the server. Omit this option while providing the bind DN to ensure that the command prompts for the password, rather than entering the password as a command argument.

`-W | --keyStorePassword {keyStorePassword}`

Keystore cleartext password.

`-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:

`-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.

`-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 uses the modrate command to write random 16-character description values to user entries:

```
$ modrate \
 --hostname localhost \
 --port 1389 \
 --bindDn uid=admin \
 --bindPassword password \
 --noRebind \
 --numConnections 4 \
 --numConcurrentRequests 4 \
 --argument "rand(0,2000)" --targetDn "uid=user.{1},ou=people,dc=example,dc=com" \
 --argument "randstr(16)" 'description:{2}'
-------------------------------------------------------------------------------
|     Throughput    |                 Response Time                |           |
|    (ops/second)   |                (milliseconds)                |           |
|   recent  average |   recent  average    99.9%   99.99%  99.999% |   err/sec |
-------------------------------------------------------------------------------
|  11616.6  11616.6 |    1.360    1.360    21.23   156.24   484.44 |       0.0 |
|  38501.4  25059.0 |    0.410    0.630    14.29   155.19   484.44 |       0.0 |
|  47660.4  32592.8 |    0.331    0.484    10.94    28.05   350.22 |       0.0 |
|  46837.2  36153.9 |    0.337    0.437     9.37    23.07   270.53 |       0.0 |
|  41042.0  37131.5 |    0.385    0.425     8.59    27.00   329.25 |       0.0 |
|  46397.0  38675.8 |    0.340    0.408     7.63    22.02   329.25 |       0.0 |
^C
```

This example uses the following options:

`--hostname localhost`
`--port 1389`

Access the server running on the local system on port 1389.

`--bindDn uid=admin`
`--bindPassword password`

Authenticate as the directory root user `uid=admin` with the bind password that is literally `password`.

This user is not subject to access control, so rates may be higher than what you observe with a regular user.

`--noRebind`

Keep connections open and do not rebind.

`--numConnections 4`

Open 4 connections to the server.

`--numConcurrentRequests 4`

Perform up to 4 concurrent requests on each connection.

`--argument "rand(0,2000)" --targetDn "uid=user.{1},ou=people,dc=example,dc=com"`

Target the entry with DN `uid=user.`*`number`*`,ou=people,dc=example,dc=com`, where *number* is a random number between 0 and 2000, inclusive.

`--argument "randstr(16)" 'description:{2}'`

Write a random, 16-character string to the `description` attribute of the target entry.

The `randstr(16)` argument specifies only the length, which is 16. It does not have an optional second argument to specify a character set. Therefore, use the default character set, which is `[A-Z][a-z][0-9]`.

Notice the following characteristics of the output:

- The first two columns show the throughput in operations completed per second.

  The recent column shows the average rate for operations reflected in this row of output.

  The average column shows the average rate since the beginning of the run.

- The response time columns indicate characteristics of response latency in milliseconds.

  The recent column shows the average latency for operations reflected in this row of output.

  The average column shows the average latency since the beginning of the run.

  The "99.9%" column shows the latency after which 99.9% of operations have completed. Only 1 operation in 1000 took longer than this.

  The "99.99%" column shows the latency after which 99.99% of operations have completed. Only 1 operation in 10,000 took longer than this.

  The "99.999%" column shows the latency after which 99.999% of operations have completed. Only 1 operation in 100,000 took longer than this.

- The "err/sec" column show the rate of error results per second for this row of output.

  Unless you have intentionally set up the command to generate errors, this column should indicate `0.0`. Check that this column matches your expectations before looking at any other columns.

Chapter 27

rebuild-index — rebuild index after configuration change

Synopsis

rebuild-index {options}

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. When using this option, the command writes to server files. Run the command as a user having the same filesystem permissions as the user running the server.

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 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.

`--description {description}`

Gives a description to the task.

`--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.

The schedule pattern for a recurring task supports only the following crontab features:

Time and Date Fields

Field	Allowed Values
minute	0-59
hour	0-23
day of month	1-31
month	1-12 (or names)
day of week	0-7 (0 or 7 is Sunday, or use names)

A field can contain an asterisk, `*`. An asterisk stands for *`first-last`*.

Fields can include ranges of numbers. A range is two numbers separated by a hyphen, and is inclusive. For example, `8-10` for an "hour" field means execution at hours 8, 9, and 10.

Fields can include lists. A list is a set of numbers or ranges separated by commas. For example, `4, 8-10` for an "hour" field means execution at hours 4, 8, 9, and 10.

When using names for in "month" or "day of week" fields, use the first three letters of the particular month or day of the week. Case does not matter. Ranges and lists of names are not supported.

`-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.

`--taskId {taskID}`

Gives an ID to the task.

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: uid=admin

`-E | --reportAuthzId`

Use the authorization identity control.

Default: false

`-h | --hostname {host}`

Fully-qualified server host name or IP address.

Default: localhost.localdomain

`-N | --certNickname {nickname}`

Nickname of the certificate that should be sent to the server for SSL client authentication.

`-o | --saslOption {name=value}`

SASL bind options.

`-p | --port {port}`

Directory server administration port number.

`-T | --trustStorePassword {trustStorePassword}`

Truststore cleartext password.

`--useJavaKeyStore {keyStorePath}`

JKS keystore containing the certificate which should be used for SSL client authentication.

`--useJavaTrustStore {trustStorePath}`

Use a JKS truststore file for validating server certificate.

`--useJceKeyStore {keyStorePath}`

JCEKS keystore containing the certificate which should be used for SSL client authentication.

`--useJceTrustStore {trustStorePath}`

Use a JCEKS truststore file for validating server certificate.

`--useJvmTrustStore`

Use the a JVM truststore for validating server certificate.

Default: false

`--usePasswordPolicyControl`

Use the password policy request control.

Default: false

`--usePkcs11KeyStore`

PKCS#11 keystore containing the certificate which should be used for SSL client authentication.

Default: false

`--usePkcs12KeyStore {keyStorePath}`

PKCS#12 keystore containing the certificate which should be used for SSL client authentication.

`--usePkcs12TrustStore {trustStorePath}`

Use a PKCS#12 truststore file for validating server certificate.

`-w | --bindPassword {bindPassword}`

Password to use to bind to the server. Omit this option while providing the bind DN to ensure that the command prompts for the password, rather than entering the password as a command argument.

`-W | --keyStorePassword {keyStorePassword}`

Keystore cleartext password.

`-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 schedules a task to start immediately that rebuilds the `cn` (common name) index:

```
$ rebuild-index \
 --hostname opendj.example.com \
 --port 4444 \
 --bindDn uid=admin \
 --bindPassword password \
 --baseDn dc=example,dc=com \
 --index cn \
 --start 0
Rebuild Index task <taskId> scheduled to start <date>
```

Chapter 28

searchrate — measure search throughput and response time

Synopsis

searchrate {options} [filter template 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 'uid=admin' -w password \

-F -c 4 -t 4 -b 'dc=example,dc=com' -g 'rand(0,2000)' '(uid=user.{})'

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 template string.

`-B | --warmUpDuration {warmUpDuration}`

Warm up duration in seconds.

Default: 0

`-c | --numConnections {numConnections}`

Number of connections.

Default: 1

`-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 template strings in program parameters (ie. Base DN, Search Filter). The set of all arguments provided form 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 | --numConcurrentRequests {numConcurrentRequests}`

Number of concurrent requests per connection.

Default: 1

`-Y | --proxyAs {authzID}`

Use the proxied authorization control with the given authorization ID.

LDAP 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:

`-E | --reportAuthzId`

Use the authorization identity control.

Default: false

`-h | --hostname {host}`

Fully-qualified server host name or IP address.

Default: localhost.localdomain

`-N | --certNickname {nickname}`

Nickname of the certificate that should be sent to the server for SSL client authentication.

`-o | --saslOption {name=value}`

SASL bind options.

`-p | --port {port}`

Directory server port number.

`-q | --useStartTls`

Use StartTLS to secure communication with the server.

Default: false

`-T | --trustStorePassword {trustStorePassword}`

Truststore cleartext password.

`--useJavaKeyStore {keyStorePath}`

JKS keystore containing the certificate which should be used for SSL client authentication.

`--useJavaTrustStore {trustStorePath}`

Use a JKS truststore file for validating server certificate.

`--useJceKeyStore {keyStorePath}`

JCEKS keystore containing the certificate which should be used for SSL client authentication.

`--useJceTrustStore {trustStorePath}`

Use a JCEKS truststore file for validating server certificate.

`--useJvmTrustStore`

Use the a JVM truststore for validating server certificate.

Default: false

`--usePasswordPolicyControl`

Use the password policy request control.

Default: false

`--usePkcs11KeyStore`

PKCS#11 keystore containing the certificate which should be used for SSL client authentication.

Default: false

`--usePkcs12KeyStore {keyStorePath}`

PKCS#12 keystore containing the certificate which should be used for SSL client authentication.

`--usePkcs12TrustStore {trustStorePath}`

Use a PKCS#12 truststore file for validating server certificate.

-w | --bindPassword {bindPassword}

Password to use to bind to the server. Omit this option while providing the bind DN to ensure that the command prompts for the password, rather than entering the password as a command argument.

-W | --keyStorePassword {keyStorePassword}

Keystore cleartext password.

-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:

-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.

-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 measures search performance:

```
$ searchrate \
 --hostname localhost \
 --port 1389 \
 --baseDn dc=example,dc=com \
 --numConnections 4 \
 --numConcurrentRequests 4 \
 --argument "rand(0,2000)" "(uid=user.{})"
--------------------------------------------------------------------------------------------
|     Throughput    |                 Response Time                  |       Additional      |
|    (ops/second)   |                (milliseconds)                  |       Statistics      |
|   recent  average |   recent  average    99.9%   99.99%  99.999% |  err/sec Entries/Srch |
--------------------------------------------------------------------------------------------
|  38515.0  38515.0 |    0.410    0.410     9.57    19.66    26.61 |      0.0          1.0 |
|  47742.4  43128.7 |    0.332    0.367     7.18    15.93    25.56 |      0.0          1.0 |
|  48027.6  44761.7 |    0.330    0.353     6.26    14.55    23.07 |      0.0          1.0 |
|  47773.6  45514.7 |    0.331    0.348     5.80    13.30    22.81 |      0.0          1.0 |
|  47833.8  45978.5 |    0.331    0.344     5.34    12.32    22.02 |      0.0          1.0 |
|  47891.2  46297.3 |    0.331    0.342     4.98    11.99    21.23 |      0.0          1.0 |
|  46579.8  46337.6 |    0.340    0.341     4.82    11.80    20.97 |      0.0          1.0 |
^C
```

This example uses the following options:

--hostname localhost
--port 1389

Access the server running on the local system on port 1389.

`--baseDn dc=example,dc=com`

Search under the base DN `dc=example,dc=com`.

This user is not subject to access control, so rates may be higher than what you observe with a regular user.

No `--bindDn` or `--bindPassword` options

Perform the search as an anonymous user.

`--noRebind`

Keep connections open and do not rebind.

`--numConnections 4`

Open 4 connections to the server.

`--numConcurrentRequests 4`

Perform up to 4 concurrent requests on each connection.

`--argument "rand(0,2000)" "(uid=user.{})"`

Search for an entry with UID equal to `uid=user.number`, where *number* is a random number between 0 and 2000, inclusive.

Notice the following characteristics of the output:

- The first two columns show the throughput in operations completed per second.

  The recent column shows the average rate for operations reflected in this row of output.

  The average column shows the average rate since the beginning of the run.

- The response time columns indicate characteristics of response latency in milliseconds.

  The recent column shows the average latency for operations reflected in this row of output.

  The average column shows the average latency since the beginning of the run.

  The "99.9%" column shows the latency after which 99.9% of operations have completed. Only 1 operation in 1000 took longer than this.

  The "99.99%" column shows the latency after which 99.99% of operations have completed. Only 1 operation in 10,000 took longer than this.

  The "99.999%" column shows the latency after which 99.999% of operations have completed. Only 1 operation in 100,000 took longer than this.

- The additional statistics columns show information about what is happening during the run.

The "err/sec" column shows the rate of error results per second for this row of output. Unless you have intentionally set up the command to generate errors, this column should indicate `0.0`.

The "Entries/Srch" column shows the average number of entries returned for each search. If you expect one result entry per search, this column should indicate `1.0`.

Check that these columns match your expectations before looking at any other columns.

Chapter 29
setup — install OpenDJ server

Synopsis

setup {options}

Description

This utility sets up an OpenDJ server. Use the --help-profiles option to list available profiles.

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.

`--bootstrapReplicationServer {bootstrapReplicationServer}`

The addresses of one or more replication servers within the topology which the server should connect to for discovering the rest of the topology. Use syntax "hostname:port" or "[IPv6Address]:port" for IPv6 addresses.

`-D | --rootUserDn {rootUserDN}`

DN for the initial root user for the Directory Server.

Default: uid=admin

--deploymentKey {deploymentKey}

The deployment key which should be used for securing the deployment. If this option is not provided then a new deployment key will be generated automatically and displayed so that it can be re-used for subsequent servers in the deployment. If no existing certificates are specified using the key-store and trust-store options then the deployment key will also be used for securing all TLS network communication.

--deploymentKeyPassword {deploymentKeyPassword}

Deployment key password. The value is used as a new deployment key password or the password associated to an existing deployment key depending on whether the --deploymentKey is also used.

-h | --hostname {host}

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

--help-profile {name[:version]}

Display profile parameters.

--help-profiles

Display all available profiles.

Default: false

--httpPort {port}

Port on which the server should listen for HTTP communication.

--httpsPort {port}

Port on which the server should listen for HTTPS communication.

--instancePath {path}

Path were the instance should be set up.

Default: /tmp

--monitorUserDn {monitorUserDn}

DN of the default user allowed to query monitoring information.

Default: uid=Monitor

--monitorUserPassword {monitorUserPassword}

Password of the default user allowed to query monitoring information.

-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.

-p | --ldapPort {port}

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

--profile {name[:version]}

Setup profile to apply when initially configuring the server. If the version is not specified, the most recent version older or equal to this OpenDJ version is used. Use this option multiple times to apply multiple profiles. This option cannot be combined with data import options. There are no setup profiles available for this OpenDJ version.

-q | --enableStartTls

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

Default: false

-Q | --quiet

Use quiet mode.

Default: false

-r | --replicationPort {port}

Port used for replication protocol communications with other servers. Use this option to configure a local replication server. When this option is not used, this server is configured as a standalone DS (no local replication server).

-s | --start

Start the server when the configuration is completed.

Default: false

-S | --skipPortCheck

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

Default: false

--serverId {serverId}

Specify the server ID for this server. An acceptable ID is an ASCII alpha-numeric string; it may also contain underscore and hyphen characters provided they are not the first character.

--set {[profileName/]parameterName:value}

Assign a value to a setup profile parameter. Profile name must be provided if multiple profiles are provided, indicate the profile that a parameter applies to by using the profileName/ parameterName format.

-T | --trustStorePassword {trustStorePassword}

Truststore cleartext password.

--useJavaKeyStore {keyStorePath}

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

--useJavaTrustStore {trustStorePath}

Use existing JKS truststore file for validating peer SSL certificates.

--useJceKeyStore {keyStorePath}

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

--useJceTrustStore {trustStorePath}

Use existing JCEKS truststore file for validating peer SSL certificates.

--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.

--usePkcs12TrustStore {trustStorePath}

Use existing PKCS12 truststore file for validating peer SSL certificates.

-w | --rootUserPassword {rootUserPassword}

Password for the initial root user for the Directory Server.

-W | --keyStorePassword {keyStorePassword}

Keystore cleartext password.

-X | --trustAll

Blindly trust peer SSL certificates.

Default: false

-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 option is explicitly specified.

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 installs a directory server, enables StartTLS and imports 100 example entries:

```
$ /path/to/opendj/setup \
 --deploymentKey AFPxL0RlmdMZHeVkkcC3GYFsAHNlNQ5CBVN1bkVDM7FyW2gWxnvQdQ \
 --deploymentKeyPassword password \
 --rootUserDn uid=admin \
 --rootUserPassword password \
 --hostname opendj.example.com \
 --adminConnectorPort 4444 \
 --ldapPort 1389 \
 --enableStartTls \
 --profile ds-evaluation \
 --set ds-evaluation/generatedUsers:100 \
 --acceptLicense


Validating parameters..... Done
Configuring certificates..... Done
Configuring server..... Done
Configuring profile DS evaluation......... Done
Starting directory server.......... Done

To see basic server status and configuration, you can launch
/path/to/opendj/bin/status
```

Chapter 30

setup-profile — configure profiles in an offline OpenDJ server instance

Synopsis

setup-profile {options}

Description

This utility configures profiles in an offline OpenDJ server instance. There are no setup profiles available for this OpenDJ version

.

Options

The setup-profile command takes the following options:

Command options:

`--help-profile {name[:version]}`

Display profile parameters.

`--instancePath {path}`

Path of the server instance where profiles should be setup.

Default: /mnt/scratch/workspaces/workspace/ds-release_sustaining_7.0.x

`--profile {name[:version]}`

Name of the profile to be configured. If the version is not specified, the most recent version older or equal to this OpenDJ version is used. Use this option multiple times to apply multiple profiles.

`--set {[profileName/]parameterName:value}`

Assign a value to a setup profile parameter. Profile name must be provided if multiple profiles are provided, indicate the profile that a parameter applies to by using the profileName/ parameterName format.

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 command setup AM identity and configuration store profiles:

```
$ stop-ds
$ setup-profile \
--instancePath /path/to/opendj \
--profile am-identity-store \
--set am-identity-store/amIdentityStoreAdminPassword:password \
--profile am-config \
--set am-config/amConfigAdminPassword:password
Configuring profile AM identity data store..... Done
  Configuring profile AM configuration data store...... Done
$ start-ds
```

Chapter 31

start-ds — start OpenDJ server

Synopsis

start-ds {options}

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 --quiet
```

Chapter 32 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:

`--offline`

Indicates that the command must be run in offline mode.

Default: false

LDAP 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: uid=admin

`-E | --reportAuthzId`

Use the authorization identity control.

Default: false

`-h | --hostname {host}`

Fully-qualified server host name or IP address.

Default: localhost.localdomain

`-N | --certNickname {nickname}`

Nickname of the certificate that should be sent to the server for SSL client authentication.

`-o | --saslOption {name=value}`

SASL bind options.

`-p | --port {port}`

Directory server administration port number.

`-T | --trustStorePassword {trustStorePassword}`

Truststore cleartext password.

`--useJavaKeyStore {keyStorePath}`

JKS keystore containing the certificate which should be used for SSL client authentication.

`--useJavaTrustStore {trustStorePath}`

Use a JKS truststore file for validating server certificate.

`--useJceKeyStore {keyStorePath}`

JCEKS keystore containing the certificate which should be used for SSL client authentication.

`--useJceTrustStore {trustStorePath}`

Use a JCEKS truststore file for validating server certificate.

`--useJvmTrustStore`

Use the a JVM truststore for validating server certificate.

Default: false

`--usePasswordPolicyControl`

Use the password policy request control.

Default: false

`--usePkcs11KeyStore`

PKCS#11 keystore containing the certificate which should be used for SSL client authentication.

Default: false

`--usePkcs12KeyStore {keyStorePath}`

PKCS#12 keystore containing the certificate which should be used for SSL client authentication.

`--usePkcs12TrustStore {trustStorePath}`

Use a PKCS#12 truststore file for validating server certificate.

`-w | --bindPassword {bindPassword}`

Password to use to bind to the server. Omit this option while providing the bind DN to ensure that the command prompts for the password, rather than entering the password as a command argument.

`-W | --keyStorePassword {keyStorePassword}`

Keystore cleartext password.

`-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

The following command displays the status of a running directory server:

```
$ status \
 --hostname opendj.example.com \
 --port 4444 \
 --bindDn uid=admin \
 --bindPassword password \
 --trustAll
>>>> General details

Version                        : ForgeRock Directory Services 7.0.2
```

```
Installation and instance path : /path/to/opendj
Run status                     : Started
Host name                      : <fqdn>
Administration port (LDAPS)    : 4444
Open connections               : 1


>>>> Running server Java details

Java version       : <version>
Java vendor        : <vendor>
JVM available CPUs : <cores>
JVM max heap size  : <size>


>>>> Connection handlers

Name  : Port : Protocol : Security  : Status   : Load m1 rate : Load m5 rate
------:------:----------:-----------:----------:--------------:-------------
HTTP  : 8080 : HTTP     : Unsecured : Enabled  : 0.0          : 0.0
HTTPS : 8443 : HTTP     : SSL       : Enabled  : 0.0          : 0.0
LDAP  : 1389 : LDAP     : Unsecured : Enabled  : 0.0          : 0.0
LDAPS : 1636 : LDAP     : SSL       : Enabled  : 0.0          : 0.0
LDIF  : -    : LDIF     : -         : Disabled : -            : -
SNMP  : 161  : SNMP     : -         : Disabled : -            : -


>>>> Local backends

Base DN             : Entries : Replication : Backend     : Type                             : Status
--------------------:---------:-------------:-------------:----------------------------------:--------
uid=Monitor         : 1       : -           : monitorUser : LDIF                             : Enabled
uid=admin           : 1       : -           : rootUser    : LDIF                             : Enabled
dc=example,dc=com   : 180     : -           : userData    : DB (<size> active cache size)    : Enabled


>>>> Disk space

Disk space : State  : Free space
-----------:--------:-----------
/          : normal : <size>
```

Chapter 33

stop-ds — stop OpenDJ server

Synopsis

stop-ds {options}

Description

This utility can be used to request that the Directory Server stop running or perform a restart. When run without explicit connection options, this utility sends a signal to the OpenDJ process to stop the server. When run with explicit 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:

--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: uid=admin

-E | --reportAuthzId

Use the authorization identity control.

Default: false

-h | --hostname {host}

Fully-qualified server host name or IP address.

Default: localhost.localdomain

-N | --certNickname {nickname}

Nickname of the certificate that should be sent to the server for SSL client authentication.

-o | --saslOption {name=value}

SASL bind options.

-p | --port {port}

Directory server administration port number.

-T | --trustStorePassword {trustStorePassword}

Truststore cleartext password.

--useJavaKeyStore {keyStorePath}

JKS keystore containing the certificate which should be used for SSL client authentication.

--useJavaTrustStore {trustStorePath}

Use a JKS truststore file for validating server certificate.

`--useJceKeyStore {keyStorePath}`

JCEKS keystore containing the certificate which should be used for SSL client authentication.

`--useJceTrustStore {trustStorePath}`

Use a JCEKS truststore file for validating server certificate.

`--useJvmTrustStore`

Use the a JVM truststore for validating server certificate.

Default: false

`--usePasswordPolicyControl`

Use the password policy request control.

Default: false

`--usePkcs11KeyStore`

PKCS#11 keystore containing the certificate which should be used for SSL client authentication.

Default: false

`--usePkcs12KeyStore {keyStorePath}`

PKCS#12 keystore containing the certificate which should be used for SSL client authentication.

`--usePkcs12TrustStore {trustStorePath}`

Use a PKCS#12 truststore file for validating server certificate.

`-w | --bindPassword {bindPassword}`

Password to use to bind to the server. Omit this option while providing the bind DN to ensure that the command prompts for the password, rather than entering the password as a command argument.

`-W | --keyStorePassword {keyStorePassword}`

Keystore cleartext password.

`-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

Exit Codes

0

The command completed successfully.

> 0

An error occurred.

Examples

The following example restarts a server:

```
$ stop-ds --restart
Stopping Server...
...The Directory Server has started successfully
```

Chapter 34

supportextract — extract support data

Synopsis

supportextract {options}

Description

This tool collects support data from the OpenDJ instance it is bound to.

Options

The supportextract command takes the following options:

Command options:

`-d | --outputDirectory {directory}`

The folder into which the files will be placed into.

`--logsAfterDate {date}`

Collect log files after this date. Format "YYYYMMDDhhmmss" like "20161123143612" = 23 November 2016, 14:36 12s. Overrides --maxLogFiles.

`--maxLogFiles {number}`

Maximum number of log files to collect. Ignored if --logsAfterDate is provided.

Default: 100

`--needJavaHeapDump`

Specifies whether a Java Heap Dump (using jmap) should be produced. The binary file is generated at the same location as the ZIP archive before being added to it; please make sure that the target directory's volume has sufficient capacity.

Default: false

--noAuditFiles

Specifies whether audit files are excluded.

Default: false

--noKeystoreFiles

Specifies whether keystore files are excluded.

Default: false

--noServerInteraction

Specifies that the tool should not interact with the server, that is no LDAP operation, and no jstack sampling.

Default: false

--serverPID {pid}

When the server is embedded in OpenAM, there is no PID file. Therefore this option indicates the server PID of the OpenAM application server.

-t | --jdkToolsDirectory {directory}

Path to the JDK utility binaries directory such as jstack.

Default: /opt/jdk-11.0.4+11/bin

LDAP connection options:

-D | --bindDn {bindDN}

DN to use to bind to the server.

Default:

-w | --bindPassword {password}

Password to use to bind to the server.

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 creates a support archive in a custom directory:

```
$ supportextract \
 --bindDn uid=admin \
 --bindPassword password \
 --outputDirectory /path/to/output/directory

The instance is running
No value was provided for --jdkToolsDirectory, JDK tool directory is set to
</path/to/jdk/bin>
VERSION: <version>
/path/to/output/directory/data/dev/opendj-support-data-<timestamp>.zip.lock
Collecting the monitoring info from cn=monitor
Collecting process statistics
Cannot extract process statistics (by running "top" command) on OS '<OS>'.
Only jcmd dump samples will be collected
- Generating stack dump, sample number : 1 using jcmd for pid <pid>
- Generating stack dump, sample number : 2 using jcmd for pid <pid>
- Generating stack dump, sample number : 3 using jcmd for pid <pid>
- Generating stack dump, sample number : 4 using jcmd for pid <pid>
- Generating stack dump, sample number : 5 using jcmd for pid <pid>
- Generating stack dump, sample number : 6 using jcmd for pid <pid>
- Generating stack dump, sample number : 7 using jcmd for pid <pid>
- Generating stack dump, sample number : 8 using jcmd for pid <pid>
- Generating stack dump, sample number : 9 using jcmd for pid <pid>
- Generating stack dump, sample number : 10 using jcmd for pid <pid>
Collecting the configuration files
- Adding rootUser.ldif
- Adding monitorUser.ldif
- Adding schema files
- Adding HTTP configuration file(s)
- Listing the security stores
* config/keystore
Collecting system node information
- OS information
- Network information
- Disk information
- Processor information
```

```
Collecting ChangelogDb information
- No changelogDb data found (is a DS or is not replicated)
Collecting backend statistics
- amCts: total jdb files 1
- Adding je.info.0
- Adding je.config.csv
- Adding je.stat.csv
Collecting the log files
- /path/to/output/directory/logs/access
- /path/to/output/directory/logs/filtered-ldap-access.audit.json
- /path/to/output/directory/logs/ldap-access.audit.json
- /path/to/output/directory/logs/errors
- /path/to/output/directory/logs/replication
- /path/to/output/directory/logs/server.out
Collecting the GC log files
- /path/to/output/directory/logs/cust11.log.0
- /path/to/output/directory/logs/cust11.log

The following archive has been created :
/path/to/output/directory/data/dev/opendj-support-data-<timestamp>.zip
```

Chapter 35

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 performs only part of the upgrade process, which includes the following phases for a single server:

1. Get and unpack a newer version of the software.
2. Stop the current server.
3. Overwrite existing binary and script files with those of the newer version, and then run this utility before restarting the server.
4. Start the upgraded server.

> **Important**
>
> This utility *does not back up your data before you upgrade, nor does it restore your data if the utility fails*. In order to revert a failed upgrade, make sure you back up directory data 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

`--dataOnly`

Upgrades only application data. OpenDJ configuration must have been upgraded before.

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 performed automatically. 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.

Chapter 36

verify-index — check index for consistency or errors

Synopsis

verify-index {options}

Description

This utility ensures that index data is consistent within an indexed backend database. Stop the server before running this tool.

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:

```
$ verify-index --baseDn dc=example,dc=com --index sn --clean --countErrors
... msg=Maximum number of entries referenced by any record is 32
```

Chapter 37

windows-service — register DS as a Windows Service

Synopsis

windows-service options

Description

This utility can be used to run the 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 the 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.