IBM Security Directory Server
Version 6.3.1.5

Command Reference

SC27-2753-02

IBM Security Directory Server
Version 6.3.1.5

Command Reference

SC27-2753-02

Note

Before using this information and the product it supports, read the general information under "Notices" on page 165.

Edition notice

Note: This edition applies to version 6.3.1.5 of *IBM Security Directory Server* (product number 5724-J39) and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

About this publication

IBM® Security Directory Server, previously known as IBM Tivoli® Directory Server, is an IBM implementation of Lightweight Directory Access Protocol for the following operating systems:

- Microsoft Windows
- AIX®
- Linux (System x®, System z®, System p®, and System i®)
- Solaris
- Hewlett-Packard UNIX (HP-UX) (Itanium)

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

Access to publications and terminology

This section provides:

- A list of publications in the "IBM Security Directory Server library."
- Links to "Online publications" on page vi.
- A link to the "IBM Terminology website" on page vi.

IBM Security Directory Server library

The following documents are available in the IBM Security Directory Server library:

- *IBM Security Directory Server, Version 6.3.1.5 Product Overview*, GC27-6212-01

  Provides information about the IBM Security Directory Server product, new features in the current release, and system requirements information.
- *IBM Security Directory Server, Version 6.3.1.5 Quick Start Guide*, GI11-9351-02

  Provides help for getting started with IBM Security Directory Server. Includes a short product description and architecture diagram, and a pointer to the product documentation website and installation instructions.
- *IBM Security Directory Server, Version 6.3.1.5 Installation and Configuration Guide*, SC27-2747-02

  Contains complete information for installing, configuring, and uninstalling IBM Security Directory Server. Includes information about upgrading from a previous version of IBM Security Directory Server.
- *IBM Security Directory Server, Version 6.3.1.5 Administration Guide*, SC27-2749-02

  Contains instructions for administrative tasks through the Web Administration tool and the command line.
- *IBM Security Directory Server, Version 6.3.1.5 Reporting Guide*, SC27-6531-00

  Describes the tools and software for creating reports for IBM Security Directory Server.
- *IBM Security Directory Server, Version 6.3.1.5 Command Reference*, SC27-2753-02

  Describes the syntax and usage of the command-line utilities included with IBM Security Directory Server.

- *IBM Security Directory Server, Version 6.3.1.5 Server Plug-ins Reference* , SC27-2750-02

  Contains information about writing server plug-ins.
- *IBM Security Directory Server, Version 6.3.1.5 Programming Reference*, SC27-2754-02

  Contains information about writing Lightweight Directory Access Protocol (LDAP) client applications in C and Java™.
- *IBM Security Directory Server, Version 6.3.1.5 Performance Tuning and Capacity Planning Guide*, SC27-2748-02

  Contains information about tuning the directory server for better performance. Describes disk requirements and other hardware requirements for directories of different sizes and with various read and write rates. Describes known working scenarios for each of these levels of directory and the disk and memory used; also suggests rules of thumb.
- *IBM Security Directory Server, Version 6.3.1.5 Troubleshooting Guide*, GC27-2752-02

  Contains information about possible problems and corrective actions that can be taken before you contact IBM Software Support.
- *IBM Security Directory Server, Version 6.3.1.5 Error Message Reference*, GC27-2751-02

  Contains a list of all warning and error messages associated with IBM Security Directory Server.

Online publications

IBM posts product publications when the product is released and when the publications are updated at the following locations:

IBM Security Directory Server documentation website

The http://pic.dhe.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.IBMDS.doc/welcome.htm site displays the documentation welcome page for this product.

IBM Security Systems Documentation Central and Welcome page

IBM Security Systems Documentation Central provides an alphabetical list of all IBM Security Systems product documentation. You can also find links to the product documentation for specific versions of each product.

Welcome to IBM Security Systems documentation provides and introduction to, links to, and general information about IBM Security Systems documentation.

IBM Publications Center

The http://www-05.ibm.com/e-business/linkweb/publications/servlet/pbi.wss site offers customized search functions to help you find all the IBM publications you need.

IBM Terminology website

The IBM Terminology website consolidates terminology for product libraries in one location. You can access the Terminology website at http://www.ibm.com/software/globalization/terminology.

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 the Accessibility Appendix in the *IBM Security Directory Server Product Overview*.

Technical training

For technical training information, see the following IBM Education website at http://www.ibm.com/software/tivoli/education.

