IBM Security Directory Suite
8.0.1

Command Reference

Note

Before using this information and the product it supports, read the general information under "Notices" on page 143.

Edition notice

Note: This edition applies to version 8.0.1.x of *IBM Security Directory Suite* (product number 5725-Y17) and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

About this publication

IBM® Security Directory Suite, previously known as IBM Security Directory Server or IBM Tivoli® Directory Server, is an IBM implementation of the Lightweight Directory Access Protocol.

IBM Security Directory Suite Command Reference describes the syntax and usage of the command-line utilities included with IBM Security Directory Suite.

Accessibility

Accessibility features help users with a physical disability, such as restricted mobility or limited vision, to use software products successfully. With this product, you can use assistive technologies to hear and navigate the interface. You can also use the keyboard instead of the mouse to operate all features of the graphical user interface.

For more information, see "Accessibility features for IBM Security Directory Suite" in the IBM Knowledge Center.

Statement of Good Security Practices

IT system security involves protecting systems and information through prevention, detection, and response to improper access from within and outside your enterprise. Improper access can result in information being altered, destroyed, misappropriated, or misused or can result in damage to or misuse of your systems, including for use in attacks on others. No IT system or product should be considered completely secure and no single product, service or security measure can be completely effective in preventing improper use or access. IBM systems, products and services are designed to be part of a comprehensive security approach, which will necessarily involve additional operational procedures, and may require other systems, products or services to be most effective. IBM DOES NOT WARRANT THAT ANY SYSTEMS, PRODUCTS OR SERVICES ARE IMMUNE FROM, OR WILL MAKE YOUR ENTERPRISE IMMUNE FROM, THE MALICIOUS OR ILLEGAL CONDUCT OF ANY PARTY.

Chapter 1. IBM Security Directory Suite command reference

Use the virtual appliance command-line interface to run IBM Security Directory Suite commands.

1. Access the command-line interface (CLI) of the virtual appliance by using either an `ssh` session or the console.
2. From the command-line interface, log on to the IBM Security Directory Suite virtual appliance. The following message is displayed:

```
Welcome to the IBM Security Directory Suite appliance
Enter "help" for a list of available commands
```

3. Enter `sds` to use the IBM Security Directory Suite commands.

```
sdsva.example.com > sds
```

4. To see a list of available commands, enter the **help** command at the command-line prompt. The **help** command provides detailed information about each command from the list.

```
sdsva.example.com:sds> help
```

For the detailed syntax, see the links that are provided next to the description of each command.

Note: In a IBM Security Directory Suite virtual appliance environment where only one Directory Server instance is present, specifying the instance name is optional. To view the name of the existing instance in the virtual appliance use the **idsilist** command.

The IBM Security Directory Suite commands are categorized into the following folders:

```
client_tools        SDS client utilities.
firmware_update     Work with the SDS Appliance firmware settings.
migration_tools     Migration utilities.
server_tools        SDS server instance utilites.
```

Client tools

Use the client utilities under the `client_tools` folder to run operations on IBM Security Directory Suite from the virtual appliance command-line interface.

deleteTDISysStore

Deletes the system store for Directory Integrator based solutions.

For parameters and usage information, see "deleteTDISysStore" on page 3

deletefile

Deletes a file or directory from specific virtual appliance base directories.

For parameters and usage information, see "deletefile" on page 3.

idsdirctl

Starts or stops Directory Server instance.

For parameters and usage information, see "idsdirctl" on page 4.

idsgetfile

Uploads a file from the specified location on a remote system to the `/userdata/directory/CustomIn` folder, which can be viewed on the Custom File Management page in the virtual appliance console.

For parameters and usage information, see "idsgetfile" on page 6.

idsldapadd

Adds entries in Directory Server.

For parameters and usage information, see "idsldapadd, idsldapmodify" on page 6.

idsldapchangepwd

Sends modify password requests to an LDAP server.

For parameters and usage information, see "idsldapchangepwd" on page 12.

idsldapcompare

Compares an attribute value of an entry in Directory Server with your compare criteria.

For parameters and usage information, see "idsldapcompare" on page 15.

idsldapdelete

Deletes one or more entries from the directory information tree (DIT).

For parameters and usage information, see "idsldapdelete" on page 16.

idsldapdiff

Identifies the differences in a replica server and its master server. You can also use this command to synchronize the replica server with its master server.

For parameters and usage information, see "idsldapdiff" on page 20.

idsldapexop

Runs LDAP extended operations.

For parameters and usage information, see "idsldapexop" on page 27.

idsldapmodify

Modifies or adds entries in Directory Server.

For parameters and usage information, see "idsldapadd, idsldapmodify" on page 6.

idsldapmodrdn

Modifies the relative distinguished name (RDN) or changes the parent DN of an entry.

For parameters and usage information, see "idsldapmodrdn" on page 38

idsldapreplcfg

Configures various replication topologies for the Directory Server. This tool simplifies the process of setting up replication and reduces errors that might occur when you set up replication by using the graphical user interface or the `ldif` file.

For parameters and usage information, see "idsldapreplcfg" on page 42.

idsldapsearch

Searches existing entries from Directory Server that match a filter.

For parameters and usage information, see "idsldapsearch" on page 44.

idsldaptrace

Dynamically starts or stops Directory Server trace.

For parameters and usage information, see "idsldaptrace" on page 54

idsmonitor

Gathers information for troubleshooting.

idsunarchive

Extracts the contents of an archive file, such as `.tar`, `.zip`, `.gz`, `.tgz`, or `.bz2`.

For parameters and usage information, see "idsunarchive" on page 58.

listfiles

Lists all files and directories (including sub-directories) for the specified base directory.

For parameters and usage information, see "listfiles" on page 59.

deleteTDISysStore

Use the **deleteTDISysStore** command to delete the system store for Directory Integrator based solutions.

Description

The **deleteTDISysStore** command deletes the system store for various Directory Integrator based applications or solutions, such as Federated Directory Server with Directory Server as target, Federated Directory Server with SCIM as target, SCIM service, log management tool, and SNMP tool.

Synopsis

```
deleteTDISysStore [-b app_name]
```

Options

Use the following parameter with the **deleteTDISysStore** command:

-b *app_name*

Specifies the name of the application whose system store you want to delete.

Valid values are:

- fds
- fdsscim
- scimservice
- idslogmgmnt
- idssnmp

Example

To delete the system store for Federated Directory Server, run the following command:

```
deleteTDISysStore -b fds
```

deletefile

Use the **deletefile** command to delete a file or directory on the virtual appliance.

Description

The **deletefile** command deletes the specified file or directory from the virtual appliance base directories, such as CustomIn, CustomOut, or Certificates.

Synopsis

```
deletefile [-b base_dir] [-f file]
```

Options

Use the following parameters with the **deletefile** command:

-b *base_dir*

Specifies the virtual appliance base directory under which the file or directory that you want to delete exists.

Valid values are:

- CustomIn

- `CustomOut`
- `Certificates`

-f *file*
Name of the file or directory that you want to delete.

Example

To delete a file under the virtual appliance `CustomIn` folder, run the following command:

```
deletefile -b CustomIn -f file_name
```

idsdirctl

Use the **`idsdirctl`** command to start or stop Directory Server.

Description

The **`idsdirctl`** command is an Administration Server control program.

To run **`idsdirctl`**, you must be the primary administrator or a member of the local administrators with start or stop server authority.

Synopsis

```
idsdirctl [options] command -- [ibmslapd options]
```

Where, **command** indicates the command to run by the **`idsdiradm`** utility. The value of the parameter must be one of the following values:

`start`
Starts the Directory Server.

`stop`
Stops the Directory Server.

`restart`
Stops and then starts the Directory Server.

`status`
Indicates whether the Directory Server is running or stopped.

`statusreturn`
Sets exit code `0=running`, `1=starting`, `2=stopped`.

`admstop`
Stops the Directory Server Administration Server.

`startlogmgmt`
Starts the log management capabilities for the Directory Server.

`stoplogmgmt`
Stops the log management capabilities for Directory Server.

`statuslogmgmt`
Indicates whether the log management for Directory Server is running.

Usage

You can use the Administration Server control program, **`idsdirctl`**, to start, stop, restart, or query the status of the Directory Server. It can also be used to stop the Administration Server. For this command to function, the Administration Server, **`idsdiradm`**, must be running. For more information, see "ibmdiradm" on page 77.

If **idsslapd** options are provided, they must be preceded by **--**. Only the **-a** and **-n** parameters of **idsslapd** are supported.

To see the syntax help for**idsdirctl**, type `idsdirctl -?`.

Options

The options to the **idsdirctl** command.

-D ***adminDN***
: Specifies the bind DN to the command. You can also use `-d` instead of `-D` option.

-h ***hostname***
: Specifies the host name of the system where **ibmdiradm** is running. You can also use `-H` instead of `-h` option.

-K ***keyfile***
: Specifies the file to use for keys.

-N ***key_name***
: Specifies the private key name to use in keyfile.

-p ***port***
: Specifies the port number on which **ibmdiradm** is listening.

-P ***key_pw***
: Specifies the keyfile password.

-v
: Indicates to run in verbose mode.

-w ***adminPW***
: Specifies the bind password or `?` for non-echoed prompt. Use backslash `\?` to avoid matching single character file names on UNIX. You can also use `-W` instead of `-w` option.

-Y
: Specifies to use a secure LDAP connection by using the startTLS protocol. The `-Y` option is only supported when GSKit is installed.

-Z
: Specifies to use a secure LDAP connection by using Secure Sockets Layer (SSL) and Transport Layer Security (TLS). The `-Z` option is only supported for SSL component entry, and only when GSKit is installed.

-?
: Specifies to show the help.

-1 sec:usec
: Specifies the timeout for the `connect()` function in seconds and microseconds. The values that are provided for seconds and microseconds must be positive integers.

Examples

Example 1:

To start the server in configuration only mode, run the following command:

```
idsdirctl -h hostname -D myDN -w mypassword -p 3538 start -- -a
```

Example 2:

To stop the server, run the following command:

```
idsdirctl -h hostname -D myDN -w mypassword -p 3538 stop
```

idsgetfile

Use the **idsgetfile** command to get a file from a remote system.

Description

The **idsgetfile** command uploads a file from the specified location on a remote system to the /userdata/directory/CustomIn directory, which you can access in the virtual appliance console. See Managing custom files.

Synopsis

```
idsgetfile [-h host -u user [-p port] -f absolute_path_to_file]
```

Options

Use the following parameters with the **idsgetfile** command:

-h *host*
Specifies the host name or IP address of the remote system.

-u *user*
Specifies the user ID to access the remote system.

-p *port*
Specifies the port of the remote system. This parameter is optional.

-f *absolute_path_to_file*
Specifies the absolute path to the file on the remote system, which you want to upload.

Example

To upload a file from the specified location on the remote system to the /userdata/directory/CustomIn virtual appliance directory, run the following command:

```
idsgetfile -h hostname_or_IP -u username -f absolute_path_of_the_file_on_remote_system
```

idsldapadd, idsldapmodify

Use the **idsldapadd** and **idsldapmodify** commands to add and modify entries in Directory Server.

Description

The **idsldapadd** command is an LDAP add-entry tool, and **idsldapmodify** is an LDAP modify-entry tool. The **idsldapmodify** command is an interface to the ldap_modify and ldap_add library calls. The **idsldapadd** command is implemented as a renamed version of **idsldapmodify**. When the **idsldapadd** command is issued, the **-a** parameter, add new entry, is set automatically.

The **idsldapmodify** command opens a connection to an LDAP server and binds to the server. You can use **idsldapmodify** to modify or add entries. The command reads entry information from standard input or from a file by using the **-i** parameter.

To see the syntax help for **idsldapmodify** or **idsldapadd**, type

```
idsldapmodify -?
```

or

```
idsldapadd -?
```

Synopsis

```
idsldapmodify | idsldapmodify [-a] [-b] [-B] [-c] [-C charset] [-d debuglevel]
              [-D binddn] [-e errorfile] [-E token_pw] [-f file] [-g]
              [-G realm] [-h ldaphost] [-i file] [-I] [-j] [-k] [-K keyfile]
              [-l] [-m mechanism] [-M] [-n] [-N certificatename] [-O maxhops]
              [-p ldapport] [-P keyfilepw] [-Q operation] [-r] [-R]
              [-S token_label] [-t] [-U username] [-v] [-V] [-w passwd | ?] [-x]
              [-X lib_path] [-y proxydn] [-Y] [-Z] [-1 sec:usec]


idsldapadd | idsldapadd [-a] [-b] [-c] [-C charset] [-d debuglevel][-D binddn]
           [-e errorfile] [-E token_pw] [-f file] [-g] [-G realm]
           [-h ldaphost] [-i file] [-I] [-k] [-K keyfile] [-l] [-m mechanism]
           [-M] [-n] [-N certificatename] [O maxhops] [-p ldapport]
           [-P keyfilepw] [-Q operation] [-r] [-R] [-S token_label]
           [-U username] [-v] [-V] [-w passwd | ?][-x] [-X lib_path]
           [-y proxydn] [-Y] [-Z] [-1 sec:usec]
```

Options

-a

Adds new entries. The default action for **idsldapmodify** is to modify existing entries. If **idsldapadd** is issued, the **-a** flag is always set.

-b

Assumes a value that start with / is a binary value; and the actual value is in a file with path specified in the place of the valuer.

-B

Specifies to roll back a transaction.

-c

Specifies to run in continuous mode, and do not stop processing on error. The **idsldapmodify** command continues with operation even if errors are reported. If the **-c** parameter is not specified, the command exits if an error is encountered.

-C *charset*

Specifies the string to the command be represented in a local character set, as specified by *charset*. Use **-C** *charset* to override the default, where string must be supplied in UTF-8. When the command receives values from standard input, the specified *charset* value is used to convert the attribute values. If the value is received from an LDIF file that contains a *charset* tag, the *charset* tag in the LDIF file overrides the *charset* value that is specified to the command. For more information about the specific *charset* values that are supported for each operating system, see Chapter 6, “Supported IANA character sets,” on page 135. The supported values for *charset* are the same values that are supported for the *charset* tag that is optionally defined in version 1 LDIF files.

-d *debuglevel*

Sets the LDAP debug level to *debuglevel* in LDAP library. This option causes the utility to generate debug output to stdout. The *debuglevel* is a bit mask that controls which output is generated with values from 1 to 65535. For more information about debug levels, see Chapter 4, “Debugging levels ,” on page 131.

-D *binddn*

Specifies the *binddn* to bind to an LDAP directory. The *binddn* variable is a string-represented value. When used with **-m DIGEST-MD5**, it specifies the authorization ID. It can be either a DN or an authorization ID string that starts with dn: or u:.

-e *errorfile*

Specifies a file to which erroneous entries are written. This option must be provided with the **-c** parameter that specifies to run in continuous mode. If processing of an entry fails, that entry is written to the error file and the count of erroneous entry is increased. If input to the **idsldapmodify** or **idsldapadd** command is from a file, after the file is processed the number of entries that are written to the error file is provided.

-E *token_pw*

Specifies the token password to access a crypto device.

-f *file*

Reads an entry modification information from an LDIF file instead of standard input. If an LDIF file is not specified, you must specify the update records in LDIF format by using standard input.

Note: This option is deprecated but is supported.

-g

Specifies not to strip the trailing spaces from attribute values.

-G *realm*

Specifies the realm name. When used with **-m DIGEST-MD5**, the value is passed to the server during a bind.

-h *ldaphost*

Specifies the host name of the system where an LDAP server is running.

-i *file*

Specifies to read entry modification information from an file instead of standard input. If an LDIF file is not specified, you must specify the update records in LDIF format by using standard input.

-I

Specifies a crypto device with key storage by using PKCS11.

-j

Specifies not to send a prepare request.

-k

Specifies to send the server administration control. For information about this control, see *Programming Reference* section of the IBM Security Directory Suite documentation.

-K *keyfile*

Specifies the name of the SSL or TLS key database file with the default extension of kdb. If the key database file is not in the current directory, specify the fully qualified key database file name. If a key database file name is not specified, this utility first looks for the presence of the *SSL_KEYRING* environment variable with an associated file name. If the *SSL_KEYRING* environment variable is not defined, the default keyring file is used, if present.

A default keyring file, `ldapkey.kdb`, and the associated password stash file, `ldapkey.sth`, are installed in the `etc` directory in *IDS_LDAP_HOME*.

For more information about the default key database files and default certificate authorities (CAs), see *Programming Reference* section of the IBM Security Directory Suite documentation.

If a keyring database file cannot be located, a hard coded set of default trusted certificate authority roots is used. The key database file typically contains one or more certificates of CAs that are trusted by the client. These types of *X.509* certificates are known as trusted roots. For more information about managing an SSL or TLS key database, see *Administering* section of the IBM Security Directory Suite documentation. Also, see the Security functions section.

This parameter effectively enables the **-Z** switch.

-l

Specifies not to replicate the entry.

This parameter sends the Do not replication control to the server. For more information about this control, see *Programming Reference* section of the IBM Security Directory Suite documentation.

-m *mechanism*

Specifies the SASL mechanism to use when you bind to the server. The `ldap_sasl_bind_s()` API is used for this option. The **-m** parameter is ignored if **-V 2** is set. If **-m** is not specified, simple authentication is used.

-M

Specifies to manage referral objects as regular entries.

-n

Specifies to demonstrate the action of the operation without actually doing it. The changes that are identified are preceded by an exclamation mark and printed to standard output. Any syntax errors that are found during the processing of the file before you call the function are shown on standard error.

Tip: The **-n** parameter with the **-v** parameter is useful when you debug any related problem.

-N *certificatename*

Specifies the label that is associated with the client certificate in the key database file. If an LDAP server is configured for server authentication only, a client certificate is not required. If the LDAP server is configured for client and server authentication, a client certificate might be required. This parameter is not required if a default certificate / private key pair is assigned as the default. Similarly, *certificatename* is not required if there is a single certificate / private key pair in the designated key database file. This parameter is ignored if **-Z** or **-K** parameter is not specified.

-O *maxhops*

Specify *maxhops* to set the maximum number of hops that the client library must take when it chases the referrals. The default hop count is 10.

-p *ldapport*

Specifies a port for the LDAP server to listen. The default LDAP port is 389. If **-p** is not specified and **-Z** is specified, the default LDAP secure port, 636, is used.

-P *keyfilepw*

Specifies the key database password. Password is required to access the encrypted information in the key database file, which might include one or more private keys. If a password stash file is associated with the key database file, then the password is obtained from the password stash file. In this case, the **-P** parameter is not required. This parameter is ignored if **-Z** or **-K** is not specified.

-Q *operation*

Specifies the crypto device operation with PKCS11. The *operation* variable must be assigned one of the following values:

```
0: No accelerator mode
1: Symmetric
2: Digest
3: Digest and Symmetric
4: Random
5: Random and Symmetric
6: Random and Digest
7: Random, Digest and Symmetric
```

-r

Specifies to replace existing values by default.

-R

Specifies not to chase referrals automatically.

-S *token_label*

Specifies the token label of the crypto device.

-t

Specifies to run modify operation in a transaction.

-U *username*

Specifies the user name. This name is required with **-m DIGEST-MD5**, and is ignored when any other mechanism is used. The value of *username* depends on what attribute the server is configured to use. It might be a `uid` or any other value that is used to locate the entry.

-v

Indicates to run in verbose mode.

-V

Specifies the LDAP protocol version to use. By default, an LDAP V3 connection is established. To explicitly select LDAP V3, specify **-V 3**. To run as an LDAP V2 application, specify **-V 2**.

-w *passwd* | ?

Specifies the password for authentication. Use the `?` to generate a non-echoed password prompt.

-x

Specifies to use FIPS mode processing. This mode is applicable with SSL or TLS only.

-X *lib_path*

Specifies the library path of the crypto device.

-y *proxydn*

Specifies the DN to use for proxied authorization.

-Y

Specifies to use a secure LDAP connection by using the startTLS protocol. The `-Y` option is only supported when GSKit is installed.

-Z

Specifies to use a secure LDAP connection by using Secure Sockets Layer (SSL) and Transport Layer Security (TLS). The `-Z` option is only supported for SSL component entry, and only when GSKit is installed.

-1 sec:usec

Specifies the timeout for the `connect()` function in seconds and microseconds. The values that are provided for seconds and microseconds must be positive integers.

-?

Specifies to show the syntax format.

Input format

The contents of file or standard input if the **-i** parameter is not specified on the command line, must conform to the LDIF format.

Input format for earlier version of idsldapmodify

An input format is supported for compatibility with older versions of **idsldapmodify**. This format consists of one or more entries that are separated by blank lines, where each entry is of the following format:

```
Distinguished Name (DN)

attr=value

[attr=value ...]
```

where, `attr` is the attribute name and `value` is the attribute value.

The default action i to add values. If the **-r** parameter is specified, the default action is to replace existing values with the new one. An attribute can be specified more than one time if it is a multi-valued attribute. The multi-valued attributes can be used to add more than one value for an attribute. You can use a trailing `\\` to continue values across lines and preserve new lines in the value itself. To remove a value, `-`, hyphen, must precede the `attr` option. To remove an entire attribute, = and `value` must be omitted. To add a value along with the **-r** parameter, + must precede the `attr` option.

Notes

If you do not provide entry information by using a file with **-i** or from command line by using the *dn* and *newrdn* arguments, **ldapmodrdn** waits to read entries from standard input. To exit from the command prompt, use **Ctrl+D** on UNIX systems. On Windows systems, use **Ctrl+Z**.

Exit status

Exit status is `0` if no errors occur. If exit status is non-zero, then an error occurred. When error occurs, the messages are generated and send to the standard error.

Security functions

To use the SSL or TLS-related functions that are associated with this utility, see Chapter 3, “SSL and TLS notes,” on page 129.

See also

idsldapchangepwd, **idsldapdelete**, **idsldapexop**, **idsldapmodrdn**, **idsldapsearch**

Examples

Example 1:

Consider a `entrymods.ldif` file with the following entries:

```
dn: cn=Modify Me, o=University of Higher Learning, c=US
changetype: modify
replace: mail
mail: modme@student.example.com
-
add: title
title: Grand Poobah
-
add: jpegPhoto
jpegPhoto: modme.jpeg
-
delete: description
```

The `entrymods.ldif` requests for the following changes to an entry:

- Replace the `mail` attribute of the `cn=Modify Me` entry with the *modme@student.example.com* value
- Add the `title` attribute with the *Grand Poobah* value
- Add a file `modme.jpeg` as `jpegPhoto`
- Remove the `description` attribute

To make the following changes, run the **idsldapmodify** command:

```
idsldapmodify -D adminDN -w adminPWD -b -r -i entrymods.ldif
```

Example 2:

To modify an entry by using an earlier version of **idsldapmodify** command, run the command with the following input format:

```
idsldapmodify -D adminDN -w adminPWD
cn=Modify Me, o=University of Higher Learning, c=US
mail=modme@student.example.com
+title=Grand Poobah
+jpegPhoto=modme.jpeg
-description
```

Example 3:

To add an entry by using the `entryadd.ldif` file, run the **idsldapadd** command:

```
idsldapadd -D adminDN -w adminPWD -i entryadd.ldif
```

where, `entryadd.ldif` contains:

```
dn: cn=John Doe, o=University of Higher Learning, c=US
objectClass: person
cn: John Doe
cn: Johnny
sn: Doe
title: the world’s most famous mythical person
mail: johndoe@student.example.com
uid: jdoe
```

Example 4:

To delete an entry by using the `removeentry.ldif` file, run the **idsldapmodify** command:

```
idsldapmodify -D adminDN -w adminPWD -i removeentry.ldif
```

where, `removeentry.ldif` contains:

```
dn: cn=John Doe, o=University of Higher Learning, c=US
changetype: delete
```

idsldapchangepwd

Use the **idsldapchangepwd** command to modify password for an entry in the directory information tree (DIT).

Description

The **idsldapchangepwd** command is an LDAP modify password tool. This command sends modify password requests to an LDAP server.

Note:

1. The **idsldapchangepwd** command cannot be used to change password for the primary administrator or for members of administrative group. The **idsldapchangepwd** command works only with directory entries.
2. The **idsldapchangepwd** command works only on the `userpassword` attribute.

Synopsis

```
idsldapchangepwd | idsldapchangepwd -D binddn -w passwd | ? -n newpassword | ?
                 [-C charset] [-d debuglevel] [-E token_pw] [-G realm]
                 [-h ldaphost] [-I] [-K keyfile] [-m mechanism] [-M]
                 [-N certificatename] [-O maxhops] [-p ldapport]
                 [-P keyfilepw] [-Q operation] [-R] [-S token_label]
                 [-U username] [-v] [-V version] [-x] [-X lib_path]
                 [-y proxydn] [-Y] [-Z] [-1 sec:usec] [-?]
```

Options

The options to the **idsldapchangepwd** command.

-C ***charset***
Specifies that the DNs supplied as parameter to the **idsldapchangepwd** utility are represented in a local character set, as specified by *charset*. Use**-C** *charset* to override the default, where strings must be supplied in UTF-8. For more information about the specific *charset* values that are supported for each operating system, see Chapter 6, “Supported IANA character sets,” on page 135. The supported values for *charset* are the same values that are supported for the *charset* tag that is optionally defined in version 1 LDIF files.

-d ***debuglevel***
Sets the LDAP debugging level to *debuglevel*. This option causes the utility to generate debug output to stdout. The *debuglevel* is a bit mask that controls which output is generated with values up to 65535. For information about debug levels, see Chapter 4, “Debugging levels ,” on page 131.

-D ***bindDN***
Specifies the *bindDN* to bind to an LDAP directory. The *bindDN* variable is a string-represented value. When used with **-m DIGEST-MD5**, it specifies the authorization ID. It can be either a DN or an authorization ID string that starts with `dn:` or `u:`.

-E ***token_pw***
Specifies the token password to access a crypto device.

-G ***realm***
Specifies the realm name. When used with **-m DIGEST-MD5**, the value is passed to the server during a bind.

-h ***ldaphost***
Specifies the host name of the system where an LDAP server is running.

-I

Specifies a crypto device with key storage by using PKCS11.

-K *keyfile*

Specifies the name of the SSL or TLS key database file with the default extension of kdb. If the key database file is not in the current directory, specify the fully qualified key database file name. If a key database file name is not specified, this utility first looks for the presence of the *SSL_KEYRING* environment variable with an associated file name. If the *SSL_KEYRING* environment variable is not defined, the default keyring file is used, if present.

A default keyring file, `ldapkey.kdb`, and the associated password stash file, `ldapkey.sth`, are installed in the `etc` directory in *IDS_LDAP_HOME*.

For more information about the default key database files and default certificate authorities (CAs), see *Programming Reference* section of the IBM Security Directory Suite documentation.

If a keyring database file cannot be located, a hard coded set of default trusted certificate authority roots is used. The key database file typically contains one or more certificates of CAs that are trusted by the client. These types of *X.509* certificates are known as trusted roots. For more information about managing an SSL or TLS key database, see *Administering* section in the IBM Security Directory Suite documentation. Also, see the Security functions section.

This parameter effectively enables the **-Z** switch.

-m *mechanism*

Specifies the SASL mechanism to use when you bind to the server. The `ldap_sasl_bind_s()` function is used. The **-m** parameter is ignored if `-V 2` is set. If **-m** is not specified, simple authentication is used.

-M

Specifies to manage referral objects as regular entries.

-n *newpassword* | ?

Specifies the new password. Use ? to generate a password prompt. If you use the password prompt, it prevents your password from being visible through the **ps** command.

-N *certificatename*

Specifies the label that is associated with the client certificate in the key database file. If an LDAP server is configured to use server authentication only, a client certificate is not required. If the LDAP server is configured to use client and server authentication, a client certificate might be required. This parameter is not required if a default certificate / private key pair is assigned as the default. Similarly, *certificatename* is not required if there is a single certificate / private key pair in the designated key database file. This parameter is ignored if **-Z** or **-K** parameter is not specified.

-O *maxhops*

Specify *maxhops* to set the maximum number of hops that the client library takes when it chases the referrals. The default hop count is 10.

-p *ldapport*

Specifies a port for the LDAP server to listen. The default LDAP port is 389. If **-p** is not specified and **-Z** is specified, the default LDAP secure port 636 is used.

-P *keyfilepw*

Specifies the key database password. Password is required to access the encrypted information in the key database file, which might include one or more private keys. If a password stash file is associated with the key database file, then the password is obtained from the password stash file. In this case, the **-P** parameter is not required. This parameter is ignored if **-Z** or **-K** is not specified.

-Q *operation*

Specifies the crypto device operation with PKCS11. The *operation* variable must be assigned one of the following values:

```
0: No accelerator mode
1: Symmetric
2: Digest
3: Digest and Symmetric
4: Random
```

```
5: Random and Symmetric
6: Random and Digest
7: Random , Digest and Symmetric
```

-R
Specifies not to chase referrals automatically.

-S *token_label*
Specifies the token label of the crypto device.

-U *username*
Specifies the user name. This name is required with **-m DIGEST-MD5**, and is ignored when any other mechanism is used. The value of *username* depends on what attribute the server is configured to use. It might be a `uid` or any other value that is used to locate the entry.

-v
Indicates to run in verbose mode. With this option, messages are written to the standard output.

-V *version*
Specifies the LDAP version to use. By default, an LDAP V3 connection is established. To explicitly select LDAP V3, specify **-V 3**. To run as an LDAP V2 application, specify **-V 2**. An application, like **ldapdchangepwd**, selects LDAP V3 as the preferred protocol by using `ldap_init` instead of `ldap_open`.

-w *passwd* | ?
Specifies the password for authentication. Use the ? to generate a non-echoed password prompt. If you use the password prompt, it prevents your password from being visible through the **ps** command.

-x
Specifies to use FIPS mode processing. This mode is applicable with SSL or TLS only.

-X *lib_path*
Specifies the library path of the crypto device.

-y *proxydn*
Specifies the DN to use for proxied authorization.

-Y
Specifies to use a secure LDAP connection by using the startTLS protocol. The `-Y` option is only supported when GSKit is installed.

-Z
Specifies to use a secure LDAP connection by using Secure Sockets Layer (SSL) and Transport Layer Security (TLS). The `-Z` option is only supported for SSL component entry, and only when GSKit is installed.

-1 sec:usec
Specifies the timeout for the `connect()` function in seconds and microseconds. The values that are provided for seconds and microseconds must be positive integers.

-9 p
Sets criticality for paging to false. The search is handled without paging.

-9 s
Sets criticality for sorting to false. The search is handled without sorting.

-?
Specifies to show the syntax format.

Exit status

Exit status is `0` if no errors occur. If exit status is non-zero, then an error occurred. When error occurs, messages are written to the standard error.

Security functions

To use the SSL or TLS-related functions that are associated with this utility, seeChapter 3, "SSL and TLS notes," on page 129.

See also

idsldapadd, **idsldapdelete**, **idsldapexop**, **idsldapmodify**, **idsldapmodrdn**, **idsldapsearch**

Examples

Example 1:

To modify the password for an entry, run the **idsldapchangepwd** command:

```
idsldapchangepwd -h hostname -D myDN -w mypassword -n myNewPassword
```

In this example, the **idsldapchangepwd** command changes the password for the *myDN* entry from *mypassword* to *myNewPassword*.

idsldapcompare

Use the **idsldapcompare** to compare an attribute value of an entry in an LDAP server with your compare criteria.

Description

The **idsldapcompare** utility sends a compare request to an LDAP server. The **idsldapcompare** utility compares the attribute value of an entry with a user provided value. The command returns `true` or `false` as output based on the result of the compare request.

Synopsis

```
idsldapcompare | idsldapcompare[-c] [-d level] [-D DN] [-f file]
               [-G realm][-h host] [-m mechanism] [-n] [-p port]
               [-P on|off] [-R] [-U username] [-v] [-V version]
               [- w password|?] [-y proxyDN] [-1 sec:usec]
```

The syntax of the **idsldapcompare** command:

```
idsldapcompare [options] [dn attr=value]
```

where,

- *dn*: The DN entry for compare.
- *attr*: The attribute to use in the compare.
- *value*: The value to use in the compare.

Options

The options to the **idsldapcompare** command.

-c

Specifies to run the operation in continuous mode. In this mode, even after an error is reported the compare operation is continued. The default action is to exit the operation on an error.

-d *level*

Sets the LDAP debugging level to *level* in the LDAP library. This option causes the utility to generate debug output to stdout. The *level* value is a bit mask that controls which output is generated with values from 1 to 65535. For more information about debug levels, see Chapter 4, “Debugging levels ,” on page 131.

-D *DN*

Specifies the bind DN to bind to a Directory Server.

-f *file*

Specifies to run compare operation sequentially by using the values in the *file*.

-G *realm*

Specifies the realm name for use with **-m DIGEST-MD5** bind mechanism.

-h ***host***
Specifies the host name of the system on which an LDAP server is running.

-m ***mechanism***
Specifies the SASL mechanism to use when you bind to the server.

-n
Specifies to demonstrate the action for the operation without actually doing it.

Tip: The **-n** parameter with the **-v** parameter is useful when you debug any related problem.

-p ***port***
Specifies a port number for the LDAP server to listen.

-P ***on | off***
Specifies whether to send password policy controls to the server. The argument to the **-P** parameter indicates:

on - send the password policy controls
off - do not send password policy controls

-R
Specifies not to chase referrals automatically.

-U ***username***
Specifies the user name for the **DIGEST-MD5** bind.

-v
Specifies to run the command in verbose mode.

-V ***version***
Specifies the LDAP protocol version. The default version is 3.

-w ***passwd*** **| ?**
Specifies the bind password for authentication. Use the **?** to generate a non-echoed password prompt.

-y ***proxydn***
Specifies the DN to be used for proxied authorization.

-1 sec:usec
Specifies the timeout for the `connect()` function in seconds and microseconds. The values that are provided for seconds and microseconds must be positive integers.

Examples

Example 1:
To compare an attribute value with user provided value for an entry, run the **idsldapcompare** command of the following format:

```
idsldapcompare -D adminDN -w adminPWD -h host_name -p port \
"cn=Bob Campbell, ou=Austin, o=sample" postalcode=4502
```

In this example, the command compares the entry with an existing entry in the LDAP server. If the postal code for the *cn=Bob Campbell* entry is 4502 in the server, the command returns `true`, otherwise the command returns `false`.

idsldapdelete

Use the **ldapdetele** command to delete one or more entries from directory information tree (DIT).

Description

The **idsldapdelete** command is a command-line interface to the `ldap_delete` library call.

The **idsldapdelete** command opens a connection to an LDAP server, binds to the LDAP server, and deletes one or more entries. If one or more DN arguments are provided, entries with those DNs are deleted. Each DN is a string-represented value. If no DN arguments are provided, a list of DNs is read from standard input or from a file if the **-i** or **-f** flag is used.

To see syntax help for **idsldapdelete**, type:

```
idsldapdelete -?
```

Usage

```
idsldapdelete [options] [DNs]
idsldapdelete [options] [-i file]
```

where,

DNs: indicates one or more entries to delete
file: specifies the name of the file with entries to delete

Note: If a distinguished name (DN) or file is not specified, then entries are read from standard input.

Options

The options to the **idsldapdelete** command.

-c
Specifies to run continuous operation, and do not stop processing on error.

-C *charset*
Specifies the character set name to use, as registered with IANA. For more information about the specific *charset* values that are supported for each operating system, see Chapter 6, "Supported IANA character sets," on page 135. The supported values for *charset* are the same values that are supported for the *charset* tag that is optionally defined in version 1 LDIF files.

-d *level*
Sets the LDAP debug level to *level* in LDAP library. This option causes the utility to generate debug output to stdout. The *level* is a bit mask that controls which output is generated with values from 1 to 65535. For information about debug levels, see Chapter 4, "Debugging levels ," on page 131.

-D *DN*
Specifies the *DN* to bind to an LDAP directory.

-E *token_pw*
Specifies the token password to access a crypto device.

-f *file*
Specifies the file from which to read DN for deletion. The file must contain only one DN entry per line.

-G *realm*
Specifies the realm name for use with **-m DIGEST-MD5** bind mechanism.

-h *host*
Specifies the host name of the system where an LDAP server is running.

-i *file*
Specifies the file from which to read DN for deletion. The file must contain only one DN entry per line.

-I
Specifies a crypto device with key storage by using PKCS11.

-k
Specifies to send the server administration control. For information about the server administration control, see the *Programming Reference* section in the IBM Security Directory Suite documentation.

-K *keyfile*
Specifies the name of the SSL or TLS key database file with the default extension of kdb.

A default keyring file, `ldapkey.kdb`, and the associated password stash file, `ldapkey.sth`, are installed in the `etc` directory in *IDS_LDAP_HOME*.

For information about the default key database files and default certificate authorities (CAs), see *Programming Reference* section of the IBM Security Directory Suite documentation.

If a keyring database file cannot be located, a hard coded set of default trusted certificate authority roots is used. The key database file typically contains one or more certificates of CAs that are trusted by the client. These types of *X.509* certificates are known as trusted roots. For information about managing an SSL or TLS key database, see *Administering* section of the IBM Security Directory Suite documentation. Also, see the Security functions section.

This parameter effectively enables the **-Z** switch.

-l
Specifies not to replicate the entry.

This parameter sends the Do not replication control to the server. For information about this control, see *Programming Reference* section of the IBM Security Directory Suite documentation.

-L
Specifies to read DN from the file in LDIF format.

-m *mechanism*
Specifies the SASL mechanism to use when you bind to the server.

-M
Specifies to manage referral objects as regular entries.

-n
Specifies to demonstrate the action of the operation without actually doing it.

Tip: The **-n** parameter with the **-v** parameter is useful when you debug any related problem.

-N *key_name*
Specifies the private key name to use in the key file. This parameter is ignored if **-Z** or **-K** parameter is not specified.

-O *maxhops*
Specifies the maximum number referrals to chase in a sequence.

-p *port*
Specifies a port for the LDAP server to listen. The default LDAP port is 389. If **-p** is not specified and **-Z** is specified, the default LDAP secure port, 636, is used.

-P *key_pw*
Specifies the key database password. This parameter is ignored if **-Z** or **-K** is not specified.

-Q *operation*
Specifies the crypto device operation with PKCS11. The *operation* variable must be assigned one of the following values:

```
0: No accelerator mode
1: Symmetric
2: Digest
3: Digest and Symmetric
4: Random
5: Random and Symmetric
6: Random and Digest
7: Random , Digest and Symmetric
```

-R
Specifies not to chase referrals automatically.

-s
Specifies to delete a subtree from an LDAP server. This parameter sends the subtree delete control.

⚠ **CAUTION:** Subtree delete control request specifies to delete the subtree and all descendant entries under this subtree.

For more information about this control, see the *Programming Reference* section in the IBM Security Directory Suite documentation.

-S *token_label*
Specifies the token label of the crypto device.

-U *username*
Specifies the user name. This name is required with **-m DIGEST-MD5**, and is ignored when any other mechanism is used. The value of *username* depends on what attribute the server is configured to use. It might be a uid or any other value that is used to locate the entry.

-v
Indicates to run in verbose mode.

-V *version*
Specifies the LDAP protocol version to use. By default, an LDAP V3 connection is established.

-w *passwd* | ?
Specifies the password for authentication. Use the ? to generate a non-echoed password prompt. In UNIX, use backslash **\?** to avoid matching single character file names.

-x
Specifies to use FIPS mode processing. This mode is applicable with SSL or TLS only.

-X *lib_path*
Specifies the driver path of the crypto device.

-y *proxydn*
Specifies the DN to be used for proxied authorization.

-Y
Specifies to use a secure LDAP connection by using the startTLS protocol. The -Y option is only supported when GSKit is installed.

-Z
Specifies to use a secure LDAP connection by using Secure Sockets Layer (SSL) and Transport Layer Security (TLS). The -Z option is only supported for SSL component entry, and only when GSKit is installed.

-1 sec:usec
Specifies the timeout for the connect() function in seconds and microseconds. The values that are provided for seconds and microseconds must be positive integers.

-?
Specifies to show the syntax format.

Notes

If you do not provide DN arguments, the **idsldapdelete** command waits to read a list of DNs from standard input. To exit from the command prompt, use **Ctrl+D** on UNIX systems. On Windows systems, use **Ctrl+Z**.

Exit status

Exit status is 0 if no errors occur. If exit status is non-zero, then an error occurred. When error occurs, messages are written to the standard error.

Security functions

To use the SSL or TLS-related functions that are associated with this utility, see Chapter 3, “SSL and TLS notes,” on page 129.

See also

idsldapadd, **idsldapchangepwd**, **idsldapexop**, **idsldapmodify**, **idsldapmodrdn**, **idsldapsearch**

Examples

Example 1:

To delete an entry from a Directory Server instance, run the **idsldapdelete** command:

```
idsldapdelete -D adminDN -w adminPWD -h host -p port \
 "cn=Delete Me, o=University of Life, c=US"
```

The command attempts to delete the `cn=Delete Me` entry, which is directly under the `University of Life` organizational entry.

idsldapdiff

Use the **idsldapdiff** command to identify the differences in a replica server and its master server. You can also synchronize the replica server with its master server.

Description

You can use the**idsldapdiff** command to compare two directory subtrees on two different Directory Servers to determine whether their contents match. You can also use this command to synchronize any entries that do not match. You might want to synchronize the following two types of differences:

- Entries that have the same DN, but different contents.
- Entries that are present on one server, but not the other.

The following list shows the operational attributes that **idsldapdiff** compares and fixes.

ACL-related

- `aclEntry`
- `aclPropagate`
- `aclSource`
- `entryOwner`
- `ownerPropagate`
- `ownerSource`
- `ibm-filterAclEntry`
- `ibm-filterAclInherit`

Password policy-related

- `pwdChangedTime`
- `pwdReset`
- `ibm-pwdAccountLocked`
- `ibm-pwdIndividualPolicyDN`
- `ibm-pwdGroupPolicyDN`

Other operational attributes

- `ibm-entryUuid`
- `creatorsName`
- `createTimeStamp`
- `modifiersName`
- `modifyTimeStamp`

You must run the command when no updates are queued up or made on both the replica and master servers. The administrator must quiesce or suspend all update activities to the two subtrees that are compared. When you use the **idsldapdiff** command for compare operation, you must suspend update operations on the directory server. If the command is run while the updates are made, then all discrepancies might not be accurately reported or fixed.

Note: The **idsldapdiff** command does not check whether the servers are quiesced before it processes the request. When the tool is run in compare-only mode, the administrator might want to track down few discrepancies as an alternative to stopping updates completely.

If the command is run with the fix operation mode, use the command with the server administration control, the **-a** option. With the server administration control option, the tool writes to a read-only replica and also modifies operational attributes such as ibm-entryUuid.

You can also use the **idsldapdiff** command to bring a master and replica server in sync before you start replication. For the command to function, it requires the base DN, which is being compared, exists on both servers. If the base DN does not exist on either of the servers, the command gives an error and then exits.

The command traverses to each entry in the subtree on the master server and compares its contents with the corresponding entry on the replica server. Since each entry is read, running the utility can take a long time and can generate lots of read requests to the master and replica servers. Depending on the number of differences and whether in the fix operation mode, the tool generates an equal amount of write requests to the replica server.

Ideally, use the tool when replication is set for the first time between the servers. For example, if your topology has two peer masters and two replica servers, you might want to run **idsldapdiff** between *peer 1* and *peer 2*. Thereafter, if replication is suspended, run **idsldapdiff** concurrently between *peer 1* and *replica 1*; and between *peer 2* and *replica 2*. If replication is set up correctly, every change on a master server is propagated to its replica servers. If a replication problem occurs, the tool can be run to identify and correct the problems. This command is a diagnostic and corrective tool, it is not designed to run as routine maintenance. An administrator might decide to run the tool if there are replication-related errors in the log files.

To see syntax help for **idsldapdiff**, type:

```
idsldapdiff -?
```

Note:

- If the **idsldapdiff** command is used between a server of latest version and a server of previous supported version, then the tool reports differences for entries even if there are no user attribute changes. It is because of the higher granularity of timestamps, which is set to microseconds. Therefore, it is advisable not to use the **idsldapdiff** command in such scenarios.
- The **idsldapdiff** command shows an appropriate message after it finishes comparing every 100th entry.

Synopsis

To compare and optionally fix the differences:

```
idsldapdiff | idsldapdiff -b baseDN -sh host -ch host [-a] [-C countnumber]
            [-cD dn] [-cK keyStore] [-cw password] -[cN keyStoreType]
            [-cp port] [-cP keyStorePwd] [-ct trustStoreType]
            [-cT trustStore] [-cY trustStorePwd] [-cZ] [-F] [-j]
            [-L filename] [-O] [-sD dn] [-sK keyStore] [-sw password]
            [-sN keyStoreType] [-sp port] [-sP keyStorePwd]
            [-st trustStoreType] [-sT trustStore] [-sY trustStorePwd]
            [-sZ]
```

To compare schema:

```
idsldapdiff | idsldapdiff -S -sh host -ch host [-a] [-C countnumber]
            [-cD dn] [-cK keyStore] [-cw password] -[cN keyStoreType]
            [-cp port] [-cP keyStorePwd] [-ct trustStoreType]
            [-cT trustStore] [-cY trustStorePwd] [-cZ] [-j]
            [-L filename] [-O] [-sD dn] [-sK keyStore] [-sw password]
            [-sN keyStoreType] [-sp port] [-sP keyStorePwd]
            [-st trustStoreType] [-sT trustStore] [-sY trustStorePwd]
            [-sZ]
```

Guidelines for encryption

The **idsldapdiff** tool searches against `cn=configuration` to determine the encryption settings on the server. For search and fix operations, the administrator DN or administrator group DN is required. The tool fails if a bind DN other than the administrator DN or an administrative group member DN is used. Global administrators cannot run the **idsldapdiff** tool with compare and fix options. Only administrators and administrator group members can run **idsldapdiff** with compare and fix options.

The master and replica servers can have different encryption settings. For example:

- Non-matching one-way encryption scheme
- Two-way and one-way encryption schemes
- Two-way encryption schemes with different key stash files

Based on the type of encryption that is used, the behavior of an operation might vary, when a password or any other encrypted attribute is encountered.

Non-matching one-way encryption scheme

With this encryption setting, the servers are configured with different types of one-way encryption scheme. For example, the master server is set to use sha and the replica server is set to use `crypt` encryption scheme. On running the **idsldapdiff** tool, the value on a replica server is directly overwritten with the value from the master server. Running the **idsldapdiff** tool a second time on the same entries does not show any difference.

Two-way and one-way encryption schemes

In this encryption type, one of the servers is using a two-way encryption scheme like AES, and the other server is using one-way encryption scheme such as sha. Depending on whether the master server is using two-way or one-way encryption scheme, the results of the setup are different. When multiple encryption type is used, the performance of the **idsldapdiff** tool gets degraded.

- When a master is set with a two-way encryption scheme and the replica is set with a one-way encryption scheme, **idsldapdiff** shows that the two entries are different even if the actual values are the same. It is because the value on master is in plain text and the value on replica is encrypted. Running the **idsldapdiff** tool for a second time on the same entries shows the difference even though the actual values are the same.
- When the master has a one-way encryption scheme and the replica has a two-way encryption scheme, the values on replica are directly overwritten with the values on the master. Running the **idsldapdiff** tool for a second time on the same entries does not show any difference.

Two-way encryption schemes with different key stash files

In this case, both servers are using two-way encryption schemes but their stash files are generated with different seed or salt values. Since both servers decrypt, performance of the **idsldapdiff** tool is degraded. If the decrypted values are different, the synchronization process further degrades the performance of the**idsldapdiff** tool.

Note:

1. The password policy attributes are synchronized by the **idsldapdiff** tool only if the password policy is enabled on both the servers.
2. The **idsldapdiff** tool checks the encryption settings on both the servers. It shows warning messages if the encryption settings are different on both the servers, or if the seed and salt values are different on both servers.
3. Use the **idsldapdiff** tool only for schema comparison. Do not use **idsldapdiff** with the **-F** option.

Options

The options to the **idsldapdiff** command. There are two subgroups that apply only on the supplier server or the consumer server.

-a
Specifies to include server administration control for writing to a read-only replica.

-b *baseDN*
Specifies to use the *baseDN* search base as the starting point for the search instead of the default. If **-b** is not specified, this tool examines the *LDAP_BASEDN* environment variable for a search base definition.

-C *countnumber*
Counts the number of non-matching entries. If more than the specified number of mismatches are found, the tool exits.

-F
Specifies to use the fix option. If specified, content on the replica server is modified to match the content of the master server. This option cannot be used if the **-S** is also specified.

-j
Excludes the following operational attributes from the LDIF file.

- `creatorsName`
- `createTimeStamp`
- `modifiersName`
- `modifyTimeStamp`

Note: The **-j** option is only valid when the **-L** option is specified.

-L *filename*
Generates an LDIF file with the specified *filename* for output. For virtual appliance, the path for the file is `/userdata/directory/CustomOut` directory.

Use this option only if the **-F** option is not specified. The LDIF file can be used to update the replica server to eliminate the differences.

-O
Specifies to list DNs for non-matching entries.

Note: This option overrides the **-F** and **-L** options.

-S
Specifies to compare the schema on both of the servers. Compares and fixes by using the **-S** option can be made with any bind DN.

-x
Ignores extra entries on the replica.

The **idsldapdiff** tool takes two passes to synchronize the servers. In the first pass, **idsldapdiff** traverses the master server and does the following actions:

- Adds any extra entries on the master to the replica
- Compares and fixes entries that exist on both the servers

In the second pass, **idsldapdiff** traverses the replica server to check for any extra entries on the replica. Specifying the **-x** option causes **idsldapdiff** to skip the second pass.

Options for a replication supplier server

The following options apply to a replication supplier server and are denoted by a prefix s in the option.

-sD *dn*
Specifies to use *dn* to bind to an LDAP directory. The *dn* variable is a string-represented value.

-sh *host*
Specifies the host name.

-sK *keystore*
Specifies the name of the SSL keystore file with the default extension of `jks`. The path for the key database file is `/userdata/directory/Certificates`. This keystore file must contain the SSL certificate that is extracted from the key database (kdb) file used by the supplier LDAP server.

This parameter effectively enables the **`-sZ`** switch.

When you use the **`-sK`** parameter, you must also use the following flags with valid values: **`-sP`**, **`-sN`**, **`-sT`**, **`-sY`**, **`-st`**.

`-sN` ***keyStoreType***
Specifies the type of the SSL keystore. For this version of **`idsldapdiff`** the only supported type is `jks`. This parameter is ignored if the **`-sZ`** or **`-sK`** parameter is not specified.

`-sp` ***ldapport***
Specifies a port for the LDAP server to listen. The default LDAP port is 389. If **`-sp`** is not specified and **`-sZ`** is specified, the default LDAP secure port, 636, is used.

`-sP` ***keyStorePwd***
Specifies the keystore password. This password is required to access the encrypted information in the keystore file, which might include one or more private keys. This parameter is ignored if **`-sZ`** or **`-sK`** is not specified.

`-st` ***trustStoreType***
Specifies the type of the SSL truststore. For this version of **`idsldapdiff`** the only supported type is `jks`. This parameter is ignored if **`-sZ`** or **`-sT`** is not specified.

`-sT` ***trustStore***
Specifies the name of the SSL truststore file with default extension of `jks`. The path for the truststore file is `/userdata/directory/Certificates`. If the truststore file is not in the current directory, specify the fully qualified truststore file name. This truststore file can be the same as or different from the file keystore (see the description of the **`-sK`** flag). This file is sufficient if the supplier LDAP server is using the SSL server authentication. If the supplier LDAP server is using the SSL server client authentication, then the default certificate from truststore must be extracted. You must then add the certificate to the key database (kdb) used by the supplier LDAP server.

This parameter effectively enables the **`-sZ`** switch.

`-sw` ***password* | ?**
Specifies to use *password* as the password for authentication. Use the ? to generate a password prompt. The password prompt option prevents your password from being visible when you use the **ps** command.

`-sY` ***trustStorePwd***
Specifies a password for the trusted store file. This password is required to access the encrypted information in the truststore file, which can include one or more private keys.

`-sZ`
Specifies to use a secure SSL connection to communicate with an LDAP server.

Options for a replication consumer server

The following options apply to a replication consumer server and are denoted by a prefix c in the option.

`-cD` ***dn***
Specifies to use *dn* to bind to an LDAP directory. The *dn* variable is a string-represented value.

`-ch` ***host***
Specifies the host name.

`-cK` ***keystore***
Specifies the name of the SSL keystore file with the default extension of `jks`. The path for the keystore file is `/userdata/directory/Certificates`. This keystore file must contain the SSL certificate that is extracted from the key database (kdb) file used by the consumer LDAP server.

This parameter effectively enables the **`-cZ`** switch. The **`-cK`** parameter also requires you to provide the following flags with appropriate values: **`-cP`**, **`-cN`**, **`-cT`**, **`-cY`**, **`-ct`**.

-cN ***keyStoreType***
Specifies the type of the SSL keystore. For this version of **idsldapdiff** the only supported type is jks. This parameter is ignored if the **-cZ** or **-cK** parameter is not specified.

-cp ***ldapport***
Specifies a port for the LDAP server to listen. The default LDAP port is 389. If **-cp** is not specified and **-cZ** is specified, the default LDAP secure port, 636, is used.

-cP ***keyStorePwd***
Specifies the keystore password. This password is required to access the encrypted information in the keystore file, which might include one or more private keys. This parameter is ignored if **-cZ** or **-cK** is not specified.

-ct ***trustStoreType***
Specifies the type of the SSL truststore. For this version of **idsldapdiff** the only supported type is jks. This parameter is ignored if **-cZ** or **-cT** is not specified.

-cT ***trustStore***
Specifies the name of the SSL truststore file with default extension of jks. The path for the truststore file is /userdata/directory/Certificates. This truststore file can be same as or different from the keystore file (see the **-sK** flag description). This file is sufficient if the supplier LDAP server is using the SSL server authentication. If the consumer LDAP server is using the SSL server client authentication, then the default certificate from truststore must be extracted. You must add the certificate to the key database (kdb) used by the consumer LDAP server.

This parameter effectively enables the **-cZ** switch.

-cw ***password* | ?**
Specifies to use *password* as the password for authentication. Use the ? to generate a password prompt. The password prompt option prevents your password from being visible when you use the **ps** command.

-cY ***trustStorePwd***
Specifies a password for the trusted store file. This password is required to access the encrypted information in the truststore file, which can include one or more private keys.

-cZ
Specifies to use a secure SSL connection to communicate with an LDAP server.

Notes

If no DN arguments are provided, the **idsldapdiff** command waits to read a list of DNs from standard input. To exit from the command prompt, use **Ctrl+D** on UNIX systems.

Exit status

Exit status is 0 if no errors occur. If exit status is non-zero, then an error occurred. When error occurs, messages are written to the standard error.

Security functions

To use the SSL or TLS-related functions that are associated with this utility, search for *SSL, TLS notes* in the *Command Reference* section of the IBM Security Directory Suite documentation.

Examples

Example 1:

To see the differences that the tool reports, consider two servers one a master server and other a replica server. Consider that the suffix o=sample is present on both the servers. The entries in the master and replica servers are represented by using the two LDIF files, master.ldif and replica.ldif.

An example master.ldif file with entries:

```
dn: cn=Entry1,o=sample
    objectclass: inetOrgPerson
```

```
     objectclass: organizationalPerson
     objectclass: person
     objectclass: top
     objectclass: ePerson
     sn: entry1
     cn: testEntry1

     dn: cn=Entry2,o=sample
     objectclass: inetOrgPerson
     objectclass: organizationalPerson
     objectclass: person
     objectclass: top
     objectclass: ePerson
     sn: entry2
     cn: testEntry
```

An example `replica.ldif` file with entries:

```
     dn: cn=Entry2,o=sample
     objectclass: inetOrgPerson
     objectclass: organizationalPerson
     objectclass: person
     objectclass: top
     objectclass: ePerson
     sn: abcd
     cn: testEntry

     dn: cn=Entry3,o=sample
     objectclass: inetOrgPerson
     objectclass: organizationalPerson
     objectclass: person
     objectclass: top
     objectclass: ePerson
     sn: entry3
     cn: testEntry
```

To compare and fix the differences, run the **idsldapdiff** command.

```
 idsldapdiff -b o=sample -sh master -sD cn=root -sw passwd -ch replica
-cD cn=root -cw passwd -F -a
```

The following actions are the results of the command:

1. Entry `cn=Entry1,o=sample` gets added on the replica server. This entry is on the master server, but was not on the replica server.
2. Entry `cn=Entry2,o=sample` gets modified on the replica server. The value of the `sn` attribute gets modified to match the value on the master server.
3. Entry `cn=Entry3,o=sample` gets deleted from the replica server. The `cn=Entry3` entry is deleted because it is in the replica server but is not in the master server.

Example 2:

To find differences in schema of Directory Servers, run the **`idsldapdiff`** command.

```
idsldapdiff -S -sh supplier -sD cn=root -sw passwd -ch consumer
-cD cn=root -cw passwd
```

Example 3:

To compare and optionally fix the differences when the servers are configured for secure communications, run the following command:

```
idsldapdiff -b o=sample -sh supplier -sp 636 -sD cn=root
-sw password -sZ -sK keyfile.jks -sP keyStorePwd -sN jks
-sT keyfile.jks -sY trustStorePwd -st jks -ch consumer
-cp 636 -cD cn=root -cw password -cZ -cK keyfile.jks -cP keyStorePwd
-cN jks -cT keyfile.jks -cY trustStorePwd -ct jks -F -a
```

Example 4:

To compare schemas of servers that are configured for secure communications, run the following command:

```
idsldapdiff -S -sh supplier -sp 636 -sD cn=root -sw password -sZ
-sK keyfile.jks -sP keyStorePwd -sN jks -sT keyfile.jks -sY trustStorePwd -st jks
-ch consumer -cp 636 -cD cn=root -cw password -cZ -cK keyfile.jks -cP keyStorePwd
-cN jks -cT keyfile.jks -cY trustStorePwd -ct jks
```

idsldapexop

Use the **idsldapexop** command to run extended operations.

Description

The **idsldapexop** is an LDAP extended operation tool. The **idsldapexop** command provides the capability to bind to a directory and issue an extended operation along with any data that makes up the extended operation value.

The **idsldapexop** command supports the standard host, port, SSL, TLS, and authentication options that are used by LDAP client utilities. With this command, a set of options is defined to specify the operation and the arguments for each extended operation

To list syntax help for **idsldapexop**, type:

```
idsldapexop -?
```

or

```
idsldapexop -help
```

Synopsis

```
idsldapexop | idsldapexop[-C charset] [-d debuglevel][-D binddn][-e] [-E token_pw]
             [-G realm] [-h ldaphost] [-help] [-I] [-K keyfile] [-m mechanism]
             [-N certificatename] [-p ldapport] [-P keyfilepw] [-Q operation]
             [-?] [-S token_label] [-U username] [-v] [-w passwd | ?] [-x]
             [-X lib_path] [-y proxyDN] [-Y] [-Z] [-1 sec:usec]
             -op {acctstatus | backuprestore | cascrepl | clearlog | controlqueue |
             controlrepl | controlreplerr | evaluategroups | effectpwdpolicy |
             getattributes | getlogsize | getusertype | locateEntry | onlineBackup |
             quiesce | readconfig | readlog | repltopology | resumerole | stopserver
|
             unbind | uniqueattr }
```

Options

The options for the **idsldapexop** command are of two types.

1. General options that specify how to connect to the Directory Server. These options must be specified before extended operation-specific options.
2. Extended operation options that identify the extended operation to run.

General options

-C ***charset***

Specifies the string to the command be represented in a local character set, as specified by *charset*. Use **-C** *charset* to override the default, where string must be supplied in UTF-8. For more information about the specific *charset* values that are supported for each operating system, see Chapter 6, "Supported IANA character sets," on page 135. The supported values for *charset* are the same values that are supported for the *charset* tag that is optionally defined in version 1 LDIF files.

-d ***debuglevel***

Sets the LDAP debug level to *debuglevel* in LDAP library. This option causes the utility to generate debug output to stdout. The *debuglevel* is a bit mask that controls which output is generated with values from 1 to 65535. For information about debug levels, see Chapter 4, “Debugging levels ,” on page 131.

-D ***binddn***
Specifies the *binddn* to bind to an LDAP directory. The *binddn* variable is a string-represented value. When used with **-m DIGEST-MD5**, it specifies the authorization ID. It can be either a DN or an authorization ID string that starts with `dn:` or `u:`.

-e
Shows the LDAP library version information and then exits.

-E ***token_pw***
Specifies the token password to access a crypto device.

-G ***realm***
Specifies the realm name. When used with **-m DIGEST-MD5**, the value is passed to the server during a bind.

-h ***ldaphost***
Specifies the host name of a system where an LDAP server is running.

-help
Specifies to show help syntax.

-I
Specifies a crypto device with key storage by using PKCS11.

-K ***keyfile***
Specifies the name of the SSL or TLS key database file with the default extension of kdb. If the key database file is not in the current directory, specify the fully qualified key database file name. If a key database file name is not specified, this utility first looks for the presence of the *SSL_KEYRING* environment variable with an associated file name. If the *SSL_KEYRING* environment variable is not defined, the default keyring file is used, if present.

A default keyring file, `ldapkey.kdb`, and the associated password stash file, `ldapkey.sth`, are installed in the `etc` directory in *IDS_LDAP_HOME*.

For more information about the default key database files and default certificate authorities (CAs), see *Programming Reference* section of the IBM Security Directory Suite documentation.

If a keyring database file cannot be located, a hard coded set of default trusted certificate authority roots is used. The key database file typically contains one or more certificates of CAs that are trusted by the client. These types of *X.509* certificates are known as trusted roots. For more information about managing an SSL or TLS key database, see *Administering* section of the IBM Security Directory Suite documentation. Also, see the Security functions section.

This parameter effectively enables the **-Z** switch.

-m ***mechanism***
Specifies the SASL mechanism to use when you bind to the server. The `ldap_sasl_bind_s()` API is used for this option. The **-m** parameter is ignored if **-V 2** is set. If **-m** is not specified, simple authentication is used.

-N ***certificatename***
Specifies the label that is associated with the client certificate in the key database file. If an LDAP server is configured for server authentication only, a client certificate is not required. If the LDAP server is configured for client and server authentication, a client certificate might be required. This parameter is not required if a default certificate / private key pair is assigned as the default. If there is a single certificate / private key pair in the designated key database file, *certificatename* is not required. This parameter is ignored if **-Z** or **-K** parameter is not specified.

-p ***ldapport***
Specifies a port for the LDAP server to listen. The default LDAP port is 389. If **-p** is not specified and **-Z** is specified, the default LDAP secure port, 636, is used.

-P ***keyfilepw***

Specifies the key database password. Password is required to access the encrypted information in the key database file, which might include one or more private keys. If a password stash file is associated with the key database file, then the password is obtained from the password stash file. In this case, the **-P** parameter is not required. This parameter is ignored if **-Z** or **-K** is not specified.

-Q ***operation***

Specifies the crypto device operation with PKCS11. The *operation* variable must be assigned one of the following values:

```
0: No accelerator mode
1: Symmetric
2: Digest
3: Digest and Symmetric
4: Random
5: Random and Symmetric
6: Random and Digest
7: Random, Digest and Symmetric
```

-S ***token_label***

Specifies the token label of the crypto device.

-U ***username***

Specifies the user name. This name is required with the **-m DIGEST-MD5** parameter, and is ignored when any other mechanism is used. The value of *username* depends on what attribute the server is configured to use. It might be a `uid` or any other value that is used to locate the entry.

-v

Indicates to run in verbose mode.

-w ***passwd* | ?**

Specifies the password for authentication. Use the ? to generate a non-echoed password prompt. The password prompt option prevents showing your password when you use the **ps** command.

-x

Specifies to use FIPS mode processing. This mode is applicable with SSL or TLS only.

-X ***lib_path***

Specifies the library path of the crypto device.

-y ***proxyDN***

Specifies the DN to use for proxied authorization.

-Y

Specifies to use a secure LDAP connection by using the startTLS protocol. The -Y option is only supported when GSKit is installed.

-Z

Specifies to use a secure LDAP connection by using Secure Sockets Layer (SSL) and Transport Layer Security (TLS). The -Z option is only supported for SSL component entry, and only when GSKit is installed.

-1 sec:usec

Specifies the timeout for the `connect()` function in seconds and microseconds. The values that are provided for seconds and microseconds must be positive integers.

-?

Specifies to show the syntax format.

Extended operations option

The **-op** option identifies the extended operation to run. The following extended operations are supported.

acctStatus -d ***userDN***

Specifies the password policy account status extended operation. Directory administrator must use the **acctStatus** extended operation option to query the server to obtain the account status of an entry that contains a `userPassword` attribute. The *userDN* value that is used in the query must contain the DN of a user account. The status for the account is open, `locked`, or `expired`.

Example:

An example to query an account status for DN `cn=Bob Garcia,ou=austin,o=sample`.

```
idsldapexop -op acctStatus -d cn=Bob Garcia,ou=austin,o=sample
```

`backuprestore -action` *actionValue*

The **`backuprestore`** extended operation sends a request to the Administration Server to back up Directory Server data and configuration files or to restore from an existing backup.

Note: To initiate back up or restore requests, the Directory Server must be already configured for backup.

Where, *actionValue* must be:

```
backup: makes a backup of the Directory Server
restore: restores the Directory Server to last backup
```

Examples:

To back up a Directory Server instance remotely, issue the following command.

```
idsldapexop -h ldaphost -p admin_port -D binddn -w password
-op backuprestore -action backup
```

To restore a Directory Server instance remotely, issue the following command.

```
idsldapexop -h ldaphost -p admin_port -D binddn -w password
-op backuprestore -action restore
```

`cascrepl -action` *actionValue* `-rc` *contextDN* `[options]`

The **`cascrepl`** extended operation is for cascading control replication. When the request is sent, cascading control replication is applied to the specified server and is also passed to all replicas for the replication context. If any server in this topology is a forwarding replica, they pass the extended operation to their replicas. The operation cascades over the entire replication topology.

The *actionValue* value must be one of the following actions and is required for the extended operation.

```
-action {quiesce | unquiesce | replnow | wait}
```

`quiesce`

No further updates are accepted, except by replication.

`unquiesce`

Resume normal operation, client updates are accepted.

`replnow`

Replicate all queued changes to all the replica servers as soon as possible, regardless of schedule.

`wait`

Wait for all updates to be replicated to all replicas.

The **`-rc`** *contextDN* is a required attribute and specifies the root of the subtree.

The **`[options]`** is an optional attribute. This attribute takes the following values.

`-timeout` *secs*

Specifies the timeout period in seconds. If not present or the value is 0, the operation waits indefinitely.

Example:

To quiesce the replication context, `o=acme,c=us`, for 60 seconds, run the following command.

```
idsldapexop -op cascrepl -action quiesce \
 -rc "o=acme,c=us" -timeout 60
```

`clearlog -log` *logname*

The **`clearlog`** extended operation clears log files. The *logname* value must be one of the following logs that requires to be cleared.

```
-log {audit | bulkload | cli | slapd | idsdiradm | adminAudit | debug
     | LostAndFound | config}
```

Example:

To clear the audit log file, run the following command.

```
idsldapexop -D bindDN -w password -op clearlog -log audit
```

`controlqueue -skip` *skipvalue* `-ra` *agreementDN*

The **`controlqueue`** extended operation controls the replication queue.

The **`-skip`** *skipvalue* is a required option. The *skipvalue* variable must contain one of the following values.

```
-skip {all | change-id}
```

`all`

Indicates to skip all pending changes for an agreement.

`change-id`

Identifies the single change to be skipped. If the server is not currently replicating this change, the request fails.

The **`-ra`** *agreementDN* is a required option. The *agreementDN* value specifies the DN of a replication agreement.

Examples:

To skip all the changes queued in a replication queue, run the following **`idsldapexop`** command. For example:

```
idsldapexop -op controlqueue -skip all -ra "cn=server3,\
ibm-replicaSubentry=master1-id,ibm-replicaGroup=default,\
o=acme,c=us"
```

To skip a specific change-id in a replication queue, run the following **`idsldapexop`** command. For example:

```
idsldapexop -op controlqueue -skip 2185 -ra "cn=server3,\
ibm-replicaSubentry=master1-id,ibm-replicaGroup=default,\
o=acme,c=us"
```

`controlrepl -action` *actionvalue* `{-rc` *contextDN* `| -ra` *agreementDN* `}`

Use the **`controlrepl`** extended operation to control replication.

The **`-action`** *actionvalue* is a required option.

The *actionvalue* value specifies the action to take.

The *actionvalue* must be `suspend`, `resume`, `replnow`, or `restart`.

For example:

```
-action {suspend | resume | replnow | restart}
```

`suspend`

Specifies to suspend replication.

`resume`

Specifies to resume the suspended replication.

`replnow`

Specifies to replicate now.

restart

Specifies to restart replication.

The **`-rc`** *contextDN* option specifies a replication context DN. The action is applied to all agreement under the *contextDN* context. The **`-ra`** *agreementDN* option specifies a replication agreement DN. The action is applied on the specified replication agreement.

Example:

To suspend replication for a replication agreement, run the following **`idsldapexop`** command. For example:

```
idsldapexop -op controlrepl -action suspend -ra "cn=server3,\
ibm-replicaSubentry=master1-id,ibm-replicaGroup=default,\
o=acme,c=us"
```

`controlreplerr {[ -delete` ***failure-ID*** **`| all ] [ -retry`** ***failure-ID*** **`| all ] [ -show`** ***failure-ID*** **`]} -ra`** ***agreementDN***

Use the **`controlreplerr`** extended operation to control replication errors.

The extended operation uses the following parameters:

`-delete` ***failure-ID*** **`| all`**

Specifies to remove the failed update. To identify the update to remove, use the following options:

failure-ID

Specifies to delete only the failed update for the agreement that is identified by the failure-ID.

`all`

Specifies to delete all the failed updates for this agreement.

`-retry` ***failure-ID*** **`| all`**

Specifies to try the failed update again. To identify the update to try again, use the following options:

failure-ID

Specifies to try only the failed update again for the agreement that is identified by the failure-ID.

`all`

Specifies to try all the failed updates again for this agreement.

`-show` ***failure-ID***

Specifies to show the failed update that is identified by the failure-ID.

`-ra` ***agreementDN***

The**`-ra`** *agreementDN* specifies the DN of the replication agreement. The action is applied on the specified replication agreement.

Example:

To delete all replication errors for a replication agreement, run the following **`idsldapexop`** command. For example:

```
idsldapexop -op controlreplerr -delete all -ra "cn=server3,\
ibm-replicaSubentry=master1-id,ibm-replicaGroup=default,\
o=acme,c=us"
```

`evaluategroups -d` ***specificDN*** **`[ -a`** ***attribute value pairs...*** **`]`**

The **`evaluategroups`** extended operation identifies all groups to which a DN belongs.

The extended operation uses the following parameters:

`-d` ***specificDN***

Specifies the DN to be evaluated to determine which groups it belongs to.

`-a` ***attribute value pairs...***

Specifies a list of whitespace-separated list of attribute value pairs. Each attribute value pair is in the `attr=value` format. If the **`-a`** option is not provided, the specified DN is evaluated for static groups only.

An attribute value pair is an attribute type and attribute value that is separated by an equal sign. User attributes are required for evaluating group membership for dynamic group. When a server receives an evaluate group request with attributes, the server uses these attributes for the group evaluation.

Example:

To evaluate groups of a DN with the specified attribute value, run the following **idsldapexop** command. For example:

```
idsldapexop -op evaluategroups \
-d "cn=John Smith,ou=Austin,o=sample" \
-a departmentNumber=G8R
```

getattributes -attrType *type* -matches *value*

The **getattributes** extended operation retrieves attributes of a specified type if the criteria is met.

The extended operation uses the following parameters:

-attrType *type*

Specifies the type of the request attribute, and is a required option. The *type* value must be one of the following attribute types.

```
-attrType {operational | language_tag | attribute_cache | unique
| configuration | encryptable | encrypted}
```

-matches { true | false }

Specifies whether the list of attributes that are returned match the attribute type that is specified by the **-attrType** option.

Examples:

To get a list of all attributes that can be defined as unique attributes, run the following **idsldapexop** command. For example:

```
idsldapexop -op getattributes -attrType unique -matches true
```

To get a list of all attributes that is not defined as unique attributes, run the following **idsldapexop** command. For example:

```
idsldapexop -op getattributes -attrType unique \
 -matches false
```

getlogsize -log *logname*

The **getlogsize** extended operation retrieves file size of a log file.

The **-log** *logname* parameter specifies the log file for which file size is to be retrieved. The size of the log file, in lines, is shown on standard output. This parameter is required. The *logname* value must be one of the following log files.

```
-log {audit | bulkload | cli | slapd | idsdiradm | adminAudit
| debug | LostAndFound | config}
```

Example:

To get file size of the `slapd` log file, run the following **idsldapexop** command. For example:

```
idsldapexop -D adminDN -w adminPW -op getlogsize
-log slapd 2000 lines
```

effectpwdpolicy -d { *user DN | group DN*}

The **effectpwdpolicy** extended operation retrieves effective password policy of a user or group entry.

Example:

To get effective password policy of a user entry, run the following **idsldapexop** command. For example:

```
idsldapexop -D adminDN -w adminPW -op effectpwdpolicy \
-d cn=Bob Garcia,ou=austin,o=sample
```

getusertype

The **getusertype** extended operation returns the user type and roles that are associated with the user entry based on the bound DN.

Examples:

To get the user type and roles associated with primary administrator, run the following **idsldapexop** command. For example:

```
idsldapexop -D adminDN -w adminPW -op getusertype
```

An example output from the command:

```
User: root_administrator
Role(s) : audit_administrator directory_data_administrator
password_administrator replication_administrator
schema_administrator server_config_administrator
server_start_stop_administrator
```

To get the user type and roles associated with a local administration group member with `ReplicationAdmin` and `ServerStartStopAdmin` roles, run the following **idsldapexop** command. For example:

```
idsldapexop -D localadminDN -w localadminPW -op getusertype

User: admin_group_member
Role(s) : replication_administrator server_start_stop_administrator
```

To get the user type and roles of a user entry in the directory information tree (DIT), run the following **idsldapexop** command. For example:

```
idsldapexop -D userDN -w userPW -op getusertype

    User    : ldap_user_type
Role(s) : ldap_user_role
```

locateEntry -d *DN*| -f *file_with_DN_list* [-c]

The **locateEntry** extended operation retrieves the back-end server details for the provided DN entries. This extended operation must be run against a Proxy Server. To extract the details of a DN entry, the **-d** option is used. To extract details for a set of DN entries, use the **-f** option. The file that is passed to the **-f** option must contain a list of DN entries that you want to locate. The **[-c]** parameter specifies to run continues operation even if errors are encountered in the file.

Example:

To locate an entry on the back-end server, run the following **idsldapexop** command. For example:

```
idsldapexop -D bindDN -w bindPW -op locateEntry \
-d "cn=user,o=sample"
```

onlineBackup -path *directoryPath*

The **onlineBackup** extended operation does an online backup of the DB2® database that is associated with a Directory Server instance. The *directoryPath* value specifies the location where you want to place the backup.

Example:

To take an online backup of DB2 database that is associated with a Directory Server instance, run the following **idsldapexop** command. For example:

```
idsldapexop -D bindDN -w bindPW -op onlineBackup \
-path directoryPath
```

quiesce -rc *contextDN* [options]

The **quiesce** extended operation does quiesce or unquiesce action on the replication context.

-rc *contextDN*

This option is required and specifies the replication context DN to be quiesced or unquiesced.

[options]

The **[options]** parameter takes **-end** as value. This optional option and specifies to unquiesce the subtree. If not specified, the default is to quiesce the subtree.

Examples:

To quiesce a replication context, run the following **idsldapexop** command. For example:

```
idsldapexop -D bindDN -w bindPW -op quiesce -rc "o=sample"
```

To unquiesce a replication context, run the following **idsldapexop** command. For example:

```
idsldapexop -D bindDN -w bindPW -op quiesce -end \
-rc "o=sample"
```

readconfig -scope *scopevalue*

The **readconfig** extended operation reads the configuration file. The *scopevalue* variable takes one of the following values.

- **entire**: Indicates to read the entire configuration file again.
- **single** *entryDN attribute*: Specifies to read the specified single entry and the attribute.
- **entry** *entryDN*: Specifies to read the provided entry.
- **subtree** *entryDN*: Specifies to read the entry and the entire subtree under it.

```
-scope {entire | single entryDN attribute | entry entryDN
 | subtree entryDN}
```

Examples:

To read the entire configuration file, run the following **idsldapexop** command. For example:

```
idsldapexop -D adminDN -w adminPW -op readconfig -scope entire
```

To read an entry and attribute from configuration file, run the following **idsldapexop** command. For example:

```
idsldapexop -D adminDN -w adminPW -op quiesce -scope\
single "cn=configuration" ibm-slapdAdminPW
```

readlog -log *logname* -lines *value*

The **readlog** extended operation reads the specified number of lines from a log file.

The **-log** *logname* is a required option. The value of *logname* must be one of the following logs: `audit`, `bulkload`, `cli`, `slapd`, `idsdiradm`, `adminAudit`, debug, `LostAndFound`, and `config`.

```
-log {audit | bulkload | cli | slapd | idsdiradm | adminAudit
| debug | LostAndFound | config}
```

The **-lines** *value* is a required option. The *value* specifies the first and last lines to be read from the file or all lines. Numbering of lines starts from 0. The lines from logs are written to standard output.

```
-lines {first last | all }
```

Examples:

To read the entire `slapd` log file, run the following **idsldapexop** command. For example:

```
idsldapexop -D adminDN -w adminPW -op readlog \
-log slapd -scope all
```

To read lines that are specified by *first* and *last* variable from the `slapd` log file, run the following **idsldapexop** command. For example:

```
idsldapexop -D adminDN -w adminPW -op readlog\
-log audit -lines 10 20
```

repltopology -rc *contextDN* [options]

The **repltopology** extended operation replicates the replication topology-related entries under the specified context.

-rc *contextDN*

This option is required, and specifies the replication context DN.

[options]

The **[options]** parameter takes parameters.

-timeout *secs*

This parameter is optional and if present, specifies the timeout period in seconds. If this parameter is not present or value of *secs* is 0, the extended operation waits indefinitely.

-ra *agreementDN*

The **-ra** *agreementDN* parameter specifies the replication agreement DN. You can use this parameter to specify the replication agreement for which the replication must be run. If the **-ra** parameter is not specified, the replication is run against all the replication agreements that are defined under the context.

Example:

To replicate entries for the specific agreement under a replication context, run the following **idsldapexop** command. For example:

```
idsldapexop -D adminDN -w adminPW -op repltopology \
-rc "o=acme,c=us" -ra "cn=server3,\
ibm-replicaSubentry=master1-id,ibm-replicaGroup=default,\
o=acme,c=us" -timeout 60
```

resumerole -type *typeValue*

The **resumerole** extended operation resumes the configured role of a back-end server that is associated with a Proxy Server in a distributed directory environment.

The *typeValue* variable takes one of the following values:

all

Specifies to resume roles for all the configured back-end servers.

partition *partitionName*

Specifies to resume roles of all configured back-end servers in a partition.

server *serverName*

Specifies to resume the role of a back-end server for all partitions in which the server is configured.

serverinapartition *serverName partitionName*

Specifies to resume the role of a back-end server in the specified partition.

Example:

To resume roles for all the configured back-end servers, run the following **idsldapexop** command. For example:

```
idsldapexop -D adminDN -w adminPW -op resumerole -type all
```

stopserver

The **stopserver** extended operation stops a Directory Server instance.

Example:

To stop a Directory Server instance, run the following **idsldapexop** command. For example:

```
idsldapexop -p port -D adminDN -w adminPW -op stopserver
```

unbind {-dn *DN* | -ip *sourceIP* | -dn *DN* -ip *sourceIP* | -all}

The **unbind** extended operation disconnects connections that are based on DN, IP, DN and IP, or all connections. Connections without any operations and connections with operations on a work queue are immediately ended. If a worker thread is working on a connection, it is ended as soon as the worker completes the operation.

-dn *DN*

Issues a request to end a connection for the specified DN. This request results in the purging of all the connections that are bound on the specified DN.

-ip *sourceIP*

Issues a request to end a connection for the specified IP address. This request results in the purging of all the connections from the specified IP source.

-dn *DN* -ip *sourceIP*

Issues a request to end a connection for the specified DN and IP. This request results in the purging of all the connections that are bound by using the specified DN from the specified IP source.

-all

Issues a request to end all the connections. This request results in the purging of all the connections except for the connection from where the request originated. This parameter cannot be used with the **-dn** or **-ip** parameters.

Examples:

To unbind all connections that are associated with a specific DN, run the following **idsldapexop** command. For example:

```
idsldapexop  -D adminDN -w adminPW -op unbind \
-dn cn=john,o=sample
```

To unbind all connections origination from a specific IP, run the following **idsldapexop** command. For example:

```
idsldapexop  -D adminDN -w adminPW -op unbind \
-ip 9.182.173.43
```

To unbind all connections that are associated with a specific DN and origination from a specific IP, run the following **idsldapexop** command. For example:

```
idsldapexop  -D adminDN -w adminPW -op unbind \
-dn cn=john,o=sample -ip 9.182.173.43
```

To unbind all connections, run the following **idsldapexop** command. For example:

```
idsldapexop  -D adminDN -w adminPW -op unbind -all
```

uniqueattr -a *attributeName*

The **uniqueattr** extended operation identifies all non-unique values for an attribute. The **-a** *attributeName* parameter specifies the attribute for which all conflicting values must be listed.

Note: Duplicate values for the `binary`, `operational`, `configuration`, and `objectclass` attributes are not shown. These attributes are not supported by the extended operation for unique attributes.

The following line is added to the configuration file under the `cn=Directory,cn=RDBM Backends,cn=IBM Directory,cn=Schema,cn=Configuration` entry for this extended operation.

```
ibm-slapdPlugin:extendedop /bin/libback-rdbm.dll initUniqueAttr
```

Example:

To retrieve the non-unique values assigned to an attribute, run the following **idsldapexop** command. For example:

```
idsldapexop  -D adminDN -w adminPW -op uniqueattr -a "uid"
```

Notes

If you do not provide DN entry information, the **idsldapexop** command waits to read a list of DN entries from standard input. To exit from the command prompt, use **Ctrl+D** on UNIX systems.

Exit status

Exit status is `0` if no errors occur. If exit status is non-zero, then an error occurred. When error occurs, messages are written to the standard error.

Security functions

To use the SSL or TLS-related functions that are associated with this utility, seeChapter 3, “SSL and TLS notes,” on page 129.

See also

idsldapadd, **idsldapchangepwd**, **idsldapdelete**, **idsldapmodify**, **idsldapmodrdn**, **idsldapsearch**

idsldapmodrdn

Use the **idsldapmodrdn** command to modify the relative distinguished name (RDN) or to change the parent DN of an entry.

Description

The **idsldapmodrdn** command is an LDAP modify RDN tool. The **idsldapmodrdn** command is a command-line interface to the `ldap_rename` library call.

The **idsldapmodrdn** command opens a connection to an LDAP server, binds to the LDAP server, modifies the RDN of an entry. An entry can be read from a standard input, a file by using the **-i** option, or from a command prompt by using the **dn**, **rdn**, or **newSuperior** option.

To see syntax help for**idsldapmodrdn**, type:

```
idsldapmodrdn -?
```

Synopsis

```
idsldapmodrdn | idsldapmodrdn [-c] [-C charset] [-d debuglevel][-D binddn]
              [-E token_pw] [-f file] [-G realm] [-h ldaphost] [-i file]
              [-I] [-k] [-K keyfile] [-l] [-m mechanism] [-M] [-n]
              [-N certificatename] [-O hopcount] [-p ldapport]
              [-P keyfilepw] [-r] [-R] [-s newSuperior] [-S token_label]
              [-U username] [-v] [-V] [-w passwd | ?] [-x] [-X lib_path]
              [-y proxydn] [-Y] [-Z] [-1 sec:usec] [dn newrdn | [-i file]]
```

Options

The options to the **idsldapmodrdn** command.

-c
Specifies to run continuous operation, and do not stop processing on error. If the **-c** parameter is not specified, the command exits if an error is encountered.

-C *charset*
Specifies a string to the command be represented in a local character set, as specified by *charset*. Use **-C** *charset* to override the default, where string must be supplied in UTF-8. For information about the specific *charset* values that are supported for each operating system, seeChapter 6, "Supported IANA character sets," on page 135. The supported values for *charset* are the same values that are supported for the *charset* tag that is optionally defined in version 1 LDIF files.

-d *debuglevel*
Sets the LDAP debug level to *debuglevel* in LDAP library. This option causes the utility to generate debug output to stdout. The *debuglevel* is a bit mask that controls which output is generated with values from 1 to 65535. For information about debug levels, see Chapter 4, "Debugging levels ," on page 131.

-D *bindDN*
Specifies the *bindDN* to bind to an LDAP directory. The *bindDN* variable is a string-represented value. When used with **-m DIGEST-MD5**, it specifies the authorization ID. It can be either a DN or an authorization ID string that starts with `dn:` or `u:`.

-E *token_pw*
Specifies the token password to access a crypto device.

-f *file*
Specifies the file from which to read entry modification information.

-G *realm*
Specifies the realm name. When used with **-m DIGEST-MD5**, the value is passed to the server during a bind.

-h *ldaphost*
Specifies the host name of the system where an LDAP server is running.

-i *file*
Specifies to read entry modification information from the file instead of standard input or command line. Standard input can be supplied from a file, for example `< file`.

-I
Specifies a crypto device with key storage by using PKCS11.

-k
Specifies to send the server administration control. For more information about the server administration control, see *Programming Reference* section of the IBM Security Directory Suite documentation.

-K *keyfile*
Specifies the name of the SSL or TLS key database file with the default extension of kdb. If the key database file is not in the current directory, specify the fully qualified key database file name. If a key database file name is not specified, this utility first looks for the presence of the *SSL_KEYRING* environment variable with an associated file name. If the *SSL_KEYRING* environment variable is not defined, the default keyring file is used, if present.

A default keyring file, `ldapkey.kdb`, and the associated password stash file, `ldapkey.sth`, are installed in the `etc` directory in *IDS_LDAP_HOME*. For more information about the default key database files and default certificate authorities (CAs), see *Programming Reference* section of the IBM Security Directory Suite documentation.

If a keyring database file cannot be located, a hard coded set of default trusted certificate authority roots is used. The key database file typically contains one or more certificates of CAs that are trusted by the client. These types of *X.509* certificates are known as trusted roots. For more information about managing an SSL or TLS key database, see *Administering* section of the IBM Security Directory Suite documentation. Also, see the Security functions section.

This parameter effectively enables the **-Z** switch.

-l

Specifies not to replicate the entry.

This parameter sends the Do not replication control to the server. For information about this control, see *Programming Reference* section of the IBM Security Directory Suite documentation.

-m ***mechanism***

Specifies the SASL mechanism to use when you bind to the server. The `ldap_sasl_bind_s()` API is used for this option. The **-m** parameter is ignored if **-V 2** is set. If **-m** is not specified, simple authentication is used.

-M

Specifies to manage referral objects as regular entries.

-n

Specifies to demonstrate the action of the operation without actually doing it.

Tip: The **-n** option with the **-v** option is useful when you debug any related problem.

-N ***certificatename***

Specifies the label that is associated with the client certificate in the key database file. If an LDAP server is configured to use server authentication only, a client certificate is not required. If the LDAP server is configured to use client and server authentication, a client certificate might be required. This parameter is not required if a default certificate / private key pair is assigned as the default. Similarly, *certificatename* is not required if there is a single certificate / private key pair in the designated key database file. This parameter is ignored if **-Z** or **-K** parameter is not specified.

-O ***hopcount***

Specify *hopcount* to set the maximum number of hops that the client library takes when it chases the referrals. The default hop count is 10.

-p ***ldapport***

Specifies a port for the LDAP server to listen. The default LDAP port is 389. If **-p** is not specified and **-Z** is specified, the default LDAP secure port, 636, is used.

-P ***keyfilepw***

Specifies the key database password. Password is required to access the encrypted information in the key database file, which might include one or more private keys. If a password stash file is associated with the key database file, then the password is obtained from the password stash file. In this case, the **-P** parameter is not required. This parameter is ignored if **-Z** or **-K** is not specified.

-Q ***operation***

Specifies the crypto device operation with PKCS11. The *operation* variable must be assigned one of the following values:

```
0: No accelerator mode
1: Symmetric
2: Digest
3: Digest and Symmetric
4: Random
5: Random and Symmetric
6: Random and Digest
7: Random, Digest and Symmetric
```

-r

Removes the old RDN value from an entry. The default action is to keep the old value.

-R

Specifies not to chase referrals automatically.

-s ***newSuperior***

Specifies the DN of the new parent entry under which the renamed RDN is relocated. The *newSuperior* value can be a zero-length string, for example `-s ""`.

-S ***token_label***
Specifies the token label of the crypto device.

-U ***username***
Specifies the user name. This name is required with **-m DIGEST-MD5**, and is ignored when any other mechanism is used. The value of *username* depends on what attribute the server is configured to use. It might be a `uid` or any other value that is used to locate the entry.

-v
Indicates to run in verbose mode.

-V ***version***
Specifies the LDAP protocol version to use. By default, an LDAP V3 connection is established. To explicitly select LDAP V3, specify **-V 3**. To run as an LDAP V2 application, specify **-V 2**.

-w ***passwd*** **| ?**
Specifies the password for authentication. Use the `?` to generate a non-echoed password prompt. In UNIX, use backslash **\?** to avoid matching single character file names.

-x
Specifies to use FIPS mode processing. This mode is applicable with SSL or TLS only.

-X ***lib_path***
Specifies the library path of the crypto device.

-y ***proxydn***
Specifies the DN to use for proxied authorization.

-Y
Specifies to use a secure LDAP connection by using the startTLS protocol. The `-Y` option is only supported when GSKit is installed.

-Z
Specifies to use a secure LDAP connection by using Secure Sockets Layer (SSL) and Transport Layer Security (TLS). The `-Z` option is only supported for SSL component entry, and only when GSKit is installed.

-1 sec:usec
Specifies the timeout for the `connect()` function in seconds and microseconds. The values that are provided for seconds and microseconds must be positive integers.

dn newrdn
Specifies the RDN value to substitute for the existing RDN value. For more information, see the Input format section.

-?
Specifies to show the syntax format.

Input format

If the command-line arguments *dn* and *newrdn* are provided, *newrdn* replaces the RDN of the entry that is specified by the DN, *dn*. Otherwise, the contents of file or standard input consist of one or more entries of DN and RDN.

Notes

If you do not provide entry information by using the file with **-i** or from command line by using *dn* and *newrdn* arguments, **idsldapmodrdn** waits to read entries from standard input. To exit from the command prompt, use **Ctrl+D** on UNIX systems.

Exit status

Exit status is `0` if no errors occur. If exit status is non-zero, then an error occurred. When error occurs, messages are written to the standard error.

Security functions

To use the SSL or TLS-related functions that are associated with this utility, seeChapter 3, “SSL and TLS notes,” on page 129.

See also

idsldapadd, **idsldapchangepwd**, **idsldapdelete**, **idsldapexop**, **idsldapmodify**, **idsldapsearch**

Examples

Example 1:

Consider a file `entrymods` contains the following entries:

```
cn=Modify Me, o=University of Life, c=US
cn=The New Me
```

To change the `cn=Modify Me` RDN from `Modify Me` to `The New Me`, run the following command:

```
idsldapmodrdn -r -i entrymods
```

After you run the command, the `cn=Modify Me` RDN is removed.

To change the RDN and to move the entry under a different subtree, run the following command:

```
idsldapmodrdn –s "o=sample" "cn=Modify Me,o=University of Life,c=US"
          "cn=The New Me"
```

This command changes the RDN from `cn=Modify Me` to `cn=The New Me`. The entry is also moved from the `o=University of Life, c=US` subtree to `o=sample`.

Note: The `o=sample` entry must exist for the operation to be successful.

idsldapreplcfg

Use the **idsldapreplcfg** configuration tool to configure various replication topologies for the Directory Server. This tool simplifies the process of setting up replication and reduces errors that might occur when you set up replication by using the graphical user interface or the `ldif` file.

Description

The **idsldapreplcfg** tool supports configuration of replication topologies with "simple bind" and "simple bind over SSL" between the supplier and consumer. You cannot use the tool for configuring replication over SSL by using certificate based authentication.

Synopsis

```
idsldapreplcfg [ [-C charset] [-d level] [-e] [-s] [-v] [-Z] [-? | -help] ]
      -topo topologyName [topology-specific options] -add serverType [add server-specific
options]
```

Options

-C *charset*

Specifies the character set name as registered with IANA (Internet Assigned Numbers Authority), which you want to use.

-d *level*

Sets the specified debugging level in the LDAP library.

-e

Displays the LDAP library version information and exits.

-s
Specifies the DN of suffix or subtree entry that you want to configure as the replication context.

-v
Shows verbose output.

-Z
Specifies to use a secure LDAP connection by using Secure Sockets Layer (SSL) and Transport Layer Security (TLS). The -Z option is only supported for SSL component entry, and only when GSKit is installed.

-? | -help
Shows the usage.

-topo*topologyName*

Specifies one of the following replication topologies that you want to set up:

PP
Sets up a peer-to-peer replication topology.

MR
Sets up a master-replica replication topology.

MFR
Sets up a master-forwarding-replica replication topology.

GG
Sets up a gateway replication topology.

To get help for a specific topology name, enter the following command at the command line:

```
idsldapreplcfg -topo topologyName -help
```

-add *serverType*

Specifies one of the following types of servers to be added to the existing replication topology:

PEER
Adds a peer server to the existing replication topology.

REPL
Adds a replica server to the existing replication topology.

FWDR
Adds a forwarder server to the existing replication topology.

GATW
Adds a gateway server to the existing replication topology.

To get help for an add server action, enter the following command at the command line:

```
idsldapreplcfg -add serverType -help
```

To unconfigure the replication, you must delete all the replication topology entries in the following sequence:

1. `ibm-replicationAgreement` entries
2. `ibm-replicaSubentry` entries that correspond to other peer server IDs
3. `ibm-replicaSubentry` entry that corresponds to its own server ID
4. `ibm-replicaGroup` entry

idsldapsearch

Use the **idsldapsearch** command to search existing entries from an LDAP server that match a filter.

Description

The **idsldapsearch** is a command-line interface to the `ldap_search` library call.

The **idsldapsearch** command opens a connection to an LDAP server, binds to the LDAP server, and does a search by using the filter. The filter must conform to the string representation for LDAP filters. For information about filters that are used in `ldap_search`, see *Programming Reference* section of the IBM Security Directory Suite documentation.

If **idsldapsearch** finds one or more entries that match the filter, the attributes and its values that are specified by *attributes* are retrieved. The entries and attribute values are printed to a standard output. If no *attributes* are listed, all attributes are returned.

To see syntax help for**idsldapsearch**, type `idsldapsearch -?`.

Note:

- The size limit for search filter is set at 4 KB in the `idsldapsearch.c` file. The **idsldapsearch** utility rejects any filter size that is larger than 4 KB. If you want to change `idsldapsearch.c` to handle a filter larger than 4 KB then change the following line in `idsldapsearch.c`. For example, change

  ```
  #define FILTERSIZE 4096
  ```

  to

  ```
  #define FILTERSIZE 16000
  ```

  You must recompile `idsldapsearch.c` for these changes to take effect. However, an altered version of **idsldapsearch** is not supported.
- Entries under `cn=configuration` are not in directory information tree (DIT). Therefore, entries under `cn=configuration` are not returned in the search results for null based searches.

Synopsis

```
ldapsearch [-b basedn] [options] filter [attributes...]
```

where,

basedn: Specifies the base DN for a search. It is optional if the `LDAP_BASEDN` variable is set in the environment.
filter: Specifies an LDAP search filter.
attributes: Specifies a list of whitespace-separated attributes to retrieve, if no attribute list is specified all attributes are retrieved.

Options

The options to the **idsldapsearch** command.

-a *deref*

Specifies how to dereference aliases. The value of *deref* must be:

`never`: specifies that aliases are never dereferenced
`always`: specifies that aliases are always dereferenced
`search`: specifies that aliases are dereferenced for searching
`find`: specifies that aliases are dereferenced only when used to locate the base object for the search

The default behavior of *deref* is to never dereference aliases.

-A

Specifies to retrieve attributes only (no values). This option is useful when you want to see whether an attribute is present in an entry and is not interested in the specific values.

-b *searchbase*

Specifies to use *searchbase* as the starting point for the search, instead of the default. If **-b** is not specified, this utility examines the LDAP_BASEDN environment variable for a *searchbase* definition. If neither is set, the default base is set to `""`, which is a null search. To see all entries under a subtree, the search requires a **-s** subtree option. Otherwise, an error message is returned. Null based search requests use considerable system resource.

-B

Specifies not to suppress non-ASCII values from showing in output. This option is useful when you use values that contain character sets such as `ISO-8859.1`. This option is implied by the **-L** option.

-c *pattern*

Runs a persistent search. The pattern format must be `ps:changeType[:changesOnly[:entryChangeControls]]`, where `changeType` can be operations such as `add`, `delete`, `modify`, `moddn`, and `any`. The `changesOnly` and `entryChangeControls` parameters can be set to TRUE or FALSE.

Note: When alias dereferencing option is `find`, then only the search base object is dereferenced if it is an alias. This means that even if it is a one-level or subtree search, the subordinate alias entries under the base are not expected to be dereferenced. If a persistent search reports changed entries, and the entry is an alias then it is dereferenced even though it is subordinate to the search base.

-C *charset*

Specifies a string to the command be represented in a local character set, as specified by *charset*. String input includes the filter, the bind DN, and the base DN. When search result is returned, **idsldapsearch** converts data that is received from the LDAP server to the specified character set. Also, if the **-C** option and the **-L** option are both specified, parameter is assumed to be in the specified character set. However, the output from **idsldapsearch** is always preserved in its UTF-8 representation, or a base-64 encoded representation of the data when non-printable characters are detected. The reason for the conversion is because standard LDIF files contain UTF-8 (or base-64 encoded UTF-8) representations of string data.

Use **-C** *charset* to override the default, where string must be supplied in UTF-8. For information about the specific *charset* values that are supported for each operating system, see Chapter 6, "Supported IANA character sets," on page 135. The supported values for *charset* are the same values that are supported for the *charset* tag that is optionally defined in version 1 LDIF files.

-d *debuglevel*

Sets the LDAP debug level to *debuglevel* in LDAP library. This option causes the utility to generate debug output to stdout. The *debuglevel* is a bit mask that controls which output is generated with values from 1 to 65535. For information about debug levels, see Chapter 4, "Debugging levels ," on page 131.

-D *bindDN*

Specifies the *bindDN* to bind to an LDAP directory. The *bindDN* variable is a string-represented value. When used with **-m DIGEST-MD5**, it specifies the authorization ID. It can be either a DN or an authorization ID string that starts with *dn:* or *u:*.

-e

Specifies to show the LDAP library version information.

-E *token_pw*

Specifies the token password to access a crypto device.

-f *file*

Specifies to run searches by using the filters in the *file* file. For the filter, %s must be substituted.

-F *sep*

Specifies to use *sep* as the field separator between attribute names and values. The default separator is =, unless **-L** is specified, in which case this option is ignored.

-g ***before:after:index:count* | *before:after:value***

The *before* and *after* values are the number of entries around *index*, *count* is the content count, and *value* is the assertion value for the primary sort key.

-G ***realm***

Specifies the realm name. When used with **-m DIGEST-MD5**, the value is passed to the server during a bind.

-h ***ldaphost***

Specifies the host name of the system where an LDAP server is running.

-i ***file***

Specifies to read a series of lines from *file*, and to run one LDAP search for each line. In this option, the filter that is provided to the command is treated as a pattern, where the first occurrence of %s is replaced with a line from file. If file is a single - character, then the lines are read from standard input.

For example, in this example, `idsldapsearch -V3 -v -b "o=sample" -D "cn=admin" -w ldap -i filter.input %s dn`, the `filter.input` file might contain the following filter information.

```
(cn=*Z)
(cn=*Z*)
(cn=Z*)
(cn=*Z*)
(cn~=A)
(cn>=A)
(cn<=B)
```

Note: Each filter must be specified on a separate line.

In the example, the command runs a search on `o=sample` for each of the filters that begin with `cn=*Z`. When the search is complete, another search begins for the next filter `cn=*Z*`, and then the next filter, until the search for the last filter `cn<=B` is completed.

Note: The **-i** *file* option replaces the **-f** *file* option. The **-f** option is still supported, although it is deprecated.

-I

Specifies a crypto device with key storage by using PKCS11.

-j ***limit***

Specifies the maximum number of values that can be returned for an attribute within an entry. The default value is 0, which means unlimited.

-J ***limit***

Specifies the maximum number of values that can be returned for an attribute within an entry. The default value is 0, which means unlimited.

-k

Specifies to send the server administration control. For more information about this control, see *Programming Reference* section of the IBM Security Directory Suite documentation.

-K ***keyfile***

Specifies the name of the SSL or TLS key database file with the default extension of kdb. If the key database file is not in the current directory, specify the fully qualified key database file name. If a key database file name is not specified, this utility first looks for the presence of the *SSL_KEYRING* environment variable with an associated file name. If the *SSL_KEYRING* environment variable is not defined, the default keyring file is used, if present.

A default keyring file, `ldapkey.kdb`, and the associated password stash file, `ldapkey.sth`, are installed in the `etc` directory in *IDS_LDAP_HOME*. For information about the default key database files and default certificate authorities (CAs), see *Programming Reference* section of the IBM Security Directory Suite documentation.

If a keyring database file cannot be located, a hard coded set of default trusted certificate authority roots is used. The key database file typically contains one or more certificates of CAs that are trusted by the client. These types of *X.509* certificates are known as trusted roots. For information about managing an SSL or TLS key database, see *Administering* section of the IBM Security Directory Suite documentation. Also, see the Security functions section.

This parameter effectively enables the **-Z** switch.

-l ***timelimit***
Specifies to wait at most *timelimit* seconds for a search to complete.

-L
Specifies to show search results in LDIF format. This option activates the **-B** option, and causes the **-F** option to be ignored.

-m ***mechanism***
Specifies the SASL mechanism to use when you bind to the server. The `ldap_sasl_bind_s()` API is used for this option. The **-m** parameter is ignored if **-V 2** is set. If **-m** is not specified, simple authentication is used.

-M
Specifies to manage referral objects as regular entries.

-N ***certificatename***
Specifies the label that is associated with the client certificate in the key database file. If an LDAP server is configured to use server authentication only, a client certificate is not required. If the LDAP server is configured to use client and server authentication, a client certificate might be required. This parameter is not required if a default certificate / private key pair is assigned as the default. Similarly, *certificatename* is not required if there is a single certificate / private key pair in the designated key database file. This parameter is ignored if **-Z** or **-K** parameter is not specified.

-o ***attr_type***
Specifies an attribute to use for sort criteria of search results, you can use the **-o** parameter. You can use multiple **-o** parameters to further define the sort order. In the example, the search results are sorted first by *sn* and then by *givenname*. The *givenname* values are sorted in reverse (descending) order, which is specified by the prefixed minus sign `(-)`.

```
-o sn -o -givenname
```

The syntax of the sort parameter is

```
[-]attribute_name [:matching rule OID]
```

where,

attribute_name is the name of the attribute you want to sort
matching rule OID is the optional OID of a matching rule that you want to use for sorting
- minus sigh indicates that the result must be ordered in reverse order
The criticality for this option is always critical

By default, the **idsldapsearch** operation does not return result in the sorted order.

This option sends the Sorted search results control to the LDAP server. For information about sorted search results control, see *Programming Reference* section of the IBM Security Directory Suite documentation.

-O ***maxhops***
Specify *maxhops* to set the maximum number of hops that the client library takes when it chases the referrals. The default hop count is 10.

-p ***ldapport***
Specifies a port for the LDAP server to listen. The default LDAP port is 389. If **-p** is not specified and **-Z** is specified, the default LDAP secure port, 636, is used.

-P ***keyfilepw***
Specifies the key database password. Password is required to access the encrypted information in the key database file, which might include one or more private keys. If a password stash file is associated with the key database file, then the password is obtained from the password stash file. In this case, the **-P** parameter is not required. This parameter is ignored if **-Z** or **-K** is not specified.

-q *pagesize*

Specifies the page size for search results. To set page size for results, use the two parameters **-q** (query page size), and **-T** (time between searches in seconds).

Use the example values to return a page of 25 entries at a time, every 15 seconds until all the results for the search is returned.

```
-q 25 -T 15
```

The **idsldapsearch** client handles all connection continuation for each paged result request for the life of the search operation.

If the **-v** parameter is specified, **idsldapsearch** lists how many entries are returned.

You can provide multiple **-q** parameters to specify different page sizes throughout the life of a single search operation. Use the example values to specify that the first page is of 15 entries, the second page of 20 entries, and the third parameter to end the paged result.

```
-q 15 -q 20 -q 0
```

To specify the first page is of 15 entries, and the rest of the pages are of 20 entries, continuing with the last specified **-q** value until the search operation completes, use the example values.

```
-q 15 -q 20
```

By default, the **idsldapsearch** operation returns all entries in a single request. No paging is done for the default **idsldapsearch** operation.

This option sends the Paged search results control. For information about paged search results control, see *Programming Reference* section of the IBM Security Directory Suite documentation.

-Q *operation*

Specifies the crypto device operation with PKCS11. The *operation* variable must be assigned one of the following values:

```
0: No accelerator mode
1: Symmetric
2: Digest
3: Digest and Symmetric
4: Random
5: Random and Symmetric
6: Random and Digest
7: Random, Digest and Symmetric
```

-r

Specified to return deleted entries.

-R

Specifies not to chase referrals automatically.

-s *scope*

Specifies the scope of the search. The *scope* variable must be assigned one of the following values:

- `base`: specifies a base object search
- `one`: specifies a one-level search
- `sub`: specifies a subtree search

The default scope is sub.

-S *token_label*

Specifies the token label of the crypto device.

-t

Specifies to write retrieved values to a set of temporary files. This option is useful for dealing with non-ASCII values such as *jpegPhoto* or *audio*.

-T *seconds*

Specifies the time in seconds between searches. The **-T** option is only supported when the **-q** option is specified.

-U *username*

Specifies the user name. This name is required with **-m DIGEST-MD5**, and is ignored when any other mechanism is used. The value of *username* depends on what attribute the server is configured to use. It might be a `uid` or any other value that is used to locate the entry.

-v

Indicates to run in verbose mode.

-V

Specifies the LDAP protocol version to use. By default, an LDAP V3 connection is established. To explicitly select LDAP V3, specify **-V 3**. To run as an LDAP V2 application, specify **-V 2**.

-w *passwd* | ?

Specifies the password for authentication. Use the ? prompt to generate a non-echoed password prompt. In UNIX, use backslash **\?** to avoid matching single character file names. If you use the password prompt, it prevents your password from being visible through the **ps** command.

-x

Specifies to use FIPS mode processing. This mode is applicable with SSL or TLS only.

-X *lib_path*

Specifies the library path of the crypto device.

-y *proxydn*

Specifies the DN to use for proxied authorization.

-Y

Specifies to use a secure LDAP connection by using the startTLS protocol. The `-Y` option is only supported when GSKit is installed.

-z *sizelimit*

Specifies to limit the search results to at most *sizelimit* entries. This option makes it possible to place an upper bound on the number of entries that are returned for a search operation.

-Z

Specifies to use a secure LDAP connection by using Secure Sockets Layer (SSL) and Transport Layer Security (TLS). The `-Z` option is only supported for SSL component entry, and only when GSKit is installed.

-1 sec:usec

Specifies the timeout for the `connect()` function in seconds and microseconds. The values that are provided for seconds and microseconds must be positive integers.

-9 p

Sets criticality for paging to false. The search is handled without paging.

-9 s

Sets criticality for sorting to false. The search is handled without sorting.

filter

Specifies a string representation of the filter to apply in the search. Simple filters can be specified as *attributetype*=`attributevalue`. More complex filters are specified by using a prefix notation according to the following *Backus Naur Form* (BNF):

```
<filter> ::='('<filtercomp>')'
<filtercomp> ::= <and>|<or>|<not>|<simple>
<and> ::= '&' <filterlist>
<or> ::= '|' <filterlist>
<not> ::= '!' <filter>
<filterlist> ::= <filter>|<filter><filtertype>
<simple> ::= <attributetype><filtertype>
<attributevalue>
<filtertype> ::= '='|'~='|'<='|'>='
```

The ~= construct specifies an approximate matching. The representation for *attributetype* and *attributevalue* are as described in "RFC 2252, LDAP V3 Attribute Syntax Definitions". In addition, *attributevalue* can be a single * to achieve an attribute existence test, or can contain text and asterisks (*) interspersed to achieve substring matching.

For example, the filter *mail*=`*` finds any entries that have a mail attribute. The filter *mail*=`*@student.of.life.edu` finds any entries that have a mail attribute that ends in the specified string. To put parentheses in a filter, escape them with a backslash (\) character.

Note: A filter like `cn=Bob *`, where there is a space between Bob and the asterisk (*), matches "Bob Carter" but not "Bobby Carter" in IBM Directory. The space between "Bob" and the wildcard character (*) affects the outcome of a search by using filters.

For information about the complete description of allowed filters, see "RFC 2254, A String Representation of LDAP Search Filters".

attrs

Specifies a whitespace-separated list of attribute type names to return for each entry that matches the search filter. Individual attribute type names might be specified. Additionally, the following special notations can be used:

*****

Indicates to return all attribute types other than operational attributes.

1.1

Specifies to return no attributes and requests the search to return only the matching distinguished names.

+

Indicates to return the operational attributes.

+ibmaci

Indicates to return the access control related operational attributes.

+ibmentry

Indicates to return the operational attributes that every entry contains, such as *creatorsName*, *create_Timestamp*, and *modifiersname* to name a few.

+ibmrepl

Indicates to return operational attributes that are related to replication.

+ibmpwdpolicy

Indicates to return operational attributes that are related to password policy.

++

Indicates to return ALL operational attributes, even attributes considered expensive to return such as *ibm-allGroups* and *ibm-replicationPendingChanges*.

++ibmaci

Includes ALL access control related operational attributes.

++ibmentry

Includes ALL operational attributes that every entry contains such as *numsubordinates* and *ibm-entryChecksum*.

++ibmrepl

Includes ALL operational attributes that are related to replication.

++ibmpwdpolicy

Includes ALL operational attributes that are related to password policy.

-?

Specifies to show the syntax format.

Output format

If one or more entries are found, each entry is written to standard output in the following form.

```
Distinguished Name (DN)

attributename=value

attributename=value

attributename=value
```

```
…
```

Multiple entries are separated with a single blank line. If the **-F** option is used to specify a separator character, then this separator is used instead of the = character. If the **-t** option is used, the name of a temporary file is used in place of the actual value. If the **-A** option is used, only the *attributename* part is written.

Exit status

Exit status is `0` if no errors occur. If exit status is non-zero, then an error occurred. When error occurs, messages are written to the standard error.

Security functions

To use the SSL or TLS-related functions that are associated with this utility, seeChapter 3, “SSL and TLS notes,” on page 129.

See also

idsldapadd, **idsldapchangepwd**, **idsldapdelete**, **idsldapexop**, **idsldapmodify**, **idsldapmodrdn**

Examples

Some examples of the **idsldapsearch** command and their search results.

Example 1:

```
 idsldapsearch "cn=john doe" cn telephoneNumber
```

This command runs a subtree search by using the default search base for entries with a `commonName`, `cn`, of `john doe`. The `commonName` and `telephoneNumber` values are retrieved and printed to standard output. An example output when two entries are found.

```
cn=John E Doe, ou="College of Literature, Science, and the Arts",
ou=Students, ou=People, o=University of Higher Learning, c=US

cn=John Doe

cn=John Edward Doe

cn=John E Doe 1

cn=John E Doe

telephoneNumber=+1 313 555-5432


cn=John B Doe, ou=Information Technology Division,
ou=Faculty and Staff, ou=People, o=University of Higher Learning, c=US

cn=John Doe

cn=John B Doe 1

cn=John B Doe

telephoneNumber=+1 313 555-1111
```

Example 2:

```
idsldapsearch -t "uid=jed" jpegPhoto audio
```

This command runs a subtree search by using the default search base for entries with user ID, *uid*, of `jed`. The *jpegPhoto* and *audio* values are retrieved and written to temporary files. An example output when one entry with one value for each of the requested attributes is found.

```
cn=John E Doe, ou=Information Technology Division,
ou=Faculty and Staff,
ou=People, o=University of Higher Learning, c=US
audio=/tmp/idsldapsearch-audio-a19924
jpegPhoto=/tmp/idsldapsearch-jpegPhoto-a19924
```

Example 3:

```
idsldapsearch -L -s one -b "c=US" "o=university*" o description
```

This command runs a one-level search at the c=US level for all organizations whose *organizationName*, *o*, begins with university. With the **-L** option, search result is returned in the LDIF format. The *organizationName* and *description* attribute values are retrieved and printed to standard output, resulting in output that is shown in the example.

```
dn: o=University of Alaska Fairbanks, c=US
o: University of Alaska Fairbanks
description: Preparing Alaska for a brave new tomorrow
description: leaf node only

dn: o=University of Colorado at Boulder, c=US
o: University of Colorado at Boulder
description: No personnel information
description: Institution of education and research

dn: o=University of Colorado at Denver, c=US
o: University of Colorado at Denver
o: UCD
o: CU/Denver
o: CU-Denver
description: Institute for Higher Learning and Research

dn: o=University of Florida, c=US
o: University of Florida
o: UFl
description: Shaper of young minds

 …
```

Example 4:

```
idsldapsearch -b "o=sample" -o ibm-slapdDN "objectclass=person" ibm-slapdDN
```

This command runs a subtree level search at the `o=sample` level for all persons. When this special attribute is used for sorted searches, the search results are sorted by the string representation of the distinguished name (DN). The output might look as shown in the example.

```
cn=Al Edwards,ou=Widget Division,ou=Austin,o=sample

cn=Al Garcia,ou=Home Entertainment,ou=Austin,o=sample

cn=Amy Nguyen,ou=In Flight Systems,ou=Austin,o=sample

cn=Arthur Edwards,ou=Widget Division,ou=Austin,o=sample

cn=Becky Garcia,ou=In Flight Systems,ou=Austin,o=sample

cn=Ben Catu,ou=In Flight Systems,ou=Austin,o=sample

cn=Ben Garcia Jr,ou=Home Entertainment,ou=Austin,o=sample

cn=Bill Keller Jr.,ou=In Flight Systems,ou=Austin,o=sample

cn=Bob Campbell,ou=In Flight Systems,ou=Austin,o=sample
```

Example 5:

```
idsldapsearch -b "o=sample" -s base "objectclass=*" numSubordinates
```

This command runs a one-level search at the `o=sample` level and returns the entries for the one-level search. The count that is returned does not take into account whether the bound client has authority to read any of the entries that are included in the count. The count considers entry that contains the value. If the LDAP server is loaded with entries from the example file, `sample.ldif`, then the command with the *numSubordinates* attribute might return output as shown in the example.

```
o=sample
numSubordinates=2
```

Example 6:

The following examples explain the usage of **–c** used to run a persistent search.

```
idsldapsearch -D adminDN -w adminPW –b o=sample –c ps:delete:false:true \
objectclass=*
```

The command runs a search on the `o=sample` suffix and returns the entries like a normal search. After the entries are returned, the connection stays open. Any delete operations that happen after this point triggers an update notification and is sent to the client.

```
idsldapsearch -D adminDN -w adminPW –s base –b o=sample –c ps:modify \
objectclass=*
```

The search command returns modify changes to the `o=sample` entry only. The whole entry is returned whenever there is any change in the entry. However, the entry is not returned in the initial search.

Example 7:

The following example shows all password policy attributes for an entry.

```
idsldapsearch -s base -D adminDN -w adminPW -b "uid=user1,cn=users,o=sample"\
 "objectclass=*" +ibmpwdpolicy
```

Example 8:

Binary values are not searchable. You can search on an attribute that contains binary data and the entries with that attribute are returned. However, the binary data itself is not returned nor is it searchable. The two attributes, *userPassword* and *secretKey*, are unique in that they do not have a binary syntax. The data strings for the two attributes are stored as binary syntax. Therefore, the values for these two attributes are also not searchable. For instance, a search on the *userPassword* attribute returns entries that have the attribute *userPassword*.

```
idsldapsearch -h hostname -D adminDN -w adminPW -b subtree \
"(userpassword=*)"
```

However, a search on *userPassword* =`secret` as fails.

```
idsldapsearch -h hostname -D adminDN -w adminPW -b subtree \
"(userpassword=secret)"
```

idsldaptrace

Use the **idsldaptrace** command to start or stop server trace.

Description

The **idsldaptrace** command is an administration trace utility. You can use the **idsldaptrace** command to dynamically start or stop trace against a Directory Server. You can also use this command to set the message level and to specify the file name to which you want to redirect the output. If you want to use the LDAP trace facility, **ldtrc**, options, you must use (`--`) before the **ldtrc** options. The **idsldaptrace** command is used in conjunction with IBM support to solve specific problems.

To see syntax help for **idsldaptrace**, type `idsldaptrace -?`.

Important:

1. The **idsldaptrace** command supports only the simple bind mechanism. You can use the **idsldaptrace** command with SSL or TLS.
2. Only the primary directory administrator can run this command.
3. The **idsldaptrace** command uses system resources and affects the performance of the Directory Servers.
4. If the **idsldaptrace** command is run against a non-default port, other than 389, of a server, then the **-a** and **-p** parameters must be specified. That is, both Directory Server port and administration server port must be specified.

Synopsis

```
idsldaptrace | idsldaptrace [-a port -l [on|off|clr|chg|info|dump] --[ldtrc options]
             -d debuglevel -D adminDn -E token_pw -h hostname [-I] -K keyfile
             -m debugLevel -N key_name -o debugFile -p port -P key_pw
             -S token_label -t [start|stop] -v -w adminPW|? -x -X lib_path
             -Z -1 sec:usec] -?
```

Options

The options to the **idsldaptrace** command.

-a *port*
Specifies a port number for the Administration Server, **idsdiradm**, to listen. The default port is 3538. If this port number is not specified and **-Z** is specified, the default administration server secure port, 3539, is used.

-d *debuglevel*
Specifies to debug the program.

-D *adminDN*
Specifies the DN to bind to an LDAP Directory Server. The *adminDN* variable is a string-represented value.

-E *token_pw*
Specifies the token password to access a crypto device.

-h *ldaphost*

Specifies the host name of the system where an LDAP server and the Administration Server are running.

-I

Specifies a crypto device with key storage by using PKCS11.

-K *keyfile*

Specifies the name of the SSL or TLS key database file with the default extension of kdb. If the key database file is not in the current directory, specify the fully qualified key database file name. If a key database file name is not specified, this utility first looks for the presence of the *SSL_KEYRING* environment variable with an associated file name. If the *SSL_KEYRING* environment variable is not defined, the default keyring file is used, if present.

A default keyring file, `ldapkey.kdb`, and the associated password stash file, `ldapkey.sth`, are installed in the `etc` directory in *IDS_LDAP_HOME*.

For information about the default key database files and default certificate authorities (CAs), see the *Programming Reference* section of the IBM Security Directory Suite documentation.

If a keyring database file cannot be located, a hard coded set of default trusted certificate authority roots is used. The key database file typically contains one or more certificates of CAs that are trusted by the client. These types of *X.509* certificates are known as trusted roots. For information about managing an SSL or TLS key database, see *Administering* section in the IBM Security Directory Suite documentation. Also, see the Security functions section.

This parameter effectively enables the **-Z** switch.

-l [on|off|clr|chg|info|dump] --[ldtrc options]

on

Activates the tracing facility. You can specify any of the following **ldtrc** options preceded by (`--`) symbol.

- `[-m <mask>]` where, `<mask> = <products>.<events>.<components>.<classes>.<functions>`
- `[-p <pid>[.<tid>]]`: traces only the specified process or thread
- `[-c <cpid>]`: traces only the specified companion process
- `[-e <maxSeverErrors>]`: stops tracing after the maximum number of server errors, `maxSevereErrors`, is reached
- `[-s | -f <fileName>]`: sends the output to shared memory or a file
- `[-l [<bufferSize>] | -i [<bufferSize>]]`: specifies to retain the last or the initial records, the default buffer size is 1M
- `[-this <thisPointer>]`: traces only the specified object
- `[-perf]`: traces only performance records

Remember: The tracing facility must be on, to trace the server data.

off

Deactivates the tracing facility.

clr

Clears the existing trace buffer.

chg

Changes the values for the following **ldtrc** options. The trace must be active before you can use the `chg` option.

- `[-m <mask>]` where, `<mask> = <products>.<events>.<components>.<classes>.<functions>`
- `[-p <pid>[.<tid>]]`: traces only the specified process or thread
- `[-c <cpid>]`: traces only the specified companion process

- [-e <maxSeverErrors>]: stops tracing after the maximum number of server errors, maxSevereErrors, is reached
- [-this <thisPointer>]: traces only the specified object

info

Gets information about the trace. You must specify the source file, which can be either a binary trace file or trace buffer and a destination file. The following example shows the information that the info parameter contains:

```
sdsva.example.com> sds server_tools ldtrc info
Trace Version           :     1.00
Op. System              :   Linux2
Op. Sys. Version        :      2.2
H/W Platform            :      X86

Mask                    : *.*.*.*.*.*
pid.tid to trace        : all
cpid    to trace        : all
this pointer to trace   : all
Treat this rc as sys err: none
Max severe errors       : 1
Max record size         : 32768 bytes
Trace destination       : shared memory
Records to keep         : last
Trace buffer size       : 1048576 bytes
Trace data pointer check: no
```

dump

Dumps the trace information to a file. The trace information includes process flow data and server debug messages. You can specify the name of the destination file where you want to dump the trace.

The default location for the file is CustomOut.

Note: The trace dump file contains binary **ldtrc** data that must be formatted with the **ldtrc format** command.

-m *debuglevel*

Sets the LDAP debug level to *debuglevel* in LDAP library. This option causes the utility to generate debug output to stdout. The *debuglevel* is a bit mask that controls which output is generated with values from 1 to 65535. For information about debug levels, see Chapter 4, “Debugging levels ,” on page 131.

-N *certificatename*

Specifies the label associated with the client certificate in the key database file. If an LDAP server is configured to use server authentication only, a client certificate is not required. If the LDAP server is configured to use client and server authentication, a client certificate might be required. This parameter is not required if a default certificate / private key pair is assigned as the default. Similarly, *certificatename* is not required if there is a single certificate / private key pair in the designated key database file. This parameter is ignored if **-Z** or **-K** parameter is not specified.

-o *debugfile*

Specifies the output file name for the server debug messages.

By default, the file with the file name that you specify is created in the /userdata/directory/CustomOut/logs folder.

-p *port*

Specifies a port for the LDAP server to listen. The default LDAP port is 389. If **-p** is not specified and **-Z** is specified, the default LDAP secure port, 636, is used.

-P *keyfilepw*

Specifies the key database password. Password is required to access the encrypted information in the key database file, which might include one or more private keys. If a password stash file is associated with the key database file, then the password is obtained from the password stash file. In this case, the **-P** parameter is not required. This parameter is ignored if **-Z** or **-K** is not specified.

-Q ***operation***

Specifies the crypto device operation with PKCS11. The *operation* variable must be assigned one of the following values:

```
0: No accelerator mode
1: Symmetric
2: Digest
3: Digest and Symmetric
4: Random
5: Random and Symmetric
6: Random and Digest
7: Random, Digest and Symmetric
```

-S ***token_label***

Specifies the token label of the crypto device.

-t [start | stop]

Specifies to start or stop server tracing.

`start`: Starts server trace data collection.

`stop`: Stops server trace data collection.

-v

Indicates to run in verbose mode.

-w ***passwd*** **| ?**

Specifies the password for authentication. Use the `?` prompt to generate a non-echoed password prompt. In UNIX, use backslash **\?** to avoid matching single character file names. If you use the password prompt, it prevents your password from being visible through the **ps** command.

-x

Specifies to use FIPS mode processing. This mode is applicable with SSL or TLS only.

-X ***lib_path***

Specifies the library path of the crypto device.

-Y

Specifies to use a secure LDAP connection by using the startTLS protocol. The `-Y` option is only supported when GSKit is installed.

-Z

Specifies to use a secure LDAP connection by using Secure Sockets Layer (SSL) and Transport Layer Security (TLS). The `-Z` option is only supported for SSL component entry, and only when GSKit is installed.

-1 sec:usec

Specifies the timeout for the `connect()` function in seconds and microseconds. The values provided for seconds and microseconds must be positive integers.

-?

Specifies to show the syntax format.

Security functions

To use the SSL or TLS-related functions associated with this utility, seeChapter 3, “SSL and TLS notes,” on page 129.

See also

ldtrc

Examples

To activate the**ldtrc** facility and to start the server trace with a 2M trace buffer, run the following command.

```
idsldaptrace -h hostname -D adminDN -w adminPW -l on -t start -- -I 2000000
```

To stop the server trace, run the following command.

```
idsldaptrace -h hostname -D adminDN -w adminPW -t stop
```

To switch off the **ldtrc** facility, run the following command.

```
idsldaptrace -h hostname -D adminDN -w adminPW -l off
```

To create an output file with server debug messages in `/userdata/directory/CustomOut/logs/trace.log`, run the following command:

```
idsldaptrace -D cn=root -w root -h 9.113.62.234 -m 65535 -t start -o trace.log
```

To create an output file with server debug messages in `/userdata/directory/CustomOut/logs/mylogs/trace.log`, run the following command:

```
idsldaptrace -D cn=root -w root -h 9.113.62.234 -m 65535 -t start -o mylogs/trace.log
```

To create an output file with server debug messages in `/userdata/directory/CustomOut/logs/tmp/trace.log`, run the following command:

```
idsldaptrace -D cn=root -w root -h 9.113.62.234 -m 65535 -t start -o /tmp/trace.log
```

idsunarchive

Use the **idsunarchive** command to extract the contents of an archive file.

Description

The **idsunarchive** command extracts the contents of the specified archive file. The following archive file types are supported:

- `.tar`
- `.zip`
- `.gz`
- `.tgz`
- `.bz2`

The extracted contents are available in the `/userdata/directory/CustomIn` directory, which you can access on the virtual appliance console. See Managing custom files.

Synopsis

```
idsunarchive [-t archive_type -f archive_name]
```

Options

Use the following parameters with the **idsunarchive** command:

-t *archive_type*
Specifies the type of archive file.

-f *archive_name*
Specifies the file name of the archive file.

Examples

Example 1:

To extract the contents of `myarchive.tar` file to `/userdata/directory/CustomIn` directory, run the following command:

```
idsunarchive -t tar -f myarchive.tar
```

Example 2:

To extract the contents of `myarchive.zip` file to `/userdata/directory/CustomIn` directory, run the following command:

```
idsunarchive -t zip -f myarchive.zip
```

listfiles

Use the **`listfiles`** command to list all files and directories (including sub-directories) of specified base directory.

Description

`listfiles` command lists all files and directories (including sub-directories) of specified base directory. It also gives size of files (in 1K blocks).

Synopsis

`listfiles``[-b base_dir]`

Options

Use the following parameters with the **`listfiles`** command:

```
-b base_dir
```

Specifies the virtual appliance base directory under which you want to get list of the files or directories.

Valid values are:

- `CustomIn`
- `CustomOut`
- `Certificates`

```
sdsva.example.com : client_tools> listfiles
USAGE: listfiles -b <base_dir>
```

where,

```
-b <base_dir>  Base directory to list files.
    Valid values are: CustomIn, CustomOut and Certificates
```

Example 1: To list contents of CustomIn base directory

```
sdsva.example.com : client_tools> listfiles -b CustomIn
   0 Backups
   20 sample.ldif
```

Example 2: To list contents of Certificates base directory:

```
sdsva.example.com : client_tools> listfiles -b Certificates
   4 FDS_Default_SSLCerts
   4 FDS_SCIMTarget_Default_SSLCerts
   4 SCIMService_Default_SSLCerts
   4 SDS_Default_SSLCerts
   4 client.arm
  12 clientc.kdb
   4 defaultwebadmin.arm
   4 defaultwebadmin.jks
   4 server.arm
  20 serverc.kdb
```

Example 3: To list contents of archived_logs sub-directory of CustomOut base directory:

```
sdsva.example.com : client_tools> listfiles -b CustomOut/archived_logs
10268 2018-07-16-16_34_07+05_30_idslogmgmt.log
20536 2018-07-17-11_24_09+05_30_audit.log
```

Note: The size of archived `idslogmgmt.log` is 10MB and `audit.log` is 20MB.

Firmware upgrade commands

Use the firmware upgrade commands under `firmware_update` folder to work with IBM Security Directory Suite appliance firmware updates from the virtual appliance command-line interface.

delete_firmware
Deletes firmware updates from the system.

install_firmware
Installs the available firmware update to the system.

Note: As a precautionary measure to avoid issues, you must ensure that all processes are in the stopped state before you proceed with firmware upgrade.

Important: If FIPS mode is enabled for the virtual appliance, see FIPS compliance for important information before you install a firmware upgrade on a FIPS-compliant virtual appliance.

list_firmware
Lists firmware updates that are available at the download location.

For more information about using these commands, see Firmware upgrades.

Migration tools

Use the migration utilities under the `migration_tools` folder to run operations from the virtual appliance command-line interface for migrating to IBM Security Directory Suite.

fdsmigr

Migrates the configuration files of Federated Directory Server that is configured with Directory Server as target from IBM Security Directory Integrator to IBM Security Directory Suite, Version 8.0.1.x.

For parameters and usage information, see "fdsmigr" on page 61.

fdsscimmigr

Migrates the configuration files of Federated Directory Server that is configured with SCIM as target from IBM Security Directory Integrator to IBM Security Directory Suite, Version 8.0.1.x.

For parameters and usage information, see "fdsscimmigr" on page 62.

idsimigr

Migrates Directory Server instance schema and configuration from IBM Security Directory Server to IBM Security Directory Suite, Version 8.0.1.x.

For parameters and usage information, see "idsimigr" on page 63.

idslogmgmtmigr

Migrates Directory Server log management tool configuration files, which are required for integration with QRadar/Cognos, from IBM Security Directory Server to IBM Security Directory Suite, Version 8.0.1.x.

For parameters and usage information, see "idslogmgmtmigr" on page 64.

idssnmpmigr

Migrates Directory Server SNMP agent configuration files from IBM Security Directory Server to IBM Security Directory Suite, Version 8.0.1.x.

For parameters and usage information, see "idssnmpmigr" on page 64

idswmigr

Migrates Directory Server Web Administration Tool configuration files from IBM Security Directory Server to IBM Security Directory Suite, Version 8.0.1.x.

For parameters and usage information, see "idswmigr" on page 65

migbkup

Creates a backup of the schema and configuration files the IBM Security Directory Suite, Version 8.0.1.x components.

For parameters and usage information, see "migbkup" on page 66

fdsmigr

Use the **fdsmigr** command to migrate Federated Directory Server that is configured with Directory Server as target.

Description

The **fdsmigr** command is a migration utility. It migrates the configuration files of Federated Directory Server that is configured with Directory Server as target from the specified archive file to IBM Security Directory Suite, Version 8.0.1.x.

Migration from IBM Security Directory Integrator, Version 7.2 or later is supported.

Important: As a precautionary measure, take a backup of the current state of the virtual appliance before using this command.

Synopsis

Use the following syntax for the **fdsmigr** command:

```
fdsmigr [-u archive_file]
```

Options

Use the following parameter with the **fdsmigr** command:

-u *archive_file*

Specifies the file name of the archive file that contains the files required for migration.

This is a required parameter.

The following archive file types are supported:

- `.tar`
- `.zip`
- `.gz`
- `.tgz`
- `.bz2`

Example

To migrate the configuration files of Federated Directory Server that is configured with Directory Server as target from the specified archive file, run the following command:

```
fdsmigr -u archive_file
```

fdsscimmigr

Use the **fdscimmigr** command to migrate Federated Directory Server that is configured with SCIM as target.

Description

The **fdscimmigr** command is a migration utility. It migrates the configuration files of Federated Directory Server that is configured with SCIM as target from the specified archive file to IBM Security Directory Suite, Version 8.0.1.x.

Migration from IBM Security Directory Integrator, Version 7.2 or later is supported.

Important: As a precautionary measure, take a backup of the current state of the virtual appliance before using this command.

Synopsis

Use the following syntax for the **fdsscimmigr** command:

```
fdscimmigr [-u archive_file]
```

Options

Use the following parameter with the **fdscimmigr** command:

-u *archive_file*

Specifies the file name of the archive file that contains the files required for migration.

This is a required parameter.

The following archive file types are supported:

- `.tar`
- `.zip`
- `.gz`
- `.tgz`
- `.bz2`

Example

To migrate the configuration files of Federated Directory Server that is configured with SCIM as target from the specified archive file, run the following command:

```
fdscimmigr -u archive_file
```

idsimigr

Use the **idsimigr** command to migrate the Directory Server.

Description

The **idsimigr** command is a migration utility. This command migrates the schema and configuration files of the Directory Server from the specified archive file to IBM Security Directory Suite, Version 8.0.1.x.

Migration from IBM Security Directory Server, Version 6.x.x is supported.

Before starting the migration process, this utility stops the service, if it is running.

After you migrate the schema and configuration files, the command creates a Directory Server instance with the migrated information.

Important: As a precautionary measure, take a backup of the current state of the virtual appliance before using this command.

Synopsis

Use the following syntax for the **idsimigr** command:

```
idsimigr [-d debug_level] [-u archive_file]
```

Options

The **idsimigr** command takes the following parameters.

-d ***debug_level***

Sets the debug level. If you specify this parameter, the command sends the debug output to stdout. The *debuglevel* value is a bit mask that controls which output is generated with values from 1 to 65535. For more information about debug levels, see Chapter 4, “Debugging levels ,” on page 131.

This is an optional parameter.

-u *archive_file*

Specifies the file name of the archive file that contains the files required for migration.

This is a required parameter.

The following archive file types are supported:

- .tar
- .zip
- .gz
- .tgz
- .bz2

Example

To migrate the schema and configuration files of the Directory Server from the specified archive file, run the following command:

```
idsimigr -u archive_file
```

idslogmgmtmigr

Use the **idslogmgmtmigr** command to migrate the Directory Server log management tool.

Description

The **idslogmgmtmigr** command is a migration utility. It migrates the configuration files of the Directory Server log management tool from the specified archive file to IBM Security Directory Suite, Version 8.0.1.x.

Migration from IBM Security Directory Server, Version 6.x.x is supported.

Important: As a precautionary measure, take a backup of the current state of the virtual appliance before using this command.

Synopsis

Use the following syntax for the **idslogmgmtmigr** command:

```
idslogmgmtmigr [-u archive_file]
```

Options

Use the following parameter with the **idslogmgmtmigr** command:

-u *archive_file*

Specifies the file name of the archive file that contains the files required for migration.

This is a required parameter.

The following archive file types are supported:

- .tar
- .zip
- .gz
- .tgz
- .bz2

Example

To migrate the configuration files of the Directory Server log management tool from the specified archive file, run the following command:

```
idslogmgmtmigr -u archive_file
```

idssnmpmigr

Use the **idssnmpmigr** command to migrate the Directory Server SNMP agent.

Description

The **idssnmpmigr** command is a migration utility. It migrates the configuration files of the Directory Server SNMP agent from the specified archive file to IBM Security Directory Suite, Version 8.0.1.x.

Migration from IBM Security Directory Server, Version 6.x.x is supported.

Important: As a precautionary measure, take a backup of the current state of the virtual appliance before using this command.

Synopsis

Use the following syntax for the **idssnmpmigr** command:

```
idssnmpmigr [-u archive_file]
```

Options

Use the following parameter with the **idssnmpmigr** command:

-u *archive_file*
Specifies the file name of the archive file that contains the files required for migration.

This is a required parameter.

The following archive file types are supported:

- .tar
- .zip
- .gz
- .tgz
- .bz2

Example

To migrate the configuration files of the Directory Server SNMP agent from the specified archive file, run the following command:

```
idssnmpmigr -u archive_file
```

idswmigr

Use the **idswmigr** command to migrate the Directory Server Web Administration Tool .

Description

The **idswmigr** command is a migration utility. It migrates the configuration files of the Directory Server Web Administration Tool from the specified archive file to IBM Security Directory Suite, Version 8.0.1.x.

Migration from IBM Security Directory Server, Version 6.x.x is supported.

Important: As a precautionary measure, take a backup of the current state of the virtual appliance before using this command.

Synopsis

Use the following syntax for the **idswmigr** command:

```
idswmigr [-u archive_file]
```

Options

Use the following parameter with the **idswmigr** command:

-u *archive_file*
Specifies the file name of the archive file that contains the files required for migration.

This is a required parameter.

The following archive file types are supported:

- .tar
- .zip

- .gz
- .tgz
- .bz2

Example

To migrate the configuration files of the Directory Server Web Administration Tool from the specified archive file, run the following command:

```
idswmigr -u archive_file
```

migbkup

Use the **migbkup** command to backup the schema and configuration files of IBM Security Directory Suite components.

Description

You can run the **migbkup** command to back up the schema and configuration files of the following IBM Security Directory Suite components:

- Directory Server instance
- Web Administration Tool
- Directory Server log management tool
- Directory Server SNMP agent
- Federated Directory Server with Directory Server as target
- Federated Directory Server with SCIM as target

The **migbkup** command creates an archive file named `migbkup.tar` in the `/userdata/directory/CustomOut` directory.

The `migbkup.tar` contains the following archive files:

- `idsimigr_backup.tar`
- `idswmigr_backup.tar`
- `idslogmgmt_backup.tar`
- `idssnmp_backup.tar`
- `fdsmigr_backup.tar`
- `fdsscim_backup.tar`

Synopsis

```
migbkup
```

Options

The **migbkup** command does not have any parameters.

Example

To create backup the schema and configuration files of the IBM Security Directory Suite components into an archive file, run the following command:

```
migbkup
```

idscfgremotedb

Use the **idscfgremotedb** script to create a new DB2 instance on the Remote DB2 server.

Description

The Directory appliance comes with a pre-configured IBM Security Directory Suite instance that is configured to a local DB2 instance. The user also has the option to configure the existing server instance to use an external or remote database instance on another machine. This is primarily intended for following users:

- Users that have an existing DB2 server
- Users having the need for fine grained control over their DB2 instance for performance tuning and compliance
- Users who need a DB2 level High Availability or Monitoring
- Users who are anticipating high volume of transactions or data having highly scalable disk space requirements.

The **idscfgremotedb** script creates a new DB2 instance on the Remote DB2 server. It requires an existing user on the remote DB2 server with a valid password and proper authority. The path specified by the db2_path must already exists and contains a pre-installed DB2. You can also use an existing DB2 instance that is used by any existing Directory Server. Ensure that the service port is not in use by another service before using the **idscfgremotedb** script.

Synopsis

```
idscfgremotedb [-c]
               [-u username]
               [-w passwd]
               [-t db_name]
               [-p db2_path]
               [-s service port]
               [-Z auth_type]
```

Options

The **idscfgremotedb** script takes the following parameters.

-c

Removes the online backup configuration setup of the database, if the online backup was configured at the database configuration stage either by using the **idscfgdb** command or GUI tools (**idsxinst** or **idsxcfg**).

Note: The **-c** parameter must not be used along with the **-a**, **-t**, and **-l** parameters, if the database is already configured. However, it can be used with the **-w** parameter.

-u*username*

Remote DB2 instance user id having proper authority.

-w *passwd*

Remote DB2 instance user password.

-t *db2_name*

Specifies the DB2 database name.

-p *db2_path*

Specifies the Remote DB2 instance user password.

-s *service port*

Specifies the TABLESPACE container location.

-Z *auth_type*

Specifies the authentication type to create the remote DB2 instance. Accepts values such as **SERVER** or **SERVER_ENCRYPT**. If not specified, the default value is '**SERVER**'.

Examples

Example 1

To configure the Directory Server instance in the virtual appliance to use the remote DB2, run the following command:

```
idscfgremotedb -c -u <user_name> -w <passwd> -t <db_name> -p <db2_path> -s <service_port>
idscfgdb -I <instance_name> -a <instance_user_id> -t <db_alias> -w <instance_user_pwd> -Y
-S <port or service_name> -l <location> -P <remote_server_ip>
-u <remote_username> -p <remote_pwd>
```

Server tools

Use the server utilities under the `server_tools` folder to configure a Directory Server instance from the virtual appliance command-line interface.

ddsetup

Splits an LDIF file into several files for loading into a distributed directory.

For parameters and usage information, see "ddsetup" on page 70.

fdsautostart

Enables the auto start service for Federated Directory Server.

For parameters and usage information, see "fdsautostart" on page 73.

fdsscimautostart

Enables the auto start service for Federated Directory Server SCIM Target.

For parameters and usage information, see "fdsscimautostart" on page 74.

ibmdiradm

Starts or stops the Administration Server that is associated with an instance.

For parameters and usage information, see "ibmdiradm" on page 77.

ibmslapd

Start or stops the Directory Server process.

For parameters and usage information, see "ibmslapd" on page 78.

idsautostart

Starts the Directory Server instance automatically at an operating system startup.

For parameters and usage information, see "idsautostart" on page 79.

idsbulkload

Loads data directly into the DB2 database from an LDIF file. The bulkload utility is a faster and better method to load large amount of data.

For parameters and usage information, see "idsbulkload" on page 79.

idscfgchglg

Configures a change log for a Directory Server instance.

For parameters and usage information, see "idscfgchglg" on page 84.

idscfgdb

Configures DB2 database for a Directory Server instance.

For parameters and usage information, see "idscfgdb" on page 86.

idscfgsch

Configures a schema file for a Directory Server instance.

For parameters and usage information, see "idscfgsch" on page 89.

idscfgsuf

Configures a suffix for a Directory Server instance.

For parameters and usage information, see "idscfgsuf" on page 90.

idsdb2ldif

Outputs Directory Server entries to a text file in LDAP Directory Interchange Format (LDIF).

For parameters and usage information, see "idsdb2ldif" on page 91.

idsdbback
Takes a backup of the directory data and configuration files.

For parameters and usage information, see "idsdbback" on page 94.

idsdbmaint
Runs database maintenance activities for a Directory Server instance, such as DB2 index reorganization, DB2 row compression on tables, and DB2 table space conversion.

For parameters and usage information, see "idsdbmaint" on page 96.

idsdbrestore
Restores a database and configuration files for a Directory Server instance.

Note: This command can restore a database only to the original database instance with the same database name. Also, the version of the Directory Server instances must be the same.

For parameters and usage information, see "idsdbrestore" on page 97.

idsdnpw
Sets the administration DN and password for an instance.

For parameters and usage information, see "idsdnpw" on page 98.

idsenvvars
Manages environment variables for Directory Server.

For parameters and usage information, see "idsenvvars" on page 100.

idsgendirksf
Regenerates a key stash file for a Directory Server instance.

For parameters and usage information, see "idsgendirksf" on page 101.

idsilist
Lists the Directory Server instances on the system.

For parameters and usage information, see "idsilist" on page 102.

idsldif2db
Loads entries from an LDIF file to a database that is associated with a Directory Server instance.

For parameters and usage information, see "idsldif2db" on page 103.

idslogmgmt
Starts or stops the Directory Server log management tool.

For parameters and usage information, see "idslogmgmt" on page 105.

idsperftune
Tunes the Directory Server performance. Administrators can use this command to achieve a higher directory performance by tuning caches, DB2 buffer pools, and DB2 parameters.

For parameters and usage information, see "idsperftune" on page 109.

Note: When you run this command with the **-b** parameter, specify `/userdata/directory/CustomOut` folder as the output directory. You can then download the output file by using the **Configure** > **Advanced Configuration** > **Custom File Management** option in the virtual appliance console.

idsrunstats
Optimizes the database of a Directory Server instance.

For parameters and usage information, see "idsrunstats" on page 112.

idssethost
Sets the IP address on which the Directory Server must listen.

For parameters and usage information, see "idssethost" on page 113.

Note: Before you use **idssethost** to configure Directory Server to bind to an IP address, you must enable an application interface for Directory Server by using the **Application Interfaces** page of the virtual appliance console. See Managing application interfaces.

idssetport

Sets the port and secure port on which the Directory Server must listen.

For parameters and usage information, see “idssetport” on page 114.

idssnmp

Starts or stops the Directory Server SNMP agent tool.

For parameters and usage information, see “idssnmp” on page 115.

idsucfgchglg

Unconfigures a change log for a Directory Server instance.

For parameters and usage information, see “idsucfgchglg” on page 116

idsucfgdb

Unconfigures the DB2 database that is associated with a Directory Server instance.

For parameters and usage information, see “idsucfgdb” on page 117.

idsucfgsch

Unconfigures a schema file for a Directory Server instance.

For parameters and usage information, see “idsucfgsch” on page 118.

idsucfgsuf

Removes a suffix from the Directory Server instance.

For parameters and usage information, see “idsucfgsuf” on page 120.

idsunlockwat

Unlocks the Directory Server Web Administration Tool console.

For parameters and usage information, see “idsunlockwat” on page 121.

ldtrc

Activates or deactivates tracing of a Directory Server. You can use the trace options that are provided with the command to troubleshoot the instance-specific issues.

For parameters and usage information, see “ldtrc” on page 121.

scimautostart

Enables the auto start service for Directory Integrator SCIM Service

For parameters and usage information, see “scimautostart” on page 123.

setservertype *server_type*

Sets the Directory Server instance type.

For parameters and usage information, see “setservertype” on page 124.

unlockadmin

Unlocks the IBM Security Directory Suite virtual appliance `admin` user.

For parameters and usage information, see “unlockadmin” on page 124.

ddsetup

Use the **ddsetup** to split an LDIF file for loading it in distributed directories.

Description

The **ddsetup** command splits a LDAP data interchange format (LDIF) file by using the partition algorithm that is specified in the configuration file of Proxy Server. The split LDIF files can be loaded into a distributed directory. You can specify the partition algorithm in the `ibm-slapdDNPartitionPlugin` attribute of a Proxy Server configuration file.

Restriction: Composite DN is not supported by the **ddsetup** command.

Synopsis

```
ddsetup [[-I proxy_inst_name] [-B base_DN] [-i input_file]]
        | [-f config_file] [-d debug_level] [-l output_location]
        [-s] [-v] -?
```

Options

The options for the **ddsetup** command are listed.

-B *base_DN*
Specifies the base DN or split DN to partition entries by the **ddsetup** command.

-d *debug_level*
Specifies the LDAP debug level to use with the **ddsetup** command.

-f *config_file*
Specifies the configuration file to use with the **ddsetup** command.

-I *proxy_inst_name*
Specifies the name of the Proxy Server instance.

-i *input_file*
Specifies the file from which to read.

-l *output_location*
Specifies the directory to place the output files from the **ddsetup** command.

-s
Specifies to set the statistics mode for the **ddsetup** command.

-v
Specifies to show the version information of the **ddsetup** command.

-?
Specifies to show the syntax help of the command.

Examples

Distributing data between back-end servers

Consider a database with 5 million entries for the o=sample subtree. You want to distribute this data over five back-end servers. Export the entries to an LDIF file for distributing the entries among the back-end servers. For information about exporting data to an LDIF file, see "idsdb2ldif" on page 91.

You must cryptographically synchronize the back-end servers. To synchronize, the encryption seed and salt values for the back-end servers must be same. To create an LDIF file for each back-end server by using the partition algorithm of the Proxy Server, run the following steps:

1. To create an LDIF file, run the **idsdb2ldif** command. For example:

   ```
   idsdb2ldif -o mydata.ldif -s o=sample -I instance_name
   ```

2. Run the **ddsetup** command to split the data.

   ```
   ddsetup –I proxy_instance -B o=sample -i mydata.ldif
   ```

   The **ddsetup** command divides the mydata.ldif file into multiple LDIF output files. The files are created based on the number of partitions that are defined in the configuration file of the Proxy Server. The first output file corresponds to the partition index 1. The second output file corresponds to the partition index 2.

3. Run the **idsldif2db** or **idsbulkload** command to load the data to an appropriate back-end server. For each partition index value, you can create an LDIF file. You must load the correct LDIF file on the back-end server with the corresponding partition index value. Otherwise, the Proxy Server might not be able to retrieve the entries.

   ```
   ServerA (partition index 1) - out1.ldif
   ServerB (partition index 2) - out2.ldif
   ```

```
ServerC (partition index 3) - out3.ldif
ServerD (partition index 4) - out4.ldif
ServerE (partition index 5) - out5.ldif
```

Distributing data between servers for multiple subtrees

You can split data among multiple subtrees. Consider a parent DN entry, `o=sample`, split among three subtrees: `ou=austin,o=sample`, `ou=raleigh,o=sample`, and `ou=poughkeepsie,o=sample`. The data on each of these subtrees is further subdivided between the back-end servers. For example:

- `ou=austin,o=sample` - five back-end servers
- `ou=raleigh,o=sample` - three back-end servers
- `ou=poughkeepsie,o=sample` - four back-end servers

1. To create an LDIF file from an existing database, run the **idsdb2ldif** command. For example:

   ```
   idsdb2ldif -o mydata.ldif -s o=sample -I instance_name
   ```

2. Run the **ddsetup** command to split the data.

   ```
   ddsetup –I proxy_instance -B "o=sample" -i mydata.ldif
   ```

   where, *proxy_instance* is a Proxy Server instance.

   The **ddsetup** command divides the `mydata.ldif` file into multiple LDIF output files. The first output file for the subtree corresponds to the partition index 1. The second output file corresponds to the partition index 2, and so on. The partition index number starts from 1 for each subtree that is being distributed.

3. Use **idsldif2db** or **idsbulkload** command to load the data to an appropriate back-end server. An example file is created for each partition index value. You must load the correct LDIF file on the back-end server with the corresponding partition index value. Otherwise, the Proxy Server might not be able to retrieve the entries.

   ```
   ServerA (partition index 1) - out1_ServerA.ldif
   ServerB (partition index 2) - out2_ServerB.ldif
   ServerC (partition index 3) - out3_ServerC.ldif
   ServerD (partition index 4) - out4_ServerD.ldif
   ServerE (partition index 5) - out5_ServerE.ldif
   ServerF (partition index 1) - out1_ServerF.ldif
   ServerG (partition index 2) - out2_ServerG.ldif
   ServerH (partition index 3) - out3_ServerH.ldif
   ServerI (partition index 1) - out1_ServerI.ldif
   ServerJ (partition index 2) - out2_ServerJ.ldif
   ServerK (partition index 3) - out3_ServerK.ldif
   ServerL (partition index 4) - out4_ServerL.ldif
   ```

Splitting the `ddsample.ldif` file

An example that describes how to use the **ddsetup** command to split the `ddsample.ldif` file.

1. Configure `o=sample` as a partition base on the Proxy Server. Run the **idscfgsuf** command, for example:

   ```
   idscfgsuf -I proxy_instance -s o=sample
   ```

   where,

   proxy_instance is the Proxy Server instance name
   `o=sample` is the configured partition base with the Proxy Server

2. Set the administrator DN and password for the Proxy Server instance. Run the **idsdnpw** command, for example:

   ```
   idsdnpw -I proxy_instance -u cn=root -p rootPWD
   ```

   where,

   proxy_instance is the Proxy Server instance name

    `cn=root` is the administrator DN
    `rootPWD` is the administrator password

3. Start the Proxy Server instance in configuration-only mode. Run the **ibmslapd** command, for example:

   ```
   ibmslapd -I proxy_instance -a
   ```

   where, *proxy_instance* is the Proxy Server instance name

4. Add the configuration for splitting `o=sample` into three partitions. Run the **ldapadd** command, for example:

   ```
   ldapadd -D cn=root -w rootPWD -p port -f ddibmslapd.conf
   ```

   where,

   `cn=root` is the administrator DN
   `rootPWD` is the administrator password
   port is the port number on which the proxy server is listening
   `ddibmslapd.conf` is the sample configuration file

5. To split the LDIF file, run **ddsetup** with the sample data.

   ```
   ddsetup -I proxy_instance -B o=sample -i ddsample.ldif
   ```

   where,

   proxy_instance is the Proxy Server instance
   `o=sample` is the partition base
   `ddsample.ldif` is the sample LDIF file

   The `ddsample.ldif` and `ddibmslapd.conf` files are available in the `examples` directory. The **ddsetup** command divides the `ddsample.ldif` into multiple LDIF output files. The first output file for the subtree corresponds to the partition index 1. The second output file corresponds to the partition index 2, and so on. The partition index number starts from 1 for each subtree that must be distributed. The **ddsetup** command generates the following files.

   ```
   sample_1.ldif
   sample_2.ldif
   sample_3.ldif
   default.ldif
   ```

   The `default.ldif` file contains all the entries that did not conform to partitioning rules configured for the Proxy Server.

6. Use **idsldif2db**, **idsbulkload**, or **ldapadd** command to load the data to the appropriate back-end server. You must load the correct LDIF file on the back-end server with the corresponding partition index value. Otherwise, the Proxy Server might not be able to retrieve the entries.

   ```
   Server1 (partition index 1) - sample_1.ldif
   Server2 (partition index 2) - sample_2.ldif
   Server3 (partition index 3) - sample_3.ldif
   ```

fdsautostart

Use the **fdsautostart** command to enable the auto start service for Federated Directory Server.

Description

Enable or disable the auto start service for Federated Directory Server with the **fdsautostart** command.

```

sdsva.example.com: server_tools> fdsautostart
```

```
USAGE: fdsautostart -e | -d
```

Options

The command takes the following parameters:

-e
Enables automatic startup of Federated Directory Server during boot up.

-d
Disables automatic startup of Federated Directory Server during boot up.

Examples

Example 1
To enable startup of Federated Directory Server during boot up, run the following command:

```
fdsautostart -e
```

Example 2
To disable startup of Federated Directory Server during boot up, run the following command:

```
fdsautostart -d
```

fdsscimautostart

Use the **fdsscimautostart** command to enable the auto start service for Federated Directory Server SCIM Target.

Description

Enable or disable the auto start service for Federated Directory Server SCIM Target with the **fdsscimautostart** command.

```
sdsva.example.com: server_tools> fdsscimautostart -e | -d

USAGE: fdsscimautostart -e | -d
```

Synopsis

```
fdsscimautostart -e | -d
```

Options

The command takes the following parameters:

-e
Enable automatic startup of Federated Directory Server SCIM Target during boot up.

-d
Disable automatic startup of Federated Directory Server SCIM Target during boot up.

Examples

Example 1:
To enable auto startup for Federated Directory Server SCIM Target, run the following command:

```
fdsscimautostart -e
```

Example 2:

To disable auto startup for Federated Directory Server SCIM Target, run the following command:

```
fdsscimautostart -d
```

idscfgauditdb

Use the **idscfgauditdb** command to create and configure the audit database that is required for audit reporting.

Description

The audit database is a DB2 database, where all the audit events from the audit log file of Directory Server instance are dumped. Use the **idscfgauditdb** command to create and configure the audit database. See Creating and configuring the audit database in the IBM Security Directory Suite documentation.

This utility uses the database schema file, `sdsAuditDB.sql`. Before you use this utility, you must make sure that the schema file is copied to the directory where this utility is located.

UNIX systems

Synopsis

```
idscfgauditdb    [ [-c | -r | -e] [-u user_name] [-w passwd] [-p db2_path]
                 [-s service_port] [-t db_name] [-d] [-v] ] | -h | -?
```

Options

The **idscfgauditdb** command takes the following parameters.

-c

Creates the DB2 instance and database.

You cannot use this parameter with the **-r** or **-e** options.

This parameter requires the **-u**, **-w**, **-p**, **-s**, and **-t** options.

-u *user_name*

Specifies the user name of the DB2 instance owner.

-w *passwd*

Specifies the password for the DB2 instance owner.

-t *db_name*

Specifies the name of DB2 database you want to create.

-p *db2_path*

Specifies the installation location of DB2.

-s *service_port*

Specifies the port at which the DB2 instance service must listen.

-r

Removes the DB2 database and instance.

You cannot use this parameter with the **-c** or **-e** options.

This parameter requires the **-u**, **-t**, and **-p** options.

-e

Indicates that the data from all DB2 tables in the database must be erased without dropping the tables.

You cannot use this parameter with the **-c** or **-r** options.

This parameter requires the **-u**, **-t**, **-w**, and **-p**options.

-d

Runs in debug mode and shows the DB2 commands as they get executed.

-v

Shows verbose output.

This option also turns on the debug mode.

-h | -?

Shows the usage.

Windows systems

Synopsis

```
idscfgauditdb.cmd  [ [-c | -r | -e] [-u user_name] [-w passwd] [-l db_loc]
                   [-p db2_path] [-s service_port] [-t db_name] [-d] [-v] ] | -h | -?
```

Options

The **idscfgauditdb.cmd** command takes the following parameters.

-c

Creates the DB2 instance and database.

You cannot use this parameter with the **-r** or **-e** options.

This parameter requires the **-u**, **-w**, **-p**, **-l**, **-t**, and **-s** options.

-u *user_name*

Specifies the user name of the DB2 instance owner.

The user must exist on the system with a valid password.

-w *passwd*

Specifies the password for the DB2 instance owner.

-t *db_name*

Specifies the name of DB2 database you want to create.

-l *db_loc*

Specifies the location to create the database.

-p *db2_path*

Specifies the installation location of DB2.

-s *service_port*

Specifies the port at which the DB2 instance service must listen.

-r

Removes the DB2 database and instance.

You cannot use this parameter with the **-c** or **-e** options.

This parameter requires the **-u**, **-t**, **-l**, and **-p** options.

-e

Indicates that the data from all DB2 tables in the database must be erased without dropping the tables.

You cannot use this parameter with the **-c** or **-r** options.

This parameter requires the **-u**, **-t**, **-w**, **-l**, and **-p** options.

-d

Runs in debug mode and shows the DB2 commands as they get executed.

-v

Shows verbose output.

This option also turns on the debug mode.

-h | -?
Shows the usage.

ibmdiradm

Use the **ibmdiradm** command to start or stop the Administration Server.

Description

The **ibmdiradm** command starts or stops the Administration Server that is associated with an instance. The **ibmdiradm** command changes the working directory to `*instance_home*/idsslapd-*instance_name*/workdir`. Therefore, relative paths are considered as relative to `*instance_home*/idsslapd-*instance_name*/workdir`.

Synopsis

```
ibmdiradm | ibmdiradm [-I instance_name [-f config_file] [-h debug_level] [-t]
          [[ [-p port] [-s secure_port] [-c]] | -k ] ] | -v | -?
          | -h ?
```

Options

The **ibmdiradm** takes the following parameters.

-f *config_file*
Specifies the full path of the configuration file to use. If this parameter is not specified, the default configuration file for the Directory Server instance is used.

-h *debug_level*
Sets the LDAP debug level to *debug_level*. If you specify this parameter, it sends the debug output to stdout. The *debug_level* value is a bit mask that controls which output is generated with values from 1 to 65535. For more information about debug levels, see Chapter 4, "Debugging levels ," on page 131.

-h ?
Specifies to show the help for debug levels.

-I *instance_name*
Specifies the name of the Administration Server instance to start or stop.

-k
Specifies to stop the Administration Server

-p *port*
Specifies the port on which Administration Server listens.

-s *secure_port*
Specifies the secure port on which Administration Server listens

-v
Specifies to print the version information.

-?
Specifies to show the syntax help.

-c
Specifies to run the server in console mode.

-t
Specifies to tail the server log until final startup messages are printed on the console.

Examples

Example 1:
To start the Administration Server that is associated with an instance, run the following command:

```
ibmdiradm -I instance_name
```

Example 2:

To stop the Administration Server that is associated with an instance, run the following command:

```
ibmdiradm -I instance_name -k
```

ibmslapd

Use the **ibmslapd** command to start or stop the Directory Server process.

Description

The **ibmslapd** command changes the working directory to *instance_home*/ibmslapd-*instance*/workdir. Therefore, relative paths are considered as relative to *instance_home*/ibmslapd-*instance*/workdir.

Synopsis

```
ibmslapd | ibmslapd [-I instancename [-f configfile] [-h debuglevel] [-t]
[[ [-p port] [-s secureport] [-R ServerID] [-c] [-a | -n] ]
| -k | -i | -u] ] | -v | -? | -h ?
```

Options

The **ibmslapd** or **ibmslapd** command takes the following parameters.

-a

Specifies to start the server in configuration only mode.

-f *configfile*

Specifies the full path to the configuration file to update. If this parameter is not specified, the default configuration file for the Directory Server instance is used.

-h *debuglevel*

Sets the LDAP debug level to *debuglevel*. If you specify this parameter, the command sends the debug output to stdout. The *debuglevel* value is a bit mask that controls which output is generated with values from 1 to 65535. For information about debug levels, see Chapter 4, “Debugging levels ,” on page 131.

-h ?

Specifies to show the debug help.

-I *instancename*

Specifies the Directory Server instance name.

-k

Specifies to stop the Directory Server process.

-n

Specifies not to start the server in configuration only mode, if an error is encountered.

-p *port*

Specifies the port on which the Directory Server instance listens on.

-R *serverID*

Specifies to use the server ID while you run a Directory Server instance.

-s *secureport*

Specifies the secure port on which the Directory Server instance listens on.

-v

Specifies to show the version information of the command.

-?

Specifies to show the syntax format.

-c

Specifies to run the server in console mode.

-t
Specifies to tail the server log until final startup messages are printed on the console.

Examples

Example 1:
To start the Directory Server process for the instance, `myinst`, run the following command:

```
ibmslapd -I myinst
```

Example 2:
To stop the Directory Server process for the instance, `myinst`, run the following command:

```
ibmslapd -I myinst -k
```

idsautostart

Use the **idsautostart** command to start the Directory Server instance automatically at an operating system startup.

Description

If you want to the Directory Server instance to start automatically at an operating system startup, run the **idsautostart** command.

Synopsis

```
idsautstart -e | -d
```

Options

Use the following parameters with the **idsautostart** command.

-e
Enables automatic startup of directory server during operating system startup.

-d
Disables automatic startup of directory server during operating system startup.

idsbulkload

Use the **idsbulkload** command to load directory data from an LDIF file to a Directory Server instance.

Description

The **idsbulkload** command loads the directory data from an LDIF file to a Directory Server instance. This command is faster than **idsldif2db** to load data in LDIF format, and is available for bulk-loading large amounts of data.

⚠ **Attention:** To import LDIF data from another instance, you must cryptographically synchronize with the instance that is importing the LDIF file. Otherwise, any AES encrypted values in the LDIF file do not get imported. For information about synchronizing directory server instances, see Synchronizing two-way cryptography between server instances.

You must consider the following points before you use the **idsbulkload** command.

Note:

- Stop the Directory Server instance before you run the server import utilities.
- Ensure that no applications are attached to the database associated with the Directory Server instance. If there are applications that are attached, the server import utilities might not run.

- Environment variables that are associated with **idsbulkload** are not supported. The *ACLCHECK*, *ACTION*, *LDAPIMPORT*, *SCHEMACHECK*, and *STRING_DELIMITER* environment variables are replaced with the **-A**, **-a**, **-L**, **-S**, **-s** command-line parameters. The command-line switches are case-sensitive.
- If archival logging is set in DB2, the **idsbulkload** command might fail. Make sure that the archival log is disabled before you run the **idsbulkload** command. To disable archival logging, run the following command.

  ```
  update database configuration for ldapdb2 using LOGRETAIN OFF USEREXIT OFF
  ```

- When you load the data that contains unique attributes, the DB2 unique constraints for the modified attributes are dropped. After you load the data, the DB2 unique constraints are established for the following attributes:
  - Attributes with unique constraints dropped.
  - Unique attributes that are listed in the unique attribute entry in the file.

  If duplicate values are loaded for attributes that are specified as unique attributes, the DB2 unique constraint is not created for that attribute. This log is recorded in the `idsbulkload.log` file.
- If you are loading data to an instance already containing data, make sure that you take a backup before you run **idsbulkload** to add entries.
- By default, the action of **bulkoad** is unrecoverable. If data loading fails for any reason, all data in the database is lost. Therefore, it is better to take a backup before and after a large bulkload activity.

Synopsis

```
idsbulkload | idsbulkload -i ldiffile [-I instancename
            [-a <parse_and_load|parseonly|loadonly>] [-A <yes|no>]
            [-b] [-c | -C <yes|no>] [-d <debuglevel>] [-e drop_index]
            [-E <number>] [-f configfile] [-g] [-G] [-k <number>]
            [-L <path>] [-n | -N] [-o <filename>]
            [-s <character>] [-R <yes|no>] [-S <yes|no|only>]
            [-t <filename>] [-v]
            [-W outputfile] [-x|-X <yes|no>]] -Y | [-?]
```

Options

The **idsbulkload** command takes the following parameters.

-a <parse_and_load|parseonly|loadonly>
Specifies the load action mode.

-A <yes|no>
Specifies whether to process the ACL information that is contained in the LDIF file. The default is **yes**. The **no** parameter loads the default ACL.

Note: This parameter is deprecated.

-b
Specifies to suppress the progress indicator.

-c | -C <yes|no>
Skips index recreation.

If you are running successive bulkload operations and you want to skip index recreation between loads, you can postpone index creation until the last bulkload. Issue the last **idsbulkload** command with **-c yes**.

-d ***debuglevel***
Specifies the *debuglevel* to assign and to set the debug mode. Use this parameter to determine the data records that might have a problem and is causing parsing errors. For more information about debug levels, see Chapter 4, “Debugging levels ,” on page 131.

Note: Ensure that the **ldtrc** command is run before you use the **-d** parameter with the command. Otherwise, no messages are shown. To run tracing, issue the `ldtrc` on command.

-e *drop_index*
Specifies whether to drop indexes before load.

-E *number*
Specifies a number limit for parsing the errors reported. When the limit is reached, the **idsbulkload** command exits. The default value is infinity.

-f *configfile*
Specifies the Directory Server instance configuration file.

-g
Specifies not to strip the trailing spaces in attribute values.

-G
Specifies to add members to existing static groups. This parameter must not be specified when the **-k** parameter is specified.

-i *ldiffile*
Specifies the name of the LDIF file with path with data to load into the Directory Server instance. The `sample.ldif` in the `CustomIn` folder contains sample data in the LDIF format.

-I *instancename*
Specifies the name of the Directory Server instance.

-k *number*
Specifies the number of entries to process in one parse-load cycle. The**-a** parameter must be set to `parse_and_load`. This parameter must not be specified when the **-G** parameter is specified.

-L *path*
Specifies the directory for storing temporary data.

The default location is `/home/sdsinst1/idsslapd-sdsinst1/tmp/ldapimport/`.

-n | -N
Specifies that the load is unrecoverable. With this parameter, **idsbulkload** uses less disk space and runs faster. If data loading fails for any reason, all the data in the database is lost.

-o *filename*
Specifies to generate an output file to preserve the `IBM-ENTRYUUID` entry and the timestamp values created during the parsing phase of **idsbulkload**.

-R <yes|no>
Specifies whether to remove the directory that was used for storing temporary data. The directory to remove is the default directory or the one specified by using the **-L** parameter. The default value is **yes**.

Note: Even if the default is **yes** for the parameter, there are two exceptions. If **idsbulkload** ends in an error condition, the temporary files are not deleted on error. It is because the files are required for recovery. If a user chooses the **-a parseonly** parameter, the temporary files are not deleted because the files are needed for the load phase.

-s *character*
Specifies the string delimiting character that is used for importing.

Note: The **idsbulkload** command might fail to load LDIF files that contain certain UTF-8 characters. The reason for the failure is because the DB2 LOAD tool parses the default **idsbulkload** string delimiter, which is a vertical bar `(|)`, in multi-byte character sets. In such scenarios, reassign the string delimiter to any of the supported delimiters except the vertical bar (|).

For example, the following symbols are supported: `"%&'()*,./:;<>?`.

To assign a delimiting character, see the following example:

```
idsbulkload -i ldiffile -I instancename -s <
```

```

idsbulkload -i ldiffile -I instancename -s <any of the supported delimiters except |>
```

To avoid this failure, ensure that the new delimiting character is not present in your `ldif` file.

`-S <yes|no|only>`
Verifies whether the directory entries are valid based on the object class definitions and attribute type definitions in the configuration files.

Schema checking verifies that all object classes and attributes are defined. It also checks whether the attributes that are specified for each entry comply with the list of `required` and `allowed` attributes in the object class definition. Also verifies whether the binary attribute values are in the correct 64-bit encoded form.

`yes`
Specifies to run schema check on the data before the command adds it to the Directory Server instance.

`no`
Specifies not to run schema check on the data before the command adds it to the Directory Server instance. It is the default option. This option improves the operational performance. This option assumes that the data in the file is valid.

`only`
Specifies to run schema check on the data only and not to add data to the Directory Server instance. This option provides the most feedback and reports errors.

It is advisable to use the **`-S only`** parameter to validate the data first, and then to use **`-S no`** to load the data.

`-t` ***filename***
Specifies to use the `IBM-ENTRYUUID` entry and the timestamp values from the file instead of generating them during the parsing. If the values are present in the LDIF file in the form of controls, the controls are ignored.

`-v`
Specifies the verbose mode for the command.

`-W` ***outputfile***
Specifies the full path of a file to redirect output.

`-x | -X <yes|no>`
Specifies whether to translate entry data to database code page. The default value is **no**.

Note: This parameter is necessary only when you use a database other than UTF-8.

`-Y`
Generates remote bulkload scripts for the remote database that is already configured to the Directory Server instance.

Note: The **`-Y`** option for bulkload utility on a remote database must always be used along with the **`-a parseonly`** option.

`-?`
Specifies to show the syntax help.

Usage

To load considerable large amount of data to a Directory Server instance, you must use the **`idsbulkload`** command. To improve operational performance of the **`idsbulkload`** command, you can ignore schema check of the data in the file. During parsing and loading, the **`idsbulkload`** command run some basic checks on the data.

When you run the **`idsbulkload`** command, you must stop the Directory Server instance (the **`idsslapd`** process).

The **`idsbulkload`** command requires disk space for storing temporary data during the parse and load stage. The **`idsbulkload`** command also requires temporary storage for data manipulation before it loads the data into the database.

The **-o** and **-t** parameters are useful when you add large amounts of data into existing replication environments. If servers A and B are peer servers and you want to add entries under the replication context of an instance, do the following steps.

1. Generate the LDIF file.
2. Run **idsbulkload** with the **-o** parameter on server A to load the data and to create a file with all operational attributes during bulkload.
3. Copy the operational attributes output file to server B.
4. Run **idsbulkload** with the **-i** and **-t** parameters to import the LDIF file with the same operational attributes. This command ensures that the operational attribute values are preserved across the replicating servers under the same replication context.

The **-G** parameter is useful when you expand an existing static group with many members. The existing entry must have an object class that accepts `member` or `uniquemember` as its attribute. For example, if you wanted to add 5 million members from the static group, `ou=static group 1, o=company1`, to another group, `ou=static group A, o=companyA`, do the following steps.

1. Create an LDIF file from the source server. Use an editor to remove any attributes other than `member` or `uniquemember` from the file. For example:

   ```
   dn: ou=static group 1, o=company1, c=us
   member: cn=member1, o=company1, c=us
   member: cn=member2, o=company1, c=us
   member: cn=member3, o=company1, c=us
   ...
   member: cn=member5000000, o=company1, c=us
   ```

2. Modify the DN of the group in the file to match the DN of the existing group entry on the target server. For example:

   ```
   dn: ou=static group A, o=companyA, c=us
   member: cn=member1, o=company1, c=us
   member: cn=member2, o=company1, c=us
   member: cn=member3, o=company1, c=us
   ...
   member: cn=member5000000, o=company1, c=us
   ```

3. Make the necessary global changes to the file. In this case, the company name must be changed for each member attribute.

   ```
   dn: ou=static group A, o=companyA, c=us
   member: cn=member1, o=companyA, c=us
   member: cn=member2, o=companyA, c=us
   member: cn=member3, o=companyA, c=us
   ...
   member: cn=member5000000, o=companyA, c=us
   ```

4. To avoid memory issues, divide the file into multiple files of manageable size. In this example, a source file is divided into five files of 1 million attributes. Later, copy the DN as the first line in each file.

   For example, `file1`:

   ```
   dn: ou=static group A, o=companyA, c=us
   member: cn=member1, o=companyA, c=us
   member: cn=member2, o=companyA, c=us
   member: cn=member3, o=companyA, c=us
   ...
   member: cn=member1000000, o=companyA, c=us
   ```

   For example, `file2`:

   ```
   dn: ou=static group A, o=companyA, c=us
   member: cn=member1000001, o=companyA, c=us
   member: cn=member1000002, o=companyA, c=us
   member: cn=member1000003, o=companyA, c=us
   ...
   member: cn=member2000000, o=companyA, c=us
   ```

`file3`:

```
dn: ou=static group A, o=companyA, c=us
member: cn=member2000001, o=companyA, c=us
member: cn=member2000002, o=companyA, c=us
member: cn=member2000003, o=companyA, c=us
...
member: cn=member3000000, o=companyA, c=us
...
```

5. Run the **idsbulkload** command with the **-G** parameter to load the files to the target server.

The **idsbulkload** command verifies whether the DN exists and that its object class and attributes are valid before you load the file.

Note: The**idsbulkload** command does not check for duplicate attributes.

You must inspect the output messages from the **idsbulkload** command carefully. If errors occur during the operation, the instance might not get populated. You might require to drop all the LDAP tables, or drop the database (re-create an empty database), and start over. If no data is added to the instance, then bulkload process must be attempted again. If you drop all the LDAP tables, you might lose any existing data in the instance.

The `/userdata/directory/CustomIn/sample.ldif` file includes sample data. You can use data in this file to experiment with populating a directory by using the **idsbulkload** command, or you can use the **idsldif2db** command. The **idsldif2db** command is considerably slower than the **idsbulkload** command for large amounts of data.

For performance reasons, the **idsbulkload** command does not check for duplicate entries. Ensure that your LDIF file does not contain duplicate entries. If any duplicates exist, remove the duplicate entries.

If **idsbulkload** fails at the DB2 LOAD phase, see the `db2load.log` file to determine the cause. The location of the log file is `/home/sdsinst1/idsslapd-sdsinst1/tmp/ldapimport/`.

Correct the problem and rerun **idsbulkload**. The **idsbulkload** command loads the files from the last successful load consistency point.

If **idsbulkload** fails, the recovery information is stored in the following file. This file is not removed until all of the data is successfully loaded, and ensures the data integrity of the Directory Server instance. If you configure the database again and start over, the `idsbulkload_status` file must be removed manually. Otherwise, **idsbulkload** tries to recover from the last successful load point.

The file is in the `/home/sdsinst1/idsslapd-sdsinst1/logs/bulkload_status` directory.

idscfgchglg

Use the **idscfgchglg** command to configure a change log for a Directory Server instance.

Description

The **idscfgchglg** command configures a change log for a Directory Server instance. The change log is a database that is created in the same database server instance as the instance database. The change log entry is added to the `ibmslapd.conf` file of a Directory Server instance. A change log requires only the directory server instance name for which it is configured. A change log uses the database instance name that is associated with the Directory Server instance and creates a database in the same database instance. Before you run this command, ensure that a database instance with the same name as the Directory Server instance must exist. Also, create a database for a Directory Server instance. On UNIX and Linux® systems, the local loopback service must be registered in the `/etc/services` file.

You can optionally specify the maximum number of entries to keep in the change log and the maximum age of the entries before it is removed. If you do not specify any options, the entries in the change log never expire and is stored in the change log. A maximum of 1,000,000 entries can be stored in the change log.

Synopsis

```
idscfgchglg [-I instancename [-m maxentries] [-y maxdays] [-h maxhours]
            [-f configfile] [-d debuglevel] [-b outputfile] [-q] [-n]] |
            -v | -? [-Y]
```

Options

The **idscfgchglg** takes the following parameters.

-b ***outputfile***
Specifies the file name to redirect console output. If you use this parameter with the **-q** parameter, errors are sent to the *outputfile* file. If the debug mode is set, the debug output is also sent to this file.

Note: The output file is created in the CustomOut folder.

-d ***debuglevel***
Sets the LDAP debug level to *debuglevel*. If you specify this parameter, it sends the debug output to stdout. The *debuglevel* value is a bit mask that controls which output is generated with values from 1 to 65535. For more information about debug levels, see Chapter 4, “Debugging levels ,” on page 131.

-f ***config_file***
Specifies the customized configuration file. If this parameter is not specified, the default configuration file for the Directory Server instance is used.

Note: The customized configuration file must be uploaded to CustomIn folder before you can use this parameter.

-h ***maxhours***
Specifies the maximum duration in hours to keep entries in the change log. This parameter can be used with the **-y** *maxdays* to specify the maximum age of a change log entry.

-I ***instancename***
Specifies the Directory Server instance name for which to configure change log.

-m ***maxentries***
Specify the maximum number of entries to keep in the change log. If 0 is specified, it indicates that there is no limit on the number of entries.

-n
Specifies to run no prompt mode. All output from the command is generated, except for messages that require user interaction. Use this parameter with the **-w** parameter.

-q
Specifies to run in quiet mode. All output from the command is suppressed, except for error messages. If you specify the **-d** parameter also, then the trace output is not suppressed.

-y ***maxdays***
Specifies the maximum duration in days to keep the entries in the change log. If 0 is specified, it indicates that there is no age limit on entries in the change log. You can use this parameter with **-h** *maxhours* to specify the maximum age of a change log entry.

-v
Specifies to show the version information of the command.

-?
Specifies to show the syntax help.

-Y
Specify the Remote Database as the target.

Examples

Example 1:
To configure a change log with no age limit or size limit, run the following command:

```
idscfgchglg –m 0
```

Example 2:

To configure a default change log with a size limit of 1,000,000 and an entry age of 25 hours, run the following command:

```
idscfgchglg –y 1 –h 1
```

Note: After you configure the change log, the **-y**, **-h**, and **-m** parameters can be used to update the maximum age and size of the entries in the change log.

idscfgdb

Use the **idscfgdb** command to configure a database for a Directory Server instance.

Description

The **idscfgdb** command configures the database for a Directory Server instance. You must set the database instance owner correctly. Otherwise, the **idscfgdb** command fails. For more information about setting up required users and groups, see the *Installing* section in the IBM Security Directory Suite documentation.

You can also configure online backup for a Directory Server instance by using the **idscfgdb** command. After you configure, you cannot unconfigure online backup by using the **idscfgdb** command with the **-c** parameter.

You can configure online backup by using the **idscfgdb** command only during the initial stage of database creation. If **idscfgdb** is used to configure online backup after the database is configured, then the operation might fail. You can use **idscfgdb** to change the DB2 password, unconfigure online backup, or both after the configuration.

Note:

- The **-a**, **-t**, and **-l** parameters must be used only during initial configuration of database.
- The **idscfgdb** command sets the DB2 buffer pools to AUTOMATIC.

The instance owner specifies a database administrator user ID, database administrator password, location to store the database, and the name of the database. The database administrator user ID must exist on the system.

After successfully creating the database, the information is added to the ibmslapd.conf file of the Directory Server instance. If the database and local loopback setting do not exist, they are created. You can create the database as a local code page database, or as a UTF-8 database, which is the default.

Synopsis

```
idscfgdb [-I instance_name
         [-w db_admin_pw] [-a db_admin_id -t db_name -l db_location [-x]]
         [-collate [on|off]]
         [-c ] [-k backup_dir]
         [-s storage_loc]
         [-z ext_size]
         [-f config_file] [-d debug_level] [-b output_file] [-q]
         [-n]] | -v | -?
         [-Y]
         [-P remote_db_server]
         [-S remote_db_port]
         [-u remote_db_user]
         [-p remote_db_pwd]
         [-L]
         [-B kdb_file]
         [-H stash_file]
         [-F]
         [-Z auth_type]
```

Options

The **idscfgdb** command takes the following parameters.

-a *db_admin_id*
Specifies the DB2 administrator ID. The DB administrator must exist on the system and must have the appropriate permissions.

-b *output_file*
Specifies the file name to redirect console output. If you use this parameter with the **-q** parameter, only errors are sent to the *outputfile* file. If debug mode is set, then the debug output also is sent to this file.

Note: The output file is created in the `CustomOut` folder.

-c
Removes the online backup configuration setup of the database, if the online backup was configured at the database configuration stage by using either the **idscfgdb** command.

Note: The **-c** parameter must not be used along with the **-a**, **-t**, and **-l** parameters, if the database is already configured.

-collate [on|off]

- The default value is on if you use it with **-x** for local codepage.

  If the database is in the local codepage, this parameter specifies that the strings must be sorted according to the system locale.

  If the database is UTF-8, specifies that the strings must be sorted by using the UCA (Unicode Collation Algorithm) collation sequence that is based on the Unicode Standard version 4.00 with normalization implicitly set to on. Details of the UCA can be found in the Unicode Technical Standard #10, which is available at the Unicode Consortium Web site (http://www.unicode.org/).
- The default value is `off` for UTF-8 database.

  The strings are sorted in binary order.

-d *debug_level*
Sets the LDAP debug level to *debuglevel*. If you specify this parameter, it sends the debug output to stdout. The *debuglevel* value is a bit mask that controls which output is generated with values from 1 to 65535. For more information about debug levels, see Chapter 4, “Debugging levels ,” on page 131.

-f *config_file*
Specifies the customized configuration file. If this parameter is not specified, the default configuration file for the Directory Server instance is used.

Note: The customized configuration file must be uploaded to `CustomIn` folder before you can use this parameter.

-I *instance_name*
Specifies the name for the Directory Server instance to update.

-k *backup_dir*
Specifies the backup location for the database. You must pass this parameter to configure online backup for the database.

Note: The *backup_dir* directory must exist with appropriate read and write permissions for the database owner. The backup files are created in a *instance_name* subdirectory in *backup_dir*.

backup_dir is created under `CustomOut`.

-l *db_location*
Specifies the DB2 database location. On Linuxsystems, the location is a directory name, for example `/home/<instance_nm>`. The database requires a minimum of 80 MB free space. More disk space might be required for as directory entries are added to database.

-n
Specifies to run in no prompt mode. All output from the command is generated, except for messages that require user interaction. Use this parameter with the **-w** parameter.

-q
Specifies to run in quiet mode. All output from the command is suppressed, except for error messages. If you also specify the **-d** parameter, then the trace output is not suppressed.

-s ***storage_loc***
Specifies the TABLESPACE container location.

-t ***db_name***
Specifies the DB2 database name.

-v
Specifies to show the version information of the command.

-w ***db_admin_pw***
Specifies the DB2 administrator password.

Note: During initial stage of database creation, the value that is specified by using **-w** is validated first with the existing DB2 Administrator password. Then, sets the DB2 Administrator password in the configuration file for the Directory Server instance. This parameter is required if the **-n** parameter is provided.

If the database is already configured, the value that is specified by using **-w** is not validated against the existing DB2 Administrator password. It is used to update the DB2 Administrator password and the change log database owner password (if change log is configured) in the server configuration file. The **-c** parameter can be used with the **-w** parameter. The **-a**, **-t**, and **-l** parameters must not be used for a configured database.

-x ***instancename***
Specifies to create the DB2 database in a local code page.

-z ***ext_size***
Specifies the table space extension size in pages. The default value for the extension size is 8192 pages.

-?
Specifies to show the syntax format.

-Y
Configure Remote Database with the Security Directory Server instance.

-P ***remote_db_server***
Specify the machine name or IP address of the remote DB2 server.

-S ***remote_db_port***
Specify the port of the new Remote DB2 instance.

-u ***remote_db_user***
Remote DB2 instance user id having proper authority.

-p ***remote_db_pwd***
Remote DB2 instance user password.

-B ***kdb_file***
Specify the full path to the kdb_file that is to be used for SSL communication with the remote DB.

-H ***stash file***
Specify the full path to the stash file that is to be used for SSL communication with the remote DB.

-L
Setup SSL communication with Remote Database.

Note: This parameter can only be used when you are configuring the Remote Database.

-F
Forcefully rewrite the VAUUID to the remote database.

You can specify this option only when you are configuring the remote database.

Any other Directory Server virtual appliance that is already configured to the remote database instance fails to start.

-Z auth_type
Specifies the authentication type for connecting to the remote DB2 instance. Acceptable values are **SERVER** or **SERVER_ENCRYPT**. If not specified, the default value is '**SERVER**'.

Examples

Example 1:
To configure a Directory Server instance with a database with the following values, run the `idscfgdb` command.

- Database: `ldapdb2`
- Location: `/home/ldapdb2`
- DB2 database administrator ID: `ldapdb2`
- Password: `secret`

```
idscfgdb -a ldapdb2 -w secret -t ldapdb2 -l /home/ldapdb2
```

If the password is not specified, you are prompted for the password. The password is not shown on the command line when you enter it.

Example 2:
To configure online backup, run the following command:

```
idscfgdb -I instance_name -a db_admin_id -t db_name -w dbadminpw
-l db_location -k backup_dir -n
```

Example 3:
To remove an online backup configuration, run the following command:

```
idscfgdb -I instance_name -c
```

idscfgsch

Use the **idscfgsch** to configure a schema file for a Directory Server instance.

Description

The **idscfgsch** command configures a schema file for a Directory Server instance. The schema file must exist on the system. The Directory Server instance owner must specify the schema file to add in the `ibmslapd.conf` file of a Directory Server instance.

Synopsis

```
idscfgsch [-I instancename -s schemafile [-f configfile] [-d debuglevel]
          [-b outputfile] [-q] [-n]] | -v | -?
```

Options

The **idscfgsch** command takes the following parameters.

-b ***outputfile***
Specifies the full path of a file to redirect console output. If you use this parameter with the **-q** parameter, errors are sent to the *outputfile* file. If debug mode is set, then the output is sent to the *outputfile* file.

-d ***debuglevel***
Sets the LDAP debug level to *debuglevel*. If you specify this parameter, it sends the debug output to stdout. The *debuglevel* value is a bit mask that controls which output is generated with values from 1 to 65535. For more information about debug levels, see Chapter 4, "Debugging levels ," on page 131.

-f ***configfile***
Specifies the full path to the configuration file to update. If this parameter is not specified, the default configuration file of the Directory Server instance is considered.

-I ***instancename***
Specifies the Directory Server instance name for which to configure the schema file.

-n
Specifies to run in no prompt mode. All output from the command is generated, except for messages that require user interaction.

-q
Specifies to run in quiet mode. All output from the command is suppressed, except for error messages. If you also specify the **-d** parameter, then the trace output is not suppressed.

-s ***schemafile***
Specifies the schema file to add to the Directory Server instance.

-v
Specifies to show the version information of the command.

-?
Specifies to show the syntax format.

Examples

Example 1:
To configure the `myschema.oc` schema file from `CustomIn` folder for a Directory Server instance, run the following command:

```
idscfgsch -I instance_name –s myschema.oc
```

idscfgsuf

Use the **idscfgsuf** command to configure a suffix for a Directory Server instance.

Description

The**idscfgsuf** command configures a suffix for a Directory Server instance. The suffix is added the `ibmslapd.conf` file of a Directory Server instance. This command fails when the suffix specified exists in the configuration file.

Synopsis

```
idscfgsuf [-I instancename -s suffix [-f configfile] [-d debuglevel] [-b outputfile]
          [-q] [-n]] | -v | -?
```

Options

The **idscfgsuf** command takes the following parameters.

-b ***outputfile***
Specifies the full path of a file to redirect console output. If you use this parameter with the **-q** parameter, errors are sent to the *outputfile* file. If debug mode is set, then the debug output is sent to this file.

-d ***debuglevel***
Sets the LDAP debug level to *debuglevel*. If you specify this parameter, it sends the debug output to stdout. The *debuglevel* value is a bit mask that controls which output is generated with values from 1 to 65535. For more information about debug levels, see Chapter 4, “Debugging levels ,” on page 131.

-f ***configfile***
Specifies the full path to the configuration file to update. If this parameter is not specified, the default configuration file of the Directory Server instance is considered.

-I ***instancename***
Specifies the Directory Server instance name. If you have multiple Directory Server instances on the system, then you must use this parameter.

-n
Specifies to run in no prompt mode. All output from the command is generated, except for messages that require user interaction.

-q
Specifies to run in quiet mode. All output from the command is suppressed, except for error messages. If you also specify the **-d** parameter, then the trace output is not suppressed.

-s ***sufix***
Specifies to add a suffix to the Directory Server instance.

-v
Specifies to show the version information of the command.

-?
Specifies to show the syntax format.

Examples

Example 1:
To configure the `o=sample` suffix on a system with a single Directory Server instance, run the following command:

```
idscfgsuf -s o=sample
```

Example 2:
To configure the `o=sample` suffix on a system with multiple Directory Server instances, run the following command:

```
idscfgsuf -I instance_name -s o=sample
```

idsdb2ldif

Use the **idsdb2ldif** command to output directory server entries to an LDIF file.

Description

The **idsdb2ldif** command gets entries from a directory and puts it in a text file in LDAP Directory Interchange Format (LDIF). You can run this command against an instance at when the instance is running or stopped.

⚠ **Attention:** You must specify the encryption seed and salt of the destination server for the following conditions:

- If you are importing data to an instance configured for Advanced Encryption Standard (AES) encryption from another instance.
- If the target and the destination servers are not cryptographically synchronized.

For information about cryptographic synchronization of servers, see Synchronizing two-way cryptography between server instances.

Depending on the encryption scheme that is set on the servers, the LDIF file might contain different encrypted values.

- The command takes the following actions when you specify the encryption seed and salt values of the destination server:
  1. Any AES encrypted data is decrypted by using the AES keys of source server.
  2. The data is then encrypted by using the encryption seed and salt values of destination server.

The encryption seed is used to generate a set of AES secret key values. The key values are stored in the stash file of a Directory Server instance. These values are used to encrypt and decrypt stored password and secret key attributes. The encryption seed must contain only printable `ISO-8859-1 ASCII` characters with values in the range 33 - 126, and must be a minimum of 12 and a maximum of 1016 characters in length. For information about ASCII characters, see Chapter 5, "ASCII characters from 33 to 126," on page 133. The encryption salt is a randomly generated value and is used to generate AES encryption keys. You can obtain the salt value of the destination server by searching the `cn=crypto,cn=localhost` entry on destination server. The attribute name is `ibm-slapdCryptoSalt`.

- The SHA encoded directory encryption seed of the source server is written to the LDIF file for reference during import. For parsing purposes, this encryption seed reference is in a `cn=crypto,cn=localhost` pseudo entry, which for information only. This value is not loaded as part of the import.

Synopsis

```
idsdb2ldif | idsdb2ldif [-o output_file -I instance_name [-f config_file]
           [-n filter_DN] [-c comments] [-k ?|key_seed -t key_salt] [-j]
           [-d debug_level] [[-s subtree_DN [-x]] | [-l] [-r]] [-W]] | ?
```

Options

The **`idsdb2ldif`** command takes the following parameters.

`-c` ***comments***
Specifies to add the comments to the output LDIF file.

`-d` ***debug_level***
Sets the debug level to *debug_level*. The **`ldtrc`** command must be running, when you use this parameter.

`-f` ***config_file***
Specifies the full path of the configuration file to use. If not specified, the default configuration file of the Directory Server instance is used.

`-I` ***instance_name***
Specifies the Directory Server instance name from which to export data.

`-j`

Specifies not to export the operational attributes to an LDIF file.

`-k` ***key_seed***
Specifies encryption key seed value of the destination server to use for encryption of password data. A ? provides a separate prompt and console masking of the seed value. You must use this parameter with the **`-t`** parameter.

`-l`

Specifies to export the entries under `cn=localhost`.

`-n` ***filter_DN***
Specifies the DN of filter entry for filtering the entries before you add to output LDIF file. If you specify this parameter, entries that are stored in the database are filtered and then the partial entry is written to the LDIF file. The filtering is done as per filter that is specified in *filter_DN*.

`-o` ***output_file***
Specifies the LDIF file to store the directory entries. All entries from the specified subtree are written in LDIF format to the output file. This parameter is required. If you do not want the file to be created in the current directory, then a file name with full path must be specified.

`-r`

Specifies to export the entries under `cn=Deleted Objects`. If the **`-s`** parameter is also specified, then the subtree DN must be `cn=Deleted Objects`.

-s ***subtree_DN***
Specifies the DN of the top entry of a subtree to be written to the LDIF file. This entry and the descendant entries in the directory hierarchy are written to the file. If this parameter is not specified, directory entries under the suffixes are written to the file.

-t ***key_salt***
Specifies the encryption key salt value of destination server to use for encryption of password data. You must use this parameter with the **-k** parameter.

-W ***output_file***
Specifies the full path of a file in which to redirect output.

-x
Specifies to exclude the nested replication contexts that are present under the subtree that is specified by the **-s** parameter. This parameter cannot be used with the **-l** parameter.

-?
Specifies to show the syntax help.

Examples

Example 1:

To export the data to an LDIF file, run the following command.

```
idsdb2ldif -I instance_name -o without-j.ldif
```

The following output is written to the LDIF file:

```
dn: cn=tom,dc=mycompany,dc=com
control: 1.3.18.0.2.10.19 false::
MIQAAADVMIQAAAAmCgEAMIQAAAAdBAxjcmVhdG9yc05hbWUxhAAAAAkEB0NOPVJPT1QwhAA
AADgKAQAwhAAAAC8ED2NyZWF0ZVRpbWVzdGFtcDGEAAAAGAQWMjAwODAzMDcwMTMyMjcu
MDAwMDAwWjCEAAAAJwoBADCEAAAAHgQNbW9kaWZpZXJzTmFtZTGEAAAACQQHQ049Uk9PV
DCEAAAAOAoBADCEAAAALwQPbW9kaWZ5VGltZXN0YW1wMYQAAAAYBBYyMDA4MDMwNzAx
MzIyNy4wMDAwMDBa
userpassword: {SHA}loNd2L+nGL1kR8zIevia4Wddrso=
objectclass: person
objectclass: top
sn: tom
cn: tom
ibm-entryuuid: 16d448c0-8032-102c-9762-e03d72fe6fad
```

The Directory Server instance has a user entry with the distinguished name `cn=tom, dc=mycompany,dc=com`.

The output contains a control with `OID 1.3.18.0.2.10.19`, a criticality of false, and a base 64 encoded control value. The control is the means by which the operational attributes are sent to the LDIF file. The control information is difficult to understand and read in the resulting LDIF file. The control value is in binary format, which includes information about how to appropriately update the identified operational attributes for the target import.

If you run the **idsdb2ldif** command with the **-j** parameter, the operational attributes are not exported. For example:

```
idsdb2ldif -I instance_name -j -o with-j.ldif
```

The following output is written to the LDIF file:

```
dn: cn=tom,dc=mycompany,dc=com
userpassword: {SHA}loNd2L+nGL1kR8zIevia4Wddrso=
objectclass: person
objectclass: top
sn: tom
cn: tom
ibm-entryuuid: 16d448c0-8032-102c-9762-e03d72fe6fad
```

idsdbback

Use the **idsdbback** command to take a backup of the directory data and configuration files.

Description

The **idsdbback** command takes a backup of the directory data and configuration files. The Administration Server uses this command to process backup requests. For offline backup, the Directory Server must be stopped for the **idsdbback** command to succeed. Also, the Directory Server must be is stopped state when the **-u** parameter is used for the first time. Online backups require a change to the database configuration and an initial offline backup. Subsequent online backup operations can proceed with the Directory Server in running mode.

- Specifying backup location on an NFS mounted partition and restoring from an NFS mounted partition causes the following error.

  ```
  2004-10-07-21:08:00.native retcode = -1026; state = "^A";
      message = "SQL1026N The database manager is already active."
  2004-10-07-21:08:01.native retcode = -2025; state = "^A";
      message = "SQL2025N    An I/O error "6" occurred on media
      "/dbrestore/backup/SVTINST1.0.svtinst1.NODE0000.CATN0000.20041007185"."
  ```

  The **idsdbback** or **idsdbrestore** operation must be done on a local drive or partition only.
- The DB2 level that is used to back up database when the server is offline must be of same version that is used to restore database.
- The **idsdbback** command removes the files from the previous backup after successfully completing a backup. If the **-l** parameter for change log data is not provided or is not configured for the instance and there are existing change log backup files, the existing change log backup files are removed.
- The Directory Server instance must be stopped when the **-a** parameter is used to specify a new log archive directory. DB2 requires all applications to be disconnected from the database before the changes take effect. Any other applications that are connected to this database must also be disconnected. If the **-a** parameter is specified without the **-k** parameter, then the archive path is changed in the DB2 configuration but no backup is taken. The archive path gets applied to future online backups.

Synopsis

```
idsdbback | idsdbback -I instancename -k backupdir [-d debuglevel] [-b outputfile]
                    [-q] [-n][[-l] [-u [-a archive_dir]] | [-x]] | -v | -?
```

Options

The **idsdbback** command takes the following parameters.

-a ***archive_dir***
Specifies the directory for configuring online backup and to save inactive log files. For the first online backup, if this parameter is not specified, the value of *backupdir* is used. For subsequent backups, the configuration is not changed unless this parameter is specified. The **-a** parameter can be specified only with **-u** for online backups.

-b ***outputfile***
Specifies the full path of a file to redirect console output. If you use this parameter with the **-q** parameter, errors are sent to the *outputfile* file. If debug mode is set, then the debug output is sent to this file.

-d ***debuglevel***
Sets the LDAP debug level to *debuglevel*. If you specify this parameter, it sends the debug output to stdout. The *debuglevel* value is a bit mask that controls which output is generated with values from 1 to 65535. For more information about debug levels, see Chapter 4, “Debugging levels ,” on page 131.

-I ***instancename***
Specifies the Directory Server instance name for which you want to run the backup operation.

-k ***backupdir***
Specifies the folder to use to back up the database.

Note: When you take multiple backups, ensure that each backup stored is in a separate directory. If you have more than one version of database backup file in the same directory, the **idsdbrestore** command restores only the database with the most recent timestamp.

-l
Specifies to include change log data for backup, if change log configured.

-n
Specifies to run in no prompt mode. All output from the command is generated, except for messages that require user interaction.

-q
Specifies to run in quiet mode. All output from the command is suppressed, except for error messages. If you also specify the **-d** parameter, then the trace output is not suppressed.

-v
Specifies to show the version information of the command.

-u
Specifies to use online backup. For the first time, it requires Directory Server to be offline.

-x
Specifies not to back up database files, indicating a proxy backup.

-?
Specifies to show the syntax format.

Examples

Example 1:
To take an offline backup of the database, configuration, and schema files, run the following command:

```
idsdbback -I instance_name -k backupdir
```

Example 2:
To take an online backup of server, run the following command:

```
idsdbback –n –I instance_name -b outputfile -u -k backupdir
```

Example 3:
To take an online backup with a changed archive path, issue the following command:

```
idsdbback -n -I instance_name -b outputfile -k backupdir –u –a archive_dir
```

Example 4:
To take an online backup for a Directory Server with change log data, run the following command:

```
idsdbback -I instance_name –k backupdir –u –l -n
```

Example 5:
To take a backup of a Proxy Server, run the following command:

```
 idsdbback –I proxy_name –k backupdir -x -n
```

idsdbmaint

Use the **idsdbmaint** command to do database maintenance activities for a Directory Server instance.

Description

The **idsdbmaint** command runs DB2 maintenance activities on the database that is associated with a Directory Server instance. The DB2 maintenance activities include DB2 index reorganization, DB2 row compression on tables, and DB2 table space conversion.

Note:

- The Directory Server instance must be in stopped state before you run the **idsdbmaint** command.

Synopsis

```
idsdbmaint [-I instance_name [-b outputfile] [-d debuglevel]] |
           [ -i ] | [ -r ] | -h | -?
```

Options

The **idsdbmaint** command takes the following parameters.

-b ***outputfile***
Specifies the full path of a file to redirect output. If debug mode is set, the debug output is sent to this file.

-d ***debuglevel***
Sets the debug level to *debuglevel*. The **ldtrc** command must be running, when you use this parameter.

-I ***instance_name***
Specifies the Directory Server instance name.

-i
Specifies to run index reorganization on the database that is associated with the Directory Server instance.

-r
Specifies to run row compression on the database that is associated with the Directory Server instance.

-h | -?
Specifies to show the usage.

Examples

Example 1:
To do index reorganization, run the **idsdbmaint** command with the following parameters:

```
idsdbmaint –I instance_name -i
```

Example 2:
To inspect the tables and to run row compression, run the **idsdbmaint** command with the following parameters:

```
idsdbmaint –I instance_name -r
```

The command does row compression only if the compression would result in more than 30% space benefit.

idsdbrestore

Use the **idsdbrestore** command to restore a database and configuration files for a Directory Server instance.

Description

The **idsdbrestore** command restores database and configuration files for a Directory Server instance when the instance is offline. You must stop the instance before you run the **idsdbrestore** command.

- Specifying backup location on an NFS mounted partition and restoring from an NFS mounted partition causes the following error.

  ```
  2004-10-07-21:08:00.native retcode = -1026; state = "^A";
      message = "SQL1026N    The database manager is already active."
  2004-10-07-21:08:01.native retcode = -2025; state = "^A";
      message = "SQL2025N    An I/O error "6" occurred on media
      "/dbrestore/backup/SVTINST1.0.svtinst1.NODE0000.CATN0000.20041007185"."
  ```

  The **idsdbback** or **idsdbrestore** operation must be done on a local drive or partition only.
- You can run the **db2 rollforward** command when you restore from an online backup. After the restore operation and before you start the server, run the **db2 rollforward** command.

  ```
  db2 rollforward db dbname to end of logs and stop
  ```

  You must run this command if you get the following errors.

  ```
  SQL1117N A connection to or activation of database dbname cannot be
      made because of ROLL-FORWARD PENDING.
  ```
- When you restore from an online backup, the **idsdbrestore** command attempts to restore from the online backup image. This image is in the backup directory path that is specified by using the **-k** parameter. At any time, only one online backup image is in existence and only that online backup image must be used for the restore operation.
- When you run **idsdbrestore** with **-x**, you might see unexpected results if the backed up configuration file and the configuration file of the instance to restore are inconsistent. For example:
  - Server type mismatch (RDBM/PROXY). For example, restoring from inst1 an instance with RDBM to inst1 a proxy instance by using **idsdbrestore -x**.
  - Matching server type but server name mismatch. For example, restoring from inst1 an instance with RDBM to inst2 an instance with RDBM by using **idsdbrestore -x**.

Synopsis

```
idsdbrestore | idsdbrestore -I instancename -k backupdir [-d debuglevel]
             [-b outputfile] [-r] [-q] [-n][[-l] | [-x]]] | -v | -?
```

Options

The **idsdbrestore** command takes the following parameters.

-b ***outputfile***
Specifies the full path of a file to redirect console output. If you use this parameter with the **-q** parameter, errors are sent to the *outputfile* file. If debug mode is set, then the debug output is sent to this file.

-d ***debuglevel***
Sets the LDAP debug level to *debuglevel*. If you specify this parameter, it sends the debug output to stdout. The *debuglevel* value is a bit mask that controls which output is generated with values from 1 to 65535. For more information about debug levels, see Chapter 4, “Debugging levels ,” on page 131.

-I ***instancename***
Specifies the Directory Server instance name for which you want to restore the database and configuration and schema files.

-k ***backupdir***
Specifies the directory from which to restore. The **idsdbrestore** command restores a database into a database and database instance with the same name from a database backup location.

-l
Specifies to include change log data for restore, if change log configured.

-n
Specifies to run in no prompt mode. All output from the command is generated, except for messages that require user interaction.

-q
Specifies to run in quiet mode. All output from the command is suppressed, except for error messages. If you also specify the **-d** parameter, then the trace output is not suppressed.

-r
Specifies not to restore the `ibmslapd.conf` file.

-v
Specifies to show the version information of the command.

-x
Specifies not to restore database files, indicating a proxy restore.

-?
Specifies to show the syntax format.

Examples

Example 1:
To restore a database, configuration files, and schema files for a Directory Server instance, run the following command:

```
idsdbrestore -I instance_name -k /backupdir
```

Example 2:
To restore a Proxy Server instance, run the following command:

```
idsdbrestore –I proxy_instance –k /backup_dir –x -n
```

Example 3:
To restore a Directory Server instance and the change log data for the instance, run the following command:

```
idsdbrestore -I instance_name -l -k /backupdir
```

idsdnpw

Use the **idsdnpw** to set the administration DN and administrative password for an instance.

Description

The **idsdnpw** command sets or changes the administrator DN and password for a directory server instance.

The command can be run both when the Directory Server instance is in the stopped state or when it is running. If the directory server is running, the tool internally reloads the config extended operation for which the correct port must be specified with the **-P** parameter.When an administrator specifies an administrator password and an administrator DN, which is optional, the command writes these values to the `ibmslapd.conf` file. If the administrator DN is not specified, it is set to `cn=root` by default.

Synopsis

```
idsdnpw [-I instancename [[-u user_DN] -p password] [-f config_file] [-d debug_level]
        [-b output_file] [-q] [-n]] [-P port] | -v | -?
```

Options

The **idsdnpw** command takes the following parameters.

-b ***output_file***

Specifies the full path of a file to redirect console output. If you use this parameter with the **-q** parameter, errors are sent to the *output_file* file. If debug mode is set, then the debug output is also sent to this file.

-d ***debug_level***

Sets the LDAP debug level to *debug_level*. If you specify this parameter, it sends the debug output to stdout. The *debug_level* value is a bit mask that controls which output is generated with values from 1 to 65535. For more information about debug levels, see Chapter 4, “Debugging levels ,” on page 131.

-f ***config_file***

Specifies the full path to the configuration file to update with administration DN and password values. If this option is not specified, the default configuration file for the Directory Server instance is used.

-I ***instancename***

Specifies the Directory Server instance name. This parameter is required if there are Directory Server instances on the system.

-n

Specifies to run in no prompt mode. All output from the command is generated, except for messages that require user interaction. This parameter must be used with the **-p** parameter.

-p ***password***

Specifies to change the directory administrator password. If an administrator DN value is not specified by using the **-u** parameter, the current value of the administrator DN is used. If the administrator DN is not defined, then the default value, `cn=root`, is used. This parameter is required when the **-n** parameter is specified.

-P ***port***

Specifies the port number of the Directory Server instance.

This parameter is optional. If the port number is not specified, 389 is used as the default port number.

-q

Specifies to run in quiet mode. All output from the command is suppressed, except for error messages. If you also specify the **-d** parameter, then the trace output is not suppressed.

-u ***user_DN***

Specifies to create or change the directory administrator distinguished name (DN).

-v

Specifies to show the version information of the command.

-?

Specifies to show the syntax format.

Examples

Example 1:

To set the administrator DN to `cn=myname` and the password to `secret`, run the following command:

```
idsdnpw –u cn=myname –p secret
```

If the password is not specified, you are prompted for the password.

Note: The administrator password must conform to the administration password policy requirements, if the administration password policy is set.

idsenvvars

Use the **idsenvvars** command to manage the environment variables for Directory Server.

Description

The **idsenvvars** command enables you to add, modify, delete, or list environment variables and their values.

Synopsis

```
idsenvvars [[-a | -m] variable_name value] | [-d variable_name] | -l | [-v value]
```

Options

Use the following parameters with the **idsenvvars** command:

-a *variable_name value*
Adds an environment variable with the specified name and value.

-m *variable_name value*
Modifies the value of the specified environment variable.

-d *variable_name*
Deletes the specified environment variable.

-l
Lists all the environment variables.

-v *value*
Specifies the value of the variable that is to be added or modified.

Examples

Example 1:
To add an environment variable with the specified name and value, run the following command:

```
idsenvvars -a variable_name value
```

Example 2:
To modify the value of an environment variable, run the following command:

```
idsenvvars -m variable_name new_value
```

Example 3:
To delete the specified environment variable, run the following command:

```
idsenvvars -d variable_name
```

Example 4:
To view a list of all environment variables, run the following command:

```
idsenvvars -l
```

Example 5:
To add an environment variable with the specified name and value, run the following command:

```
idsenvvars -a variable_name -v value
```

Example 6:
To modify the value of the specified environment variable, run the following command:

```
idsenvvars -m variable_name -v value
```

idsgendirksf

Use the **idsgendirksf** command to regenerate a key stash file for a Directory Server instance.

Description

The **idsgendirksf** command uses the encryption seed and salt values of an instance to regenerate a key stash file for an instance. The encryption seed is the seed value that you supplied when you created the instance. The encryption salt value can be obtained by searching the `cn=crypto,cn=localhost` entry in the instance. The attribute that hold salt value is `ibm-slapdCryptoSalt`. The encryption seed and salt values are used to regenerate the `ibmslapddir.ksf` file for an instance.

If you use characters that have special meaning to the shell program in the encryption seed or salt, then you must use the escape character before such characters. To determine the acceptable character set for encryption seed and salt values, see Chapter 5, "ASCII characters from 33 to 126," on page 133.

Synopsis

```
idsgendirksf [-s salt [-e encrypt_seed] -l location
             [-d debug_level] [-b output_file] [-q] [-n]] | -v | -?
```

Options

The **idsgendirksf** command takes the following parameters.

-b ***output_file***
Specifies the full path of a file to redirect console output. If you use this parameter with the **-q** parameter, errors are sent to the *output_file* file. If debug mode is set, then the debug output is sent to this file.

-d ***debug_level***
Sets the LDAP debug level to *debug_level*. If you specify this parameter, the command sends the debug output to `stdout`. The *debug_level* value is a bit mask that controls which output are generated with values 1 - 65535. For more information about debug levels, see Chapter 4, "Debugging levels ," on page 131.

-e ? | ***encrypt_seed***
Specifies the encryption seed value that was used to create the directory key stash file of the server. The encryption seed must contain only printable `ISO-8859–1 ASCII` characters with values in the range of 33 to 126. The encryption seed must be a minimum of 12 and a maximum of 1016 characters in length. For more information about acceptable characters, seeChapter 5, "ASCII characters from 33 to 126," on page 133. To generate a password prompt, use **?**. The password prompt prevents your encryption seed from being visible through the **ps** command.

-l ***location***
Specifies the location to create the directory key stash file.

-n
Specifies to run in no prompt mode. All output from the command is generated, except for messages that require user interaction.

-q
Specifies to run in quiet mode. All output from the command is suppressed, except for error messages. If you also specify the **-d** parameter, then the trace output is not suppressed.

-s ***encryption_salt***
Specifies the encryption salt value that is used to create the directory key stash file. The encryption salt value can be obtained by searching the `cn=crypto,cn=localhost` entry in the instance. The attribute that hold salt value is `ibm-slapdCryptoSalt`.

-v
Specifies to show the version information of the command.

-?

Specifies to show the syntax help.

Examples

Example 1

To regenerate the key stash file for the Directory Server instance, run the following command. For example:

```
idsgendirksf -e mysecretseed –s mysecretsaltvalue
```

When the command runs successfully, the following changes occur:

- `ibmslapddir.ksf` file is generated in the `CustomOut` directory.
- You can choose to copy the generated `ksf` file to the instance's `etc/` directory. If you do this action, then only the `ibmslapddir.ksf` file is copied to the instance's `etc/` directory and the ownership of the `ibmslapddir.ksf` file in the instance `etc/` folder is updated to *`directory_server_instance owner:instance_owner_group`*.

idsilist

Use the **idsilist** command to list directory server instances on the system.

Description

Based on the parameter that is used, the command lists a Directory Server instance that exist on the system. The command retrieves detailed information about instance on a system.

Note:

Synopsis

```
idsilist [[-I instance_name][-a | -r] [-d debuglevel] [-b outputfile]] | -v | -?
```

Options

The **idsilist** takes the following parameters.

-a

Specifies to list the full information about each instance on the system. This parameter cannot be used with the **-r** parameter.

-b *outputfile*

Specifies the full path of a file to redirect console output. If debug mode is set, then the debug output is sent to this file.

-d *debuglevel*

Sets the LDAP debug level to *debuglevel*. If you specify this parameter, the command sends the debug output to stdout. The *debuglevel* value is a bit mask that controls which output is generated with values from 1 to 65535. For more information about debug levels, see Chapter 4, “Debugging levels ,” on page 131.

-I *instance_name*

Specifies the Directory Server instance name for which to list instance information.

-r

Specifies to list the full information about each instance on the system. This parameter shows the same information as the **-a** parameter, but the information is printed in a raw format. The information about each instance is printed on a separate line and each data item is separated by a number sign (#). This parameter cannot be used with the **-a** parameter.

-v

Specifies to show the version information of the command.

-?
Specifies to show the syntax format.

Examples

Example 1:
To list details about the Directory Server instance, `sdsinst1`, run the following command:

```
idsilist -I sdsinst1
```

The command generates the following output:

```
Directory Server instance(s):
sdsinst1
```

You can also use the **-a** or **-r** parameter with the **-I** *instance_name* parameter to get the detailed information about the instance. For example: **idsilist -I sdsinst1 -a** or **idsilist -I sdsinst1 -r**.

Example 2:
To list complete details about instances, run the **idsilist** command along with the **-I** and **-a** parameters. For example:

```
idsilist -I sdsinst1 -a

Directory Server instance(s):

--------------------------------------
Name: sdsinst1
Version: 8.0.1
Location: /home/sdsinst1
Description: IBM Security Directory Suite Instance V8.0.1
IP Addresses: All available
Port: 4389
Secure Port: 4636
Admin Server Port: 3544
Admin Server Secure Port: 3545
Type: Directory Server
```

Example 3:
To list complete details about instances without description for each value, run the **idsilist** command along with the **-I** and **-r** parameters. For example:

```
idsilist -I sdsinst1 -r

Directory Server instance(s):
sdsinst1#8.0.1#/home/sdsinst1# IBM Security Directory Suite Instance V8.0.1#
All available #4389#4636#3544#3545#Directory Server
```

idsldif2db

Use the **idsldif2db** command to load entries from an LDIF file to a database.

Description

You can run the **idsldif2db** command to load entries that are specified in the LDAP Directory Interchange Format (LDIF) file into a DB2 database that is associated with a Directory Server instance. The database to which you want to load entries must exist. The **idsldif2db** command can be used to add entries to an empty directory database or to a database that already contains entries.

Note:

1. You must stop the Directory Server before you use the server import utilities.
2. Ensure that no applications are attached to the directory database. If there are applications that are using the database, the server utilities might fail.

3. The **idsldif2db** command recognizes the operational attributes `creatorsname`, `modifiersname`, `modifytimestamp`, and `createtimestamp` if they are in plain text format.

If the parameters provided to the command are incorrect, a syntax error message is shown after which the correct syntax is shown.

Attention: You must specify the encryption seed and salt of the destination server for the following conditions:

- If you are importing data to an instance configured for Advanced Encryption Standard (AES) encryption from another instance.
- If the target and the destination servers are not cryptographically synchronized.

For more information about cryptographic synchronization of servers, see Synchronizing two-way cryptography between server instances.

Note: The SHA encoded directory encryption seed of the source server is written to the LDIF file by using **idsdb2ldif** is for reference during import. For parsing purposes, this encryption seed reference is in the `cn=crypto,cn=localhost` pseudo entry, which is for information only. This value is not loaded as part of the import.

Synopsis

```
idsldif2db | idsldif2db [-i inputfile -I instancename [-f configfile]
           [-d debuglevel] [-r yes | no] [-g] [-W]] | [?]
```

Options

The **idsldif2db** command takes the following parameters.

-d ***debuglevel***
Sets the LDAP debug level to *debuglevel*. If you specify this parameter, the command sends the debug output to stdout. The *debuglevel* value is a bit mask that controls which output is generated with values from 1 to 65535. For information about debug levels, see Chapter 4, "Debugging levels ," on page 131.

-f ***configfile***
Specifies the full path of the configuration file to use. If not specified, the default configuration file of the Directory Server instance is used.

-g
Specifies not to strip the trailing spaces on attribute values.

-i ***inputfile***
Specify the name of the LDIF file that contains directory entries in LDIF format. This parameter is required. If the file is not in the current directory, you must specify the absolute path with the file name.

-I ***instancename***
Specifies the Directory Server instance name to which to load entries.

-r [yes|no]
Specifies whether to replicate. The default is **yes**, which indicates that the entries are put in the change table and are replicated when the server restarts.

-W ***outputfile***
Specifies the full path of a file in which to redirect output.

-?
Specifies to show the syntax format.

Examples

Example 1:

To load the `sample.ldif` file from `CustomIn` folder that is included with IBM Security Directory Suite, run the following command:

```
idsldif2db -i sample.ldif
```

idslogmgmt

Use the **idslogmgmt** command to start or stop the Directory Server log management tool.

Description

The **idslogmgmt** starts or stops the Directory Server log management tool.

In IBM Security Directory Suite virtual appliance, the **idslogmgmt** service is started by default.

To change the log file settings, use the "idsenvvars" on page 100 command. The following environment variables can be set to specify threshold size and maximum number of archives for the administrative tool and the **idslogmgmt** tool logs.

- `IDSADM_SIZE_THRESHOLD`
- `IDSADM_ARCHIVES`
- `IDSLMG_SIZE_THRESHOLD`
- `IDSLMG_ARCHIVES`
- `IDSLMG_CHECK_INTERVAL`
- `IDSLMG_LOG_LEVEL`

However, in virtual appliance, the log file paths cannot be changed. Hence, even if you set the following environment variables, they are not honored:

- `IDSLMG_LOG_PATH`
- `IDSLMG_ARCHIVE_PATH`
- `IDSADM_ARCHIVE_PATH`

The path to the log file is `/home/sdsinst1/idsslapd-sdsinst1/etc/logmgmt/idslogmgmt.log`. You can view the log file by using the virtual appliance console option **Manage** > **Maintenance** > **Log Retrieval and View**. See Retrieving log files.

The path to the archived logs is `/userdata/directory/CustomOut/archived_logs`. You can download archived logs by using the **Configure** > **Advanced Configuration** > **Custom File Management**. See Managing custom files.

Synopsis

The syntax for the **idslogmgmt** command:

```
idslogmgmt [-k] [-n]
```

Options

The **idslogmgmt** command takes the following parameters.

-k
Stops the Directory Server log management tool.

-n
Starts the Directory Server log management tool.

Examples

Example 1:

To start the Directory Server log management tool, run the following command:

```
idslogmgmt -n
```

Example 2:

To stop the Directory Server log management tool, run the following command:

```
idslogmgmt -k
```

idsmonitor

Use the **idsmonitor** script to gather monitoring data while Directory Server is running. You can use this monitoring data to troubleshoot resource usage.

Description

The **idsmonitor** shell script collects system monitoring data about the performance and resource usage of the **ibmslapd** command and related DB2 processes.

If anonymous binds are allowed on the Directory Server and default ports are used, you can run the script without any options.

```
idsmonitor &
```

The current version of Directory Server must be installed on the system to run this script. The script finds the current version of Directory Server and binds anonymously. It attempts to detect whether an **ibmslapd** process is running. By default, it connects to the server on port 389 and writes the output to the default location `idsmonitor.out`.

You also can use the script with specific options if only SSL connections are allowed to the server, or anonymous binds are disabled, or both.

You can stop this script with one of the standard shell commands:

1. Press `Control C` in the shell where it is running.
2. If it is running in the background, run **fg** to bring it to the foreground and press `Control C`.
3. Run the **kill** command with the process ID number (PID): `kill` *pid*.

To check the progress of the script, run the following command:

```
tail -f /path/to/outputfile
```

Synopsis

```
idsmonitor [-h][-D admin][-w passwd][-Z][-P passwd]
                 [-K kdbfile][-H host|IP][-x][-d delay][-p port][-s][-X]
                 [-o output_file][-s][-r][-m][-n][-l num][-v][-V version]
```

Options

-h

Shows the usage.

-D *admin*

Specifies the administrator Distinguished Name (DN) when anonymous binds are disabled.

-w *passwd*

Specifies the corresponding password for the administrator DN.

-H *host* or *IP_address*
Specifies the address for the connection.

-Z
Specifies that the connection is an SSL connection.

If you use this option, you must also specify the **-P** and **-K** options.

-P *password*
Specifies the password for the key database (KDB) file.

-K *file*
Specifies the full path to the KDB file.

-x
Indicates that the script must run in debug mode.

-X
Indicates that the script must exit while monitoring if **ibmslapd** stops or fails.

-d *delay*
Specifies the delay in seconds between running monitor commands.

-p *port*
Specifies the LDAP server port if it is not the default server port, 389.

-s
Indicates that the script must run in silent mode.

Only errors are displayed in the command window.

-o file
Specifies the path and file name of the output file

The default is `idsmonitor.out`.

-V version
Specifies the version commands to run for multi-version installations.

-r
Enables searching for replication status attributes.

-m
Enables querying memory leak-specific information.

-n
Specifies a different LDAP port to the network statistic tool (**netstat**) for the **grep** command to run searches.

-l *n*
Specifies that the script must run *n* iterations before it quits.

-v
Shows the version of the current script.

Examples

Example 1:

To write the output to a location other than the default location, run the script with the following options:

```
idsmonitor -o idsmonitor.out &
```

Example 2:

If only SSL connections are allowed, run the script with the **-Z**, **-P**, and **-K** options:

```
idsmonitor -Z -P secret -K /opt/certs/ldap.kdb &
```

Example 3:

If anonymous binds are disabled, run the script with the **-D** and **-w** options:

```
idsmonitor -D cn=root -w secret -o idsmonitor.out &
```

When you specify ? for the **-w** option, the script prompts the user for the password. You cannot redirect the standard output and standard error when you run the script with this option and value. If you do, the script appears to hang indefinitely because it is waiting for a value to be entered.

Example 4:

The default interval for data sampling is 300 seconds (5 minutes). If the data needs to be sampled more often, then run the script with the following options:

```
idsmonitor -D cn=root -w secret -d 60 -o idsmonitor.out
```

Example 5:

To redirect all output to a debug file instead of displaying it in the standard output console, run the script with the following option:

```
idsmonitor -D cn=root -w secret -o idsmonitor.out > idsmonitor.dbg 2>&1 &
```

The script attempts to print errors to standard output. If you run the script with these options, it might prevent the user from noticing a problem.

Example 6:

To run replication status searches if an admin DN and password were specified, run the script with the **-r** option and specify the admin DN and password:

```
idsmonitor -D cn=root -w secret -p 1389 -r
```

Example 7:

To collect memory-leak debug information on Linux systems, run the script with the **-m** option:

```
idsmonitor -D cn=root -w secret -p 1389 -m
```

This command uses `pmap` *`pid`* on Linux systems.

Example 8:

To run only a specific number of iterations of data collection, run the script with the **-l** option. For example, if you want to collect only an hour of monitoring data, run the following command:

```
idsmonitor -D cn=root -w secret -p 1389 -l 12
```

As the default cycle is 300 seconds, 12 iterations collect 60 minutes of monitoring data.

Example 9:

To use a bind DN and password and to hide the password from **ps** command output, run the script with the following options:

```
idsmonitor -D cn=root -w ? -d 30 -o idsmonitor.out
```

The script prompts you for the password, so you cannot run the script as a daemon process. If you do, the script appears to hang indefinitely because it is waiting for a value to be entered.

Example 10:

To run this script against the non-SSL port, but to use **grep** for the SSL port in the **netstat** output, use the following options:

```
idsmonitor -D cn=root -w ? -p 636 -d 30 -o idsmonitor.out
```

idsperftune

Use the **idsperftune** command to tune your Directory Server performance.

Description

Administrators can use the **idsperftune** command to achieve a higher directory performance by tuning caches, DB2 buffer pools, and DB2 parameters. The command can be run in basic mode, by using the **-B** parameter. The basic tuning can be run before you use an instance or after the instance is in use for a long time. The advanced mode, with **-A** parameter, can be run only after the instance is subjected to a typical workload. The advanced tuning analyzes DB2 performance metrics and makes recommendations for fine-tuning database parameters. The **idsperftune** command provides recommendations for DB2 parameters in the `perftune_stat.log` file in following format.

```
# DB2 parameters=Current Value:Recommendation
# Recommendation can be Not Collected|OK|Increase|Decrease
```

An example with the suggested action.

```
PCKCACHESZ=1533:Increase
```

In this example, you can increase the value of PCKCACHESZ based on the recommendation.

The **idsperftune** command stores the Directory Server and DB2 database parameters values as initial parameters in the `perftune_stat.log` file. These parameters are stored under the section `INITIAL TUNING PARAMETER VALUE ( Prior to First Update Operation )` in the log file. These values do not change later and are recorded in the format: `I_<...>`. The **idsperftune** command stores the old values of Directory Server and DB2 database parameters in the `perftune_stat.log` file. These values are stored under the section `OLD DB2 PARAMETER VALUE ( Prior to last Update Operation )` in the log file. These values are recorded in the format: `O_<..>`.

Note:

- The operation of **idsperftune** depends on a list of values from the administrator, which if not specified are set to their default values. The command accepts the property file, `perftune_input.conf`, and is the only mode of input from the administrator. The property file includes a list of values as `attribute-value` pairs. An administrator must update all the attribute values as per the requirement and run the command by providing the `perftune_input.conf` property file as input.
- The **idsperftune** command does basic tuning where the directory cache size is calculated based on the input from administrator. The command also runs advanced tuning, where the health of DB2 parameter is computed. Administrator must consider the computed size of directory cache and DB2 parameter health values that are updated in the `perftune_stat.log` property file.
- Based on the DB2 parameters changes in the log file, you can run the **idsperftune** command to update the DB2 parameter values. The **idsperftune** command logs the old value of each DB2 parameter before it updates the new value, which can be used for later reference.
- The property files are at the following locations.
  - *instance-home*/idsslapd-*inst-name*/etc/perftune_input.conf
  - *instance-home*/idsslapd-*inst-name*/logs/perftune_stat.log
- You can set the value of the *SYS_MEM_AVL* variable to `false` after you finish running the **idsperftune** command. If the value is `false`, it indicates that there is not sufficient memory available on the system to cache all the entries in Directory Server entry cache. In this case, you must consider increasing the memory to be used or consider reducing the number of entries by using the **-E** parameter.
- By default, the **idsperftune** command uses 90 percent of the system memory and tries to cache 80 percent of the entries.
- The **idsperftune** command uses the default port, for example 389. To specify a port number other than the default port number, you must use the **-p** parameter. The **idsperftune** command does not use the port number from the configuration file.

Synopsis

```
idsperftune -I instance_name -B | -A | [-u -B -p port][-u]
             [-i property_file] [-s] [-m ][-o] [-b output_file]
             [-f config_file] [-E entry_cache_pct]
             [-F filter_cache_size][-d debug_level] [-v | -?]
```

Options

The **idsperftune** command takes the following parameters.

-A
: Specifies to run advanced tuning of DB2 configuration.

-B
: Specifies to run basic tuning of Directory Server cache and DB2 buffer pools.

-b ***output_file***
: Specifies the full path of a file in which to redirect output. If debug mode is set, the debug output is sent to this file.

-d ***debuglevel***
: Sets the debug level.

-E ***entrycache_size***
: Sets the target percentage of entries to be cached.

-F ***filtercache_size***
: Sets the size of filter cache.

-f ***configfile***
: Specifies the full path of the server configuration file.

-I ***instance_name***
: Specifies the name of the Directory Server instance to tune.

-i ***property_file***
: Specifies the property file, which contains tuning parameters.

-m
: Sets the monitor switches for BUFFERPOOL and SORT. If used with **-A**, it captures database snapshot after a time interval of 5 minutes.

-o
: Disables monitor switches for BUFFERPOOL and SORT.

-p ***port***
: Specifies the port number to use for the instance.

-s
: Sets the default value for the total number of entries and average entry size in the file that is based on directory content.

-u
: Updates DB2 and Directory Server cache configuration settings.

-v
: Prints the version information about the command.

-?
: Specifies to show the syntax format.

Examples

Example 1:

To update the file with total entries and average entry size, run the **idsperftune** command with the following parameters:

```
idsperftune –I instance_name -s
```

You can use the values that are generated from the **idsperftune** command with the `-s` parameter to estimate the growth in directory server.

Example 2:

To run basic tuning on the `myinst` Directory Server, run the following command:

```
idsperftune –I myinst –i property_file -B –u
```

In the **-u** parameter is specified, the server and database instance is updated with the suggested LDAP cache and DB2 buffer pool values. If specified without the **-u** parameter, then the suggested settings are updated in the `perftune_stat.log` file only.

Example 3:

To run advanced tuning on the `myinst` directory server, run the following command:

```
idsperftune –I myinst –i property_file -A –m
```

If you use the **-u** parameter with the command, monitor switches for `BUFFERPOOL` and `SORT` are set.

Example 4:

To get basic tuning recommendations, run the **idsperftune** command with the following parameters:

```
idsperftune –I instance_name -B
```

Example 5:

To update the database with the suggested parameters during the basic tuning, run the **idsperftune** command with the following parameters:

```
idsperftune –I instance_name -B –u
```

Or

```
idsperftune –I instance_name –u
```

Example 6:

To get advanced tuning recommendations without turning the monitor switches `ON`, run the **idsperftune** command with the following parameters:

```
idsperftune –I instance_name -A
```

Example 7:

To update the database with the suggested DB2 parameters during advanced tuning without turning the monitor switches `ON`, run the **idsperftune** command with the following parameters:

```
idsperftune –I instance_name -A –u
```

Example 8:

To get advanced tuning recommendations and to turn the monitor switches `ON`, run the **idsperftune** command with the following parameters:

```
idsperftune –I instance_name -A -m
```

The monitor switches are turned `OFF` after the command completes its operation.

Example 9:

To update the database with the suggested DB2 parameters during the advanced tuning and to turn the monitor switches `ON`, run the **idsperftune** command with the following parameters:

```
idsperftune –I instance_name -A –u -m
```

The monitor switches are turned `OFF` after the tool completes its operation.

Example 10:

To turn on the monitor flags for DB2 parameters, run the **idsperftune** command with the following parameters:

```
idsperftune –I instance_name -m
```

Example 11:

To turn off the monitor flags for DB2 parameters, run the **idsperftune** command with the following parameters:

```
idsperftune –I instance_name -o
```

idsrunstats

Use the **idsrunstats** command to optimize the database of a Directory Server instance.

Description

The **idsrunstats** command updates the statistics about the physical characteristics of the tables and the associated indexes in the database. These characteristics include number of records, number of pages, and average record length. The optimizer uses these statistics when it determines the access paths to the data. This command must be run when a table is updated many times, or after you reorganize a table.

Note: The **idsrunstats** command can be run even if the Directory Server is in running mode.

Synopsis

```
idsrunstats | idsrunstats [-I instancename [-f configfile] [-d debuglevel]] | -v | -?
```

Options

The **idsrunstats** command takes the following parameters.

-d ***debuglevel***

Sets the LDAP debug level to *debuglevel*. If you specify this parameter, the command sends the debug output to stdout. The *debuglevel* value is a bit mask that controls which output is generated with values from 1 to 65535. For information about debug levels, see Chapter 4, “Debugging levels ,” on page 131.

-f ***configfile***

Specifies the full path of the configuration file to be update. If this parameter is not specified, the default configuration file of the Directory Server instance is used.

-I ***instancename***

Specifies the Directory Server instance name to update.

-v

Specifies to show the version information of the command.

-?

Specifies to show the syntax format.

Examples

Example 1:

To optimize the database that is associated with an instance, run the **idsrunstats** command:

```
idsrunstats -I instancename
```

idssethost

Use the **idssethost** command to set IP addresses for a Directory Server instance to bind.

Description

This command sets the IP addresses so that a particular Directory Server instance can bind to it. The administrator specifies a Directory Server instance name and a list of IP addresses. If the Directory Server instance and the Administration Server of the instance is running, then you must stop the processes before you update. The**idssethost** command does not allow the IP addresses to be changed, if another instance is using the same ports on the specified IP addresses. The command replaces all of the current IP addresses that are configured for the Directory Server instance. If you specify to listen on all available IP addresses, the IP address attribute is removed from the configuration file.

Note: Before you use **idssethost** to configure Directory Server to bind to an IP address, you must enable an application interface for Directory Server by using the **Application Interfaces** page of the virtual appliance console. See Managing application interfaces.

Synopsis

```
idssethost [-I instance_name –i ip_address [-d debuglevel]
          [-b outputfile] [-q] [-n]] | -v | -?
```

Options

The **idssethost** command takes the following parameters.

-b ***outputfile***
Specifies the full path of a file to redirect console output. If you use this parameter with the **-q** parameter, errors are sent to the *outputfile* file. If debug mode is set, then the debug output is sent to this file.

-d ***debuglevel***
Sets the LDAP debug level to *debuglevel*. If you specify this parameter, the command sends the debug output to stdout. The *debuglevel* value is a bit mask that controls which output is generated with values from 1 to 65535. For more information about debug levels, see Chapter 4, “Debugging levels ,” on page 131.

-i ***ip_address***
Specifies the IP address to which the Directory Server instance binds. If more than one IP address is specified, the comma separator must be used with no spaces. Spaces are allowed only if the entire argument is surrounded in quotation marks. To use all available IP addresses, use the key word, `all`. All available IP addresses is the default setting, if you do not specify the **-i** parameter.

-I ***instance_name***
Specifies the Directory Server instance name to update.

-n
Specifies to run in no prompt mode. All output from the command is generated, except for messages that require user interaction.

-q
Specifies to run in quiet mode. All output from the command is suppressed, except for error messages. If you also specify the **-d** parameter, then the trace output is not suppressed.

-v
Specifies to show the version information of the command.

-?
Specifies to show the syntax format.

Examples

Example 1:

To update the IP addresses of the `myinst` directory server instance to bind on `1.3.45.668`, run the following command:

```
idssethost -I myinst –i 1.3.45.668
```

Example 2:

To update the IP addresses of the `myinst` directory server instance to bind to all available IP addresses, run the following command:

```
idssethost -I myinst –i all
```

Note: You can change the host name by using the **idsldapmodify** command or **Web Administration Tool**. The modify command might fail, if the IP address specified is not valid. To ensure that there are no conflicts with other ports on particular IP addresses, the IP address updates are done by the root on the system.

idssetport

Use the **idssetport** command to set the ports to which a Directory Server instance binds.

Description

This command sets the specified ports so that a particular Directory Server can bind to it. The administrator specifies a Directory Server instance name and the ports to update. You must stop the Directory Server instance for which you are updating the ports. If the Administration Server of the instance is running and the Administration Server port is changed, then you must restart the Administration Server.

Synopsis

```
idssetport [-I instancename
           [-p port] [-s secureport] [-a admport] [-c admsecureport]
           [-d debuglevel] [-b outputfile] [-q] [-n]] | -v | -?
```

Options

The **idssetport** command takes the following parameters:

-a ***adminport***

Specifies the port that the Administration Server of an instance listens on. Specify a positive number that is greater than 0 and less than 65535. The port that is specified must not cause a conflict with ports in use by other applications or operating systems. The ports must not be in use by other Directory Server instance that is bound to a host name or IP address.

-b ***outputfile***

Specifies the full path of a file to redirect console output. If you use this parameter with the **-q** parameter, errors are sent to the *outputfile* file. If debug mode is set, then the debug output is sent to this file.

-c ***adminsecureport***

Specifies the secure port that the Administration Server of an instance listens on. Specify a positive number that is greater than 0 and less than 65535. The port that is specified must not cause a conflict with ports in use by other applications or operating systems. The ports must not be in use by other Directory Server instance that is bound to a host name or IP address

-d ***debuglevel***

Sets the LDAP debug level to *debuglevel*. If you specify this parameter, the command sends the debug output to stdout. The *debuglevel* value is a bit mask that controls which output is generated with values from 1 to 65535. For more information about debug levels, see Chapter 4, “Debugging levels ,” on page 131.

-I ***instancename***
Specifies the Directory Server instance name to update.

-n
Specifies to run in no prompt mode. All output from the command is generated, except for messages that require user interaction.

-p ***port***
Specifies the port that the Directory Server instance listens on. Specify a positive number that is greater than 0 and less than 65535. The port that is specified must not cause a conflict with ports in use by other applications or operating systems. The ports must not be in use by other Directory Server instance that is bound to a host name or IP address

-q
Specifies to run in quiet mode. All output from the command is suppressed, except for error messages. If you also specify the **-d** parameter, then the trace output is not suppressed.

-s ***secureport***
Specifies the secure port that the Directory Server instance listens on. Specify a positive number that is greater than 0 and less than 65535. The port that is specified must not cause a conflict with ports in use by other applications or operating systems. The ports must not be in use by other Directory Server instance that is bound to a host name or IP address

-v
Specifies to show the version information of the command.

-?
Specifies to show the syntax format.

Examples

Example 1:
To update the port of the `myinst` Directory Server instance to 555, run the following command:

```
idssetport -I myinst –p 555
```

Note:

1. By default, all the ports in the range of 1 - 1024, including ports 389 and 636. These ports can be used only by the root on AIX®, Linux, and Solaris, systems.
2. You can change the host name by using the **idsldapmodify** command or **Web Administration Tool**. The modify command might fail if the IP address specified is not valid on the system. To ensure that there are no conflicts with other ports on particular IP addresses, the IP address updates must be done by the root administrator.

idssnmp

Use the **idssnmp** command to start or stop the Directory Server SNMP agent tool.

Description

The **idssnmp** command starts or stops the `idssnmp` process.

Synopsis

The syntax for the **idssnmp** command:

```
idssnmp [-k] [-n]
```

Options

The **idssnmp** command takes the following parameters.

-k
Stops the Directory Server SNMP agent.

-n
Starts the Directory Server SNMP agent.

Examples

Example 1:
To start the Directory Server SNMP agent, run the following command:

```
idssnmp -n
```

Example 2:
To stop the Directory Server SNMP agent, run the following command:

```
idssnmp -k
```

idsucfgchglg

Use the **idsucfgchglg** command to unconfigure a change log for a Directory Server instance.

Description

The **idsucfgchglg** command unconfigures a change log for a Directory Server instance. To unconfigure, the change log must be configured in the ibmslapd.conf file. The command prompts you to confirm the action before the change log is removed.

Synopsis

```
idsucfgchglg [-I instancename [-f configfile] [-d debuglevel]
[-b outputfile] [-q] [-n]] | -v | -? [-Y]
```

Options

The **idsucfgchglg** command takes the following parameters.

-b ***outputfile***
Specifies the file name to redirect console output. If you use this parameter with the **-q** parameter, errors are sent to the *outputfile* file. If debug mode is set, then the debug output is sent to this file.

Note: The output file is created in the CustomOut folder.

-d ***debuglevel***
Sets the LDAP debug level to *debuglevel*. If you specify this parameter, the command sends the debug output to stdout. The *debuglevel* value is a bit mask that controls which output is generated with values from 1 to 65535. For information about debug levels, see Chapter 4, “Debugging levels ,” on page 131.

-f ***configfile***
Specifies the customized configuration file. If this parameter is not specified, the default configuration file for the Directory Server instance is used.

Note: The customized configuration file must be uploaded to CustomIn folder before you can use this parameter.

-I ***instancename***
Specifies the Directory Server instance name to update.

-n
Specifies to run in no prompt mode. All output from the command is generated, except for messages that require user interaction.

-q
Specifies to run in quiet mode. All output from the command is suppressed, except for error messages. If you also specify the **-d** parameter, then the trace output is not suppressed.

-v
Specifies to show the version information of the command.

-?
Specifies to show the syntax format.

-Y
Specify Remote Database as the target.

Examples

Example 1:
To unconfigure the change log for a Directory Server instance without prompting for confirmation, run the following command:

```
idsucfgchglg –n
```

Example 2:
To unconfigure the change log for the `myinst` instance on a system with multiple instances, run the following command:

```
idsucfgchglg –I myinst
```

idsucfgdb

Use the **idsucfgdb** command to unconfigure a database for a Directory Server instance.

Description

The **idsucfgdb** command unconfigures the database for a Directory Server instance. By default, the command unconfigures the database only from the `ibmslapd.conf` file and does not delete the database. To delete the database during the unconfiguration process, the **-r** parameter must be specified. The command prompts you to confirm if you want to continue with the requested actions.

Synopsis

```
idsucfgdb [-I instancename [-r] [-f configfile] [-d debuglevel] [-b outputfile]
[-q] [-s] [-n]] | -v | -? [-Y]
```

Options

The **idsucfgdb** command takes the following parameters.

-b ***outputfile***
Specifies the file name to redirect console output. If you use this parameter with the **-q** parameter, errors are sent to the *outputfile* file. If debug mode is set, then the debug output is sent to this file.

Note: The output file is created in the `CustomOut` folder.

-d ***debuglevel***
Sets the LDAP debug level to *debuglevel*. If you specify this parameter, the command sends the debug output to stdout. The *debuglevel* value is a bit mask that controls which output is generated with values from 1 to 65535. For more information about debug levels, see Chapter 4, “Debugging levels ,” on page 131.

-f ***configfile***

Specifies the customized configuration file. If this parameter is not specified, the default configuration file for the Directory Server instance is used.

Note: The customized configuration file must be uploaded to `CustomIn` folder before you can use this parameter.

I ***instancename***

Specifies the Directory Server instance name to update.

-n

Specifies to run in no prompt mode. All output from the command is generated, except for messages that require user interaction.

-q

Specifies to run in quiet mode. All output from the command is suppressed, except for error messages. If you also specify the **-d** parameter, then the trace output is not suppressed.

-r

Specifies to remove any database that is configured with the directory server instance.

-s

Removes the backup copy of the database, if configured.

-v

Specifies to show the version information of the command.

-?

Specifies to show the syntax format.

-Y

Specify the Remote Database as the target.

Examples

Example 1:

To unconfigure the database for a Directory Server instance and to not prompt the user, run the following command:

```
idsucfgdb -n
```

Example 2:

To unconfigure and delete the database for an instance and to not prompt the user for the confirmation, run the following command:

```
idsucfgdb –r –n
```

Example 3:

To unconfigure a database and to remove the backup, run the following command:

```
idsucfgdb -I instance_name -r -s
```

idsucfgsch

Use the **idsucfgsch** command to unconfigure a schema file for a Directory Server instance.

Description

The **idsucfgsch** command unconfigures a schema file for a Directory Server instance. The schema file must be configured in the `ibmslapd.conf` file of the Directory Server instance. The Directory Server instance owner must specify the schema file to remove the file from the `ibmslapd.conf` file of the Directory Server instance.

Synopsis

```
idsucfgsch [-I instancename -s schemafile [-f configfile] [-d debuglevel]
[-b outputfile] [-q] [-n]] | -v | -?
```

Options

The **`idsucfgsch`** command takes the following parameters.

`-b` ***outputfile***
Specifies the full path of a file to redirect console output. If you use this parameter with the **`-q`** parameter, errors are sent to the *outputfile* file. If debug mode is set, then the debug output is sent to this file.

`-d` ***debuglevel***
Sets the LDAP debug level to *debuglevel*. If you specify this parameter, the command sends the debug output to stdout. The *debuglevel* value is a bit mask that controls which output is generated with values from 1 to 65535. For more information about debug levels, see Chapter 4, "Debugging levels ," on page 131.

`-f` ***configfile***
Specifies the full path to the configuration file to update. If this parameter is not specified, the default configuration file for the Directory Server instance is used.

`-I` ***instancename***
Specifies the Directory Server instance name to update.

`-n`
Specifies to run in no prompt mode. All output from the command is generated, except for messages that require user interaction.

`-q`
Specifies to run in quiet mode. All output from the command is suppressed, except for error messages. If you also specify the **`-d`** parameter, then the trace output is not suppressed.

`-s` ***schemafile***
Specifies the schema file to remove from the Directory Server instance.

`-v`
Specifies to show the version information of the command.

`-?`
Specifies to show the syntax format.

Examples

Example 1:
To unconfigure the `/userdata/directory/CustomIn/myschema.oc` schema file from the `ibmslapd.conf` file of an instance, run the following command:

```
idsucfgsch –s /userdata/directory/CustomIn/myschema.oc
```

Note: The following system-defined schema files cannot be removed.

- `V3.system.at`
- `V3.system.oc`
- `V3.config.at`
- `V3.config.oc`
- `V3.ibm.at`
- `V3.ibm.oc`
- `V3.user.at`
- `V3.user.oc`
- `V3.ldapsyntaxes`

- V3.matchingrules

idsucfgsuf

Use the **idsucfgsuf** command to remove a suffix from a Directory Server instance.

Description

The **idsucfgsuf** command removes a suffix from a Directory Server instance. The suffix is removed from the ibmslapd.conf file of the directory server instance. This command fails if the suffix does not exist in the configuration file.

Synopsis

```
idsucfgsuf [-I instancename -s suffix [-f configfile] [-d debuglevel]
          [-b outputfile] [-q] [-n]] | -v | -?
```

Options

The **idsucfgsuf** command takes the following parameters.

-b ***outputfile***
Specifies the full path of a file to redirect console output. If you use this parameter with the **-q** parameter, errors are sent to the *outputfile* file. If debug mode is set, then the debug output is sent to this file.

-d ***debuglevel***
Sets the LDAP debug level to *debuglevel*. If you specify this parameter, the command sends the debug output to stdout. The *debuglevel* value is a bit mask that controls which output is generated with values from 1 to 65535. For more information about debug levels, see Chapter 4, “Debugging levels ,” on page 131.

-f ***configfile***
Specifies the full path to the configuration file to update. If this parameter is not specified, the default configuration file for the Directory Server instance is used.

-I ***instancename***
Specifies the Directory Server instance name. This parameter is required if there are more Directory Server instances on the system.

-n
Specifies to run in no prompt mode. All output from the command is generated, except for messages that require user interaction.

-q
Specifies to run in quiet mode. All output from the command is suppressed, except for error messages. If you also specify the **-d** parameter, then the trace output is not suppressed.

-s ***suffix***
Specifies to remove the suffix from the Directory Server instance.

-v
Specifies to show the version information of the command.

-?
Specifies to show the syntax format.

Examples

Example 1:
To remove the o=sample suffix from the ibmslapd.conf file with a single Directory Server instance on a system, run the following command:

```
idscfgsuf -s o=sample
```

Example 2:

To remove the o=sample suffix from the ibmslapd.conf file of a Directory Server instance, run the following command:

```
idscfgsuf -I instancename -s o=sample
```

You must provide the instance name if there are multiple Directory Server instances on the system.

Note: The following system defined suffixes cannot be removed.

- cn=pwdpolicy
- cn=localhost
- cn=configuration
- cn=ibmpolicies

idsunlockwat

Use the **idsunlockwat** command to unlock the Directory Server Web Administration Tool.

Description

When the maximum number of failed login attempts are exceeded, the Directory Server Web Administration Tool console gets locked. The **idsunlockwat** command unlocks the Web Administration Tool by deleting the ids_acc.lck file.

Synopsis

```
idsunlockwat
```

Options

The **idsunlockwat** command does not have any parameters.

ldtrc

Use the **ldtrc** command to run various trace options on a system.

Description

You can run the tracing utility, **ldtrc**, to activate or deactivate tracing of a Directory Server. You can use the trace options that are provided with the command to troubleshoot the instance-specific issues. To see syntax help for **ldtrc**, run the ldtrc -? command.

Note: For format and flow options, you must set the *TRCTFIDIR* environment variable to the directory that contains the **Trace Facility Information** files(*.tfi).

Synopsis

```
ldtrc (chg|clr|dmp|flw|fmt|inf|off|on) options
```

Options

The **ldtrc** command takes the following parameters.

chg | change

The trace must be active before you can use the **chg** parameter to change the values for the following options.

- [-m <mask>]: where, <mask> = <products>.<events>.<components>.<classes>.<functions>

- [-p <pid>[.<tid>]]: Traces only the specified process or thread.
- [-c <cpid>]: Traces only the specified companion process.
- [-e <maxSeverErrors>]: Stops tracing after the maximum number of server errors (maxSevereErrors) is reached.
- [-this <thisPointer>]: Traces only the specified object.

clr | clear

Clears the existing trace buffer.

dmp | dump

Dumps the trace information to a file. This information includes process flow data and server debug messages. The default location for the file is CustomOut.

Note: This file contains binary ldtrc data that must be formatted with the **ldtrc format** command.

flw | flow

- [-m <mask>]: where <mask> = <products>.<events>.<components>.<classes>.<functions>
- [-p <pid>[.<tid>]]: Shows control flow only for the specified process or thread.
- [-r]: Specifies to output trace in reverse chronological order.
- [-x <onlyRecord> | <firstRecord> - <lastRecord>]: Shows the control flow only for the specified record or shows the control flow between the specified first and last records.
- [-this <thisPointer>]: Traces only the specified object.
- [<sourceFile> [<destFile>]: Specifies the trace file to format and the destination file for the formatted output.

fmt | format

- [-m <mask>] where <mask> = <products>.<events>.<components>.<classes>.<functions>
- [-p <pid>[.<tid>]]: Specifies to format trace records that belong to a process or thread.
- [-j]: Specifies to join the first two lines of the trace output.
- [-r]: Specifies to output trace in reverse chronological order.
- [-x <onlyRecord> | <firstRecord> - <lastRecord>]: Shows the control flow only for the specified record or shows the control flow between the specified first and last records.
- [-this <thisPointer>]: Traces only the specified object.
- [<sourceFile> [<destFile>]: Specifies the trace file to format and the destination file for the formatted output.

inf | info | information

[<sourceFile> [<destFile>]: Gets the information about the trace. You must specify the source file that can be a binary trace file or trace buffer (if file is "-") and a destination file. The following example shows information that the **info** parameter generated.

```
sdsva.example.com> sds server_tools ldtrc info
Trace Version           :     1.00
Op. System              :   Linux2
Op. Sys. Version        :      2.2
H/W Platform            :      X86

Mask                    : *.*.*.*.*.*
pid.tid to trace        : all
cpid    to trace        : all
this pointer to trace   : all
Treat this rc as sys err: none
Max severe errors       : 1
Max record size         : 32768 bytes
Trace destination       : shared memory
Records to keep         : last
```

```
Trace buffer size       : 1048576 bytes
Trace data pointer check: no
```

on

Activates the tracing facility. You can specify any of the following options.

- `[-m <mask>]` where, `<mask> = <products>.<events>.<components>.<classes>.<functions>`
- `[-p <pid>[.<tid>]]`: Traces only the specified process or thread.
- `[-c <cpid>]`: Traces only the specified companion process.
- `[-e <maxSeverErrors>]`: Stops tracing after the maximum number of server errors (`maxSevereErrors`) is reached.
- `[-s | -f <fileName>]`: Sends the output to shared memory or a file.
- `[-l [<bufferSize>] | -i [<bufferSize>]]`: Specifies to retain the last or the initial records. The default buffer is 1M.
- `[-this <thisPointer>]`: Traces only the specified object.
- `[-perf]`: Traces only performance records.

Note: The tracing utility must be on for server data to be traced.

off

Turns off the tracing facility.

Examples

Example 1:

To turn on the trace facility, run the following command:

```
ldtrc on
```

Example 2:

To turn off the trace facility, run the following command:

```
ldtrc off
```

scimautostart

Use the **scimautostart** command to enable the auto start service for Directory Integrator SCIM Service

Description

Enable or disable autostart of the Directory Integrator SCIM Service during boot up with the **scimautostart** server tool.

```
sdsva.example.com: server_tools> scimautostart

USAGE: scimautostart -e | -d
```

Synopsis

```
scimautostart -e | -d
```

Options

The **scimautostart** command takes the following parameters:

-e

Enables automatic startup of the Directory Integrator SCIM Service during boot up.

-d

Disables automatic startup of Directory Integrator SCIM Service during boot up.

Examples

Example 1:

To enable auto startup for the Directory Integrator SCIM Service, run the following command:

```
scimautostart -e
```

Example 2:

To disable auto startup for the Directory Integrator SCIM Service, run the following command:

```
scimautostart -d
```

setservertype

Use the **setservertype** command to set the Directory Server instance type.

Description

The **setservertype** command enables you to set the server type for the Directory Server instance.

Synopsis

```
setservertype server_type
```

Options

The valid server types are:

- RDBM specifies to load an RDBM server as the backend for the Directory Server instance.
- PROXY specifies to load the Proxy Server as the backend for the Directory Server instance. For more information, see The Proxy Server. This server type option is available only in IBM Security Directory Suite, Standard and Enterprise Editions.
- VD specifies to load the Virtual Directory as the backend for the Directory Server instance. For more information, see Virtual Directory administration.

  Note: This server type option is available only in IBM Security Directory Suite, Enterprise Edition.

unlockadmin

Use the **unlockadmin** command to unlock the IBM Security Directory Suite virtual appliance admin user.

Description

The IBM Security Directory Suite virtual appliance admin user is locked out after 10 failed attempts to log in to the virtual appliance console (local management interface or LMI). The **unlockadmin** command enables you to unlock the admin user so that the user can log in to the virtual appliance console.

Note: If the admin user is locked out, the user can log in after 60 minutes and the

Synopsis

```
unlockadmin
```

Options

The **`unlockadmin`** command does not have any parameters.

Chapter 2. Directory Server client utilities

The Directory Server client utilities use the `ldap_sasl_bind` or `ldap_sasl_bind_s` API to initiate a bind. The behavior and usage of the client utilities varies based on the values that you provide.

When a bind is initiated, several results can be returned. When you use various combinations of user IDs and passwords, the following bind results are observed:

- If you specify the admin DN, the password must be correctly specified or the bind is not successful.
- If a null DN or a 0 length DN is specified, you receive unauthenticated access unless you are using an external bind (SASL) such as Kerberos.
- If a DN is specified, and is non-null, a password must also be specified, or an error is returned.
- If a DN and password are specified but do not fall under any suffix in the directory, a referral is returned.
- If a DN and password are specified and are correct, the user is bound with that identity.
- If a DN and password are specified but the DN does not exist, unauthenticated access is given.
- If a DN and password are specified and the DN exists but the object does not have user password, an error message is returned.

Note:

You can change the source code for some of these LDAP client utilities and build your own version of these LDAP client utilities. You can change the following client utilities:

- **`idsldapchangepwd`**
- **`idsldapdelete`**
- **`idsldapexop`**
- **`idsldapmodify`**, **`idsldapadd`**
- **`idsldapmodrdn`**
- **`idsldapsearch`**

However, any altered versions of these LDAP utilities are not supported.

You can download the example code from the virtual appliance console, **Custom File Management** > **ClientSDK** folder. See Managing custom files.

LDAP C-client utilities (**`ibmdirctl`**, **`ldapadd`**, **`ldapchangepwd`**, **`ldapcompare`**, **`ldapdelete`**, **`ldapexop`**, **`ldapmodify`**, **`ldapmodrdn`**, **`ldapsearch`**, and **`ldaptrace`**) internally use the `connect()` system call to connect to the specified socket on the target system. When an LDAP client attempts to connect to a system that is down, then the `connect()` system call exits only when the TCP/IP timeout is met. In such case, it gives an impression that the LDAP client operation is in hung state. You can configure an LDAP client to return earlier than the system-wide TCP/IP timeout value. To return earlier, run an LDAP client command with the **`-1`** option along with the timeout value in seconds and microseconds.

Note: If the value provided is greater than the system-wide TCP/IP timeout, then the system-wide TCP/IP timeout occurs first and then the application exits. The **`-1`** option does not override the system-wide TCP/IP timeout value but provides a mechanism for LDAP C-client utilities to timeout early.

Chapter 3. SSL and TLS notes

Determine the use of SSL and TLS functions with command-line utilities. You must install the SSL and TLS libraries and tools to use the SSL or TLS-related functions that are associated with this command.

The SSL or TLS libraries and tools are provided with GSKit, which includes security software developed by RSA Security Inc.

For information about the use of 128-bit and triple DES encryption by LDAP applications, see the information about LDAP_SSL in the *Programming Reference* section of the IBM Security Directory Suite documentation. It describes the steps that are required to build the sample programs and your applications so they can use SSL with the strongest encryption algorithms available. For more information about linking an LDAP application so that it can access 128-bit and triple DES encryption algorithms, see the `makefile` associated with the sample programs.

The **`ikeyman`** tool manages the content of a client key database file. You can use the **`ikeyman`** tool to define the set of trusted certificate authorities (CAs) that are to be trusted by the client. By obtaining certificates from trusted CAs, storing them in the key database file, and marking them as `trusted`, you can establish a trust relationship with LDAP servers that use trusted certificates that are issued by one of the trusted CAs. You can also use the **`ikeyman`** tool to obtain a client certificate so that client and server authentication can be run.

If the clients use server authentication to access LDAP servers, it is sufficient to define one or more trusted root certificates in the key database file. With server authentication, the client can be assured that the target LDAP server uses a certificate by one of the trusted CAs. All LDAP transactions that use the SSL or TLS connection with the server are encrypted including the LDAP credentials that are supplied on the `ldap_bind` or `ldap_simple_bind_s`. For example, if the LDAP server is using a high-assurance VeriSign certificate, you must obtain a CA certificate from VeriSign. You must then import the certificate into your key database file, and mark it as trusted. If the LDAP server is using a self-signed server certificate, the administrator of the server can supply you a copy of the server certificate request file. Import the certificate request file into your key database file and mark it as trusted.

If the LDAP servers accessed by a client use client and server authentication, it is necessary to do the following steps.

- Define one or more trusted root certificates in the key database file. It assures the client that the target LDAP server uses a certificate by one of the trusted CAs. All LDAP transactions that flow over the SSL or TLS connection with the server are encrypted, including the LDAP credentials that are supplied on the `ldap_bind` or `ldap_simple_bind_s`.
- Create a key pair by using the **`ikeyman`** tool and request a client certificate from a CA. After you receive the signed certificate from the CA, store the certificate in the client key database file.

Chapter 4. Debugging levels

Use the debugging levels to identify an appropriate debug level to obtain debug trace for a Directory Server instance.

The **ldtrc** utility must be running to obtain the debug trace when you run the server utilities in debug mode. The **ldtrc** utility is not required for the client utilities. For example, to run the **idscfgdb** command in debug mode for a Directory Server instance, myinst, issue the following commands.

```
ldtrc on
idscfgdb -I myinst -d debuglevel
```

The specified debug level value determines which categories of debug output to generate.

Table 1. Debug categories

Hex	Decimal	Value	Description
0x0001	1	LDAP_DEBUG_TRACE	Entry and exit from routines
0x0002	2	LDAP_DEBUG_PACKETS	Packet activity
0x0004	4	LDAP_DEBUG_ARGS	Data arguments from requests
0x0008	8	LDAP_DEBUG_CONNS	Connection activity
0x0010	16	LDAP_DEBUG_BER	Encoding and decoding of data
0x0020	32	LDAP_DEBUG_FILTER	Search filters
0x0040	64	LDAP_DEBUG_MESSAGE	Messaging subsystem activities and events
0x0080	128	LDAP_DEBUG_ACL	Access Control List activities
0x0100	256	LDAP_DEBUG_STATS	Operational statistics
0x0200	512	LDAP_DEBUG_THREAD	Threading statistics
0x0400	1024	LDAP_DEBUG_REPL	Replication statistics
0x0800	2048	LDAP_DEBUG_PARSE	Parsing activities
0x1000	4096	LDAP_DEBUG_PERFORMANCE	Relational backend performance statistics
0x2000	8192	LDAP_DEBUG_RDBM	Relational backend activities
0x4000	16384	LDAP_DEBUG_REFERRAL	Referral activities
0x8000	32768	LDAP_DEBUG_ERROR	Error conditions
0xffff	65535	LDAP_DEBUG_ANY	All levels of debug

For example, when you specify a bit mask value of 65535, the command turns on full debug output and generates the most complete information.

Contact IBM Service for assistance with interpreting of the debug output and resolving of the problem.

When you are finished with debugging, issue the following command to deactivate the **ldtrc** utility.

```
ldtrc off
```

Chapter 5. ASCII characters from 33 to 126

Use the ASCII characters table to determine the characters to use for Directory Server instance encryption seed and encryption salt.

You can use the ASCII characters from 33 to 126 in the encryption seed string and encryption salt.

Table 2. ASCII characters from 33 to 126

ASCII code	Character	ASCII code	Character	ASCII code	Character
33	! exclamation point	34	" double quotation	35	# number sign
36	$ dollar sign	37	% percent sign	38	& ampersand
39	' apostrophe	40	(left parenthesis	41	) right parenthesis
42	* asterisk	43	+ plus sign	44	, comma
45	- hyphen	46	. period	47	/ slash
48	0	49	1	50	2
51	3	52	4	53	5
54	6	55	7	56	8
57	9	58	: colon	59	; semicolon
60	< less-than sign	61	= equals sign	62	> greater-than sign
63	? question mark	64	@ at sign	65	A uppercase a
66	B uppercase b	67	C uppercase c	68	D uppercase d
69	E uppercase e	70	F uppercase f	71	G uppercase g
72	H uppercase h	73	I uppercase i	74	J uppercase j
75	K uppercase k	76	L uppercase l	77	M uppercase m
78	N uppercase n	79	O uppercase o	80	P uppercase p
81	Q uppercase q	82	R uppercase r	83	S uppercase s
84	T uppercase t	85	U uppercase u	86	V uppercase v
87	W uppercase w	88	X uppercase x	89	Y uppercase y
90	Z uppercase z	91	[left square bracket	92	\ backslash
93	] right square bracket	94	^ caret	95	_ underscore
96	` grave accent	97	a lowercase a	98	b lowercase b
99	c lowercase c	100	d lowercase d	101	e lowercase e
102	f lowercase f	103	g lowercase g	104	h lowercase h
105	i lowercase i	106	j lowercase j	107	k lowercase k
108	l lowercase l	109	m lowercase m	110	n lowercase n
111	o lowercase o	112	p lowercase p	113	q lowercase q
114	r lowercase r	115	s lowercase s	116	t lowercase t

Table 2. ASCII characters from 33 to 126 (continued)

ASCII code	**Character**	**ASCII code**	**Character**	**ASCII code**	**Character**
117	u lowercase u	118	v lowercase v	119	w lowercase w
120	x lowercase x	121	y lowercase y	122	z lowercase z
123	{ left curly brace	124	| vertical bar	125	} right curly brace
126	~ tilde				

Chapter 6. Supported IANA character sets

Use the Internet Assigned Numbers Authority (IANA) character sets to identify the text string that can be assigned to the `charset` tag.

The following table defines the IANA defined character sets that can be defined for the `charset` tag in a Version 1 LDIF file, on a per-platform basis. The value in the left-most column defines the text string that can be assigned to the `charset` tag. An X indicates that conversion from the specified `charset` to UTF-8 is supported for the associated operating systems. And, all string content in the LDIF file is assumed to be represented in the specified `charset`. The `n/a` symbol indicates that the conversion is not supported for the associated operating systems.

String content is defined to be all attribute values that follow an attribute name and a single colon.

For more information about IANA registered character sets, see http://www.iana.org.

Table 3. IANA defined character sets

Character	**Locale**				**DB2 code page**	
Set Name	**Linux, Linux_390**	**NT**	**AIX**	**Solaris**	**UNIX**	**NT**
`ISO-8859-1`	X	X	X	X	819	1252
`ISO-8859-2`	X	X	X	X	912	1250
`ISO-8859-5`	X	X	X	X	915	1251
`ISO-8859-6`	X	X	X	X	1089	1256
`ISO-8859-7`	X	X	X	X	813	1253
`ISO-8859-8`	X	X	X	X	916	1255
`ISO-8859-9`	X	X	X	X	920	1254
`ISO-8859–15`	n/a	X	X	X		
`IBM437`	n/a	X	n/a	n/a	437	437
`IBM850`	n/a	X	X	n/a	850	850
`IBM852`	n/a	X	n/a	n/a	852	852
`IBM857`	n/a	X	n/a	n/a	857	857
`IBM862`	n/a	X	n/a	n/a	862	862
`IBM864`	n/a	X	n/a	n/a	864	864
`IBM866`	n/a	X	n/a	n/a	866	866
`IBM869`	n/a	X	n/a	n/a	869	869
`IBM1250`	n/a	X	n/a	n/a		
`IBM1251`	n/a	X	n/a	n/a		
`IBM1253`	n/a	X	n/a	n/a		
`IBM1254`	n/a	X	n/a	n/a		
`IBM1255`	n/a	X	n/a	n/a		
`IBM1256`	n/a	X	n/a	n/a		

Table 3. IANA defined character sets (continued)

Character	Locale				DB2 code page	
Set Name	Linux, Linux_390	NT	AIX	Solaris	UNIX	NT
TIS-620	n/a	X	X	n/a	874	874
EUC-JP	X	n/a	X	X	954	n/a
EUC-KR	n/a	n/a	X	X*	970	n/a
EUC-CN	n/a	n/a	X	X	1383	n/a
EUC-TW	n/a	n/a	X	X	964	n/a
Shift-JIS	X	X	X	X	932	943
KSC	n/a	X	n/a	n/a	n/a	949
GBK	n/a	X	X	n/a	1386	1386
Big5	n/a	X	X	X	950	950
GB18030	X	X	X	X		
HP15CN						

Appendix A. ASCII characters from 33 to 126

Use the ASCII characters table to determine the characters to use for Directory Server instance encryption seed and encryption salt.

You can use the ASCII characters from 33 to 126 in the encryption seed string and encryption salt.

Table 4. ASCII characters from 33 to 126

ASCII code	Character	ASCII code	Character	ASCII code	Character
33	! exclamation point	34	" double quotation	35	# number sign
36	$ dollar sign	37	% percent sign	38	& ampersand
39	' apostrophe	40	(left parenthesis	41	) right parenthesis
42	* asterisk	43	+ plus sign	44	, comma
45	- hyphen	46	. period	47	/ slash
48	0	49	1	50	2
51	3	52	4	53	5
54	6	55	7	56	8
57	9	58	: colon	59	; semicolon
60	< less-than sign	61	= equals sign	62	> greater-than sign
63	? question mark	64	@ at sign	65	A uppercase a
66	B uppercase b	67	C uppercase c	68	D uppercase d
69	E uppercase e	70	F uppercase f	71	G uppercase g
72	H uppercase h	73	I uppercase i	74	J uppercase j
75	K uppercase k	76	L uppercase l	77	M uppercase m
78	N uppercase n	79	O uppercase o	80	P uppercase p
81	Q uppercase q	82	R uppercase r	83	S uppercase s
84	T uppercase t	85	U uppercase u	86	V uppercase v
87	W uppercase w	88	X uppercase x	89	Y uppercase y
90	Z uppercase z	91	[left square bracket	92	\ backslash
93	] right square bracket	94	^ caret	95	_ underscore
96	` grave accent	97	a lowercase a	98	b lowercase b
99	c lowercase c	100	d lowercase d	101	e lowercase e
102	f lowercase f	103	g lowercase g	104	h lowercase h
105	i lowercase i	106	j lowercase j	107	k lowercase k
108	l lowercase l	109	m lowercase m	110	n lowercase n
111	o lowercase o	112	p lowercase p	113	q lowercase q
114	r lowercase r	115	s lowercase s	116	t lowercase t

Table 4. ASCII characters from 33 to 126 (continued)

ASCII code	Character	ASCII code	Character	ASCII code	Character
117	u lowercase u	118	v lowercase v	119	w lowercase w
120	x lowercase x	121	y lowercase y	122	z lowercase z
123	{ left curly brace	124	| vertical bar	125	} right curly brace
126	~ tilde				

Appendix B. Supported IANA character sets

Use the Internet Assigned Numbers Authority (IANA) character sets to identify the text string that can be assigned to the `charset` tag.

The following table defines the IANA defined character sets that can be defined for the `charset` tag in a Version 1 LDIF file, on a per-platform basis. The value in the left-most column defines the text string that can be assigned to the `charset` tag. An X indicates that conversion from the specified `charset` to UTF-8 is supported for the associated operating systems. And, all string content in the LDIF file is assumed to be represented in the specified `charset`. The `n/a` symbol indicates that the conversion is not supported for the associated operating systems.

String content is defined to be all attribute values that follow an attribute name and a single colon.

For more information about IANA registered character sets, see http://www.iana.org.

Table 5. IANA defined character sets

Character	**Locale**				**DB2 code page**	
Set Name	**Linux, Linux_390**	**NT**	**AIX**	**Solaris**	**UNIX**	**NT**
`ISO-8859-1`	X	X	X	X	819	1252
`ISO-8859-2`	X	X	X	X	912	1250
`ISO-8859-5`	X	X	X	X	915	1251
`ISO-8859-6`	X	X	X	X	1089	1256
`ISO-8859-7`	X	X	X	X	813	1253
`ISO-8859-8`	X	X	X	X	916	1255
`ISO-8859-9`	X	X	X	X	920	1254
`ISO-8859–15`	n/a	X	X	X		
`IBM437`	n/a	X	n/a	n/a	437	437
`IBM850`	n/a	X	X	n/a	850	850
`IBM852`	n/a	X	n/a	n/a	852	852
`IBM857`	n/a	X	n/a	n/a	857	857
`IBM862`	n/a	X	n/a	n/a	862	862
`IBM864`	n/a	X	n/a	n/a	864	864
`IBM866`	n/a	X	n/a	n/a	866	866
`IBM869`	n/a	X	n/a	n/a	869	869
`IBM1250`	n/a	X	n/a	n/a		
`IBM1251`	n/a	X	n/a	n/a		
`IBM1253`	n/a	X	n/a	n/a		
`IBM1254`	n/a	X	n/a	n/a		
`IBM1255`	n/a	X	n/a	n/a		
`IBM1256`	n/a	X	n/a	n/a		

Table 5. IANA defined character sets (continued)

Character	Locale				DB2 code page	
Set Name	**Linux, Linux_390**	**NT**	**AIX**	**Solaris**	**UNIX**	**NT**
TIS-620	n/a	X	X	n/a	874	874
EUC-JP	X	n/a	X	X	954	n/a
EUC-KR	n/a	n/a	X	X*	970	n/a
EUC-CN	n/a	n/a	X	X	1383	n/a
EUC-TW	n/a	n/a	X	X	964	n/a
Shift-JIS	X	X	X	X	932	943
KSC	n/a	X	n/a	n/a	n/a	949
GBK	n/a	X	X	n/a	1386	1386
Big5	n/a	X	X	X	950	950
GB18030	X	X	X	X		
HP15CN						

Index

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data discussed herein is presented as derived under specific operating conditions. Actual results may vary.

The client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBMproducts. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:
© (your company name) (year).
Portions of this code are derived from IBM Corp. Sample Programs.
© Copyright IBM Corp. _enter the year or years_.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.