Support information

IBM Support assists with code-related problems and routine, short duration installation or usage questions. You can directly access the IBM Software Support site at http://www.ibm.com/software/support/probsub.html.

IBM Security Directory Server Troubleshooting Guide provides details about:

- What information to collect before you contact IBM Support.
- The various methods for contacting IBM Support.
- How to use IBM Support Assistant.
- Instructions and problem-determination resources to isolate and fix the problem yourself.

Note: The **Community and Support** tab on the product information center can provide additional support resources.

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. Command-line utilities

After you install IBM Security Directory Server, you must create and configure a directory server instance. IBM Security Directory Server provides both server and client command-line utilities that you can use to accomplish these LDAP-based requirements.

You require the following privileges to use command-line server utilities on various operating system on which you installed IBM Security Directory Server .

On AIX, Linux, Solaris systems

- You might require the root privileges.
- The user must be a member of the group that is associated with command.

On Windows systems

- You might require the administrator privileges.
- The user must be a member of the group that is associated with command.

For using the client command-line utilities, you must have the required permissions to access and use commands in the `bin` directory of IBM Security Directory Server installI home.

Server and client command-line utilities

Use the server and client command-line utilities available in IBM Security Directory Server to create and configure directory server instance.

You can run the following command-line utilities that are provided with IBM Security Directory Server to achieve various tasks.

Server utilities

- "**ddsetup**" on page 75
- "**idsadduser**" on page 79
- "**idsadscfg**" on page 80
- "**idsadsrun**" on page 82
- "**idsdbmaint**" on page 101
- "**idsdbmigr**" on page 103
- "**idsbulkload**, **bulkload**" on page 83
- "**idscfgchglg**" on page 91
- "**idscfgdb**" on page 93
- "**idscfgsch**" on page 97
- "**idscfgsuf**" on page 98
- "**idsdbback**, **dbback**" on page 99
- "**idsideploy**" on page 118
- "**idsdbrestore**, **dbrestore**" on page 104
- "**idsdb2ldif**, **db2ldif**" on page 106
- "**idsdiradm**, **ibmdiradm**" on page 109

- “**idsdnpw**” on page 110
- “**idsgendirksf**” on page 112
- “**idsicrt**” on page 113
- “**idsidrop**” on page 122
- “**idsilist**” on page 123
- “**idsimigr**” on page 126
- “**idsldif2db**, **ldif2db**” on page 129
- “**idslogmgmt**” on page 131
- “**idsperftune**” on page 132
- “**IDSProgRunner**” on page 136
- “**idsrunstats**, **runstats**” on page 136
- “**idssethost**” on page 137
- “**idssetport**” on page 139
- “**idsslapd**, **ibmslapd**” on page 141
- “**idssnmp**” on page 142
- “**idsucfgchglg**” on page 143
- “**idsucfgdb**” on page 144
- “**idsucfgsch**” on page 145
- “**idsucfgsuf**” on page 146
- “**ldtrc**” on page 148
- “**ldif**” on page 128
- “**idsrun**” on page 150
- “**idsxcfg**” on page 150
- “**idsxinst**” on page 151
- “**migbkup**” on page 151

Client utilities

- “**idsdirctl**, **ibmdirctl**” on page 6
- “**idsldapchangepwd**, **ldapchangepwd**” on page 7
- “**idsldapcompare**, **ldapcompare**” on page 11
- “**idsldapdelete**, **ldapdelete**” on page 13
- “**idsldapdiff**, **ldapdiff**” on page 17
- “**idsldapexop**, **ldapexop**” on page 26
- “**idsldapmodify**, **ldapmodify**, **idsldapadd**, **ldapadd**” on page 38
- “**idsldapmodrdn**, **ldapmodrdn**” on page 44
- “**idsrmlink**” on page 71
- “**idsldapsearch**, **ldapsearch**” on page 49
- “**idsldaptrace**, **ldaptrace**” on page 61
- “**idsversion**” on page 72
- “**idslink**” on page 65
- “**tbindmsg**” on page 73

Note: Tools that accept relative path as argument treats the path as relative to *instance_home*/idsslapd-*instance*/workdir directory. The only exception to this behavior is for the following utilities:

- **idsidrop**

- **idsideploy**
- **idsilist**
- **idsicrt**

Chapter 2. Client utilities

The 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.

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.

idsdirctl, ibmdirctl

Use the **idsdirctl** or **ibmdirctl** command to start or stop directory server.

Description

The **idsdirctl** or **ibmdirctl** command is an administration server control program.

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

Synopsis

```
ibmdirctl [options] command -- [ibmslapd options]
```

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

start
starts IBM Security Directory Server

stop
stops IBM Security Directory Server

restart
stops and then starts IBM Security Directory Server

status
indicates whether IBM Security Directory Server is running or stopped

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

admstop
stops the IBM Security Directory Server administration server

startlogmgmt
starts the log management capabilities for IBM Security Directory Server

stoplogmgmt
stops the log management capabilities for IBM Security Directory Server

statuslogmgmt
indicates whether the log management for IBM Security Directory Server is running

Usage

You can use the administration server control program, **ibmdirctl**, to start, stop, restart, or query the status of IBM Security 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 about the **idsdiradm** utility, see the Directory administration server section in the *Administering* section in the IBM Security Directory Server documentation.

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**ibmdirctl**, type `ibmdirctl -?`.

Options

The options to the **ibmdirctl** 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 with Transport Layer Security (TLS).

-Z Specifies to use a secure LDAP connection with Secure Sockets Layer (SSL).

-? 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:

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

Example 2:
To stop the server, run the following command:

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

idsldapchangepwd, ldapchangepwd

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

Description

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

Note:

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

Synopsis

```
idsldapchangepwd | ldapchangepwd -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 **ldapchangepwd** command.

-C *charset*
Specifies that the DNs supplied as parameter to the **ldapchangepwd** 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 Appendix B, “Supported IANA character sets,” on page 157. 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 153.

-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*. Where, *IDS_LDAP_HOME* is the installation path of IBM Security Directory Server. The value of the *IDS_LDAP_HOME* variable varies depending on operating system. The default path on various operating system is listed.

- AIX operating systems: `/opt/IBM/ldap/V6.3.1`
- HP-UX operating systems on Itanium: `/opt/IBM/ldap/V6.3.1`
- Linux operating systems: `/opt/ibm/ldap/V6.3.1`
- Solaris operating systems: `/opt/IBM/ldap/V6.3.1`
- Windows operating systems: `C:\Program Files\IBM\ldap\V6.3.1`

  Note: The `C:\Program Files\IBM\ldap\V6.3.1` path is the default installation location. The actual *IDS_LDAP_HOME* is determined during the installation.

For more information about the default key database files and default certificate authorities (CAs), see *Programming Reference* section of the IBM Security Directory Server 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 Server 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 Transport Layer Security (TLS). The **-Y** option is only supported when IBM GSKit is installed.

-Z Specifies to use a secure LDAP connection by using Secure Sockets Layer (SSL). The **-Z** option is only supported for SSL component entry, as provided by IBM 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, see"SSL and TLS notes" on page 73.

See also

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

Examples

Example 1:

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

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

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

idsldapcompare, ldapcompare

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

Description

The **ldapcompare** utility sends a compare request to an LDAP server. The **ldapcompare** 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 | ldapcompare[-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 **ldapcompare** command:

```
ldapcompare [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 **`ldapcompare`** 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 153.

`-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 **ldapcompare** command of the following format:

```
ldapcompare -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, ldapdelete

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

```
ldapdelete [options] [DNs]
ldapdelete [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 **ldapdelete** 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 Appendix B, “Supported IANA character sets,” on page 157. 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 153.

-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 *IBM Security Directory Server Version 6.3.1.5 Programming Reference*.

-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*. Where, *IDS_LDAP_HOME* is the installation path of IBM Security Directory Server. The value of the *IDS_LDAP_HOME* variable varies depending on the operating system. The default path on various operating system is listed.

- AIX operating systems: `/opt/IBM/ldap/V6.3.1`
- HP-UX operating systems on Itanium: `/opt/IBM/ldap/V6.3.1`
- Linux operating systems: `/opt/ibm/ldap/V6.3.1`
- Solaris operating systems: `/opt/IBM/ldap/V6.3.1`
- Windows operating systems: `C:\Program Files\IBM\ldap\V6.3.1`

Note: The`C:\Program Files\IBM\ldap\V6.3.1` path is the default installation location. The actual *IDS_LDAP_HOME* is determined during the installation.

For information about the default key database files and default certificate authorities (CAs), see *Programming Reference* section of the IBM Security Directory Server 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 Server 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 Server 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 *IBM Security Directory Server Version 6.3.1.5 Programming Reference*.

-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 Transport Layer Security (TLS).

-Z Specifies to use a secure LDAP connection by using Secure Sockets Layer (SSL).

-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 “SSL and TLS notes” on page 73.

See also

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

Examples

Example 1:

To delete an entry from a directory server instance, run the **ldapdelete** 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, ldapdiff

Use the **ldapdiff** 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 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 in IBM Security Directory Server, version 6.2 and later, 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 | ldapdiff -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 | ldapdiff -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*
Generate an LDIF file for output. 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`. If the key database file is not in the current directory, specify the fully qualified keystore file name. 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. 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`. If the keystore file is not in the current directory, specify the fully qualified keystore file name. 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`. If the trust database file is not in the current directory, specify the fully qualified truststore file name. 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. 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 "SSL, TLS notes" in the *IBM Security Directory Server Version 6.3.1.5 Command Reference*.

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:

Platform	Run this command:
AIX, Linux, Solaris, and HP-UX	idsldapdiff -b o=sample -sh supplier -sp 636 -sD cn=root -sw password -sZ -sK *pathname*/keyfile.jks -sP keyStorePwd -sN jks -sT *pathname*/keyfile.jks -sY trustStorePwd -st jks -ch consumer -cp 636 -cD cn=root -cw password -cZ -cK *pathname*/keyfile.jks -cP keyStorePwd -cN jks -cT *pathname*/keyfile.jks -cY trustStorePwd -ct jks -F -a
Windows	idsldapdiff -b o=sample -sh supplier -sp 636 -sD cn=root -sw password -sZ -sK *pathname*\keyfile.jks -sP keyStorePwd -sN jks -sT *pathname*\keyfile.jks -sY trustStorePwd -st jks -ch consumer -cp 636 -cD cn=root -cw password -cZ -cK *pathname*\keyfile.jks -cP keyStorePwd -cN jks -cT *pathname*\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:

Platform	Run this command:
AIX, Linux, Solaris, and HP-UX	idsldapdiff -S -sh supplier -sp 636 -sD cn=root -sw password -sZ -sK *pathname*/keyfile.jks -sP keyStorePwd -sN jks -sT *pathname*/keyfile.jks -sY trustStorePwd -st jks -ch consumer -cp 636 -cD cn=root -cw password -cZ -cK *pathname*/keyfile.jks -cP keyStorePwd -cN jks -cT *pathname*/keyfile.jks -cY trustStorePwd -ct jks

Platform	Run this command:
Windows	idsldapdiff -S -sh supplier -sp 636 -sD cn=root -sw password -sZ -sK *pathname*\keyfile.jks -sP keyStorePwd -sN jks -sT *pathname*\keyfile.jks -sY trustStorePwd -st jks -ch consumer -cp 636 -cD cn=root -cw password -cZ -cK *pathname*\keyfile.jks -cP keyStorePwd -cN jks -cT *pathname*\keyfile.jks -cY trustStorePwd -ct jks

idsldapexop, ldapexop

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

Description

The **ldapexop** 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 | ldapexop[-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 Appendix B, “Supported IANA character sets,” on page 157. 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 153.

-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*. Where, *IDS_LDAP_HOME* is the installation path of IBM Security Directory Server. The value of the *IDS_LDAP_HOME* variable varies depending on the operating system. The default path on various operating system is listed.

- AIX operating systems: `/opt/IBM/ldap/V6.3.1`
- HP-UX operating systems on Itanium: `/opt/IBM/ldap/V6.3.1`
- Linux operating systems: `/opt/ibm/ldap/V6.3.1`
- Solaris operating systems: `/opt/IBM/ldap/V6.3.1`
- Windows operating systems: `C:\Program Files\IBM\ldap\V6.3.1`

  Note: The `C:\Program Files\IBM\ldap\V6.3.1` path is the default installation location. The actual *IDS_LDAP_HOME* is determined during the installation.

For more information about the default key database files and default certificate authorities (CAs), see *Programming Reference* section of the IBM Security Directory Server 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 Server 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 Transport Layer Security (TLS). This option is supported only if IBM GSKit is installed.

-Z Specifies to use a secure LDAP connection by using Secure Sockets Layer (SSL). This option is supported only if IBM 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* value must be suspend, resume, or | replnow. For example:

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

suspend

Specifies to suspend replication.

resume

Specifies to resume the suspended replication.

replnow

Specifies to replicate now.

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 **ldapexop** command waits to read a list of DN 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, messages are written to the standard error.

Security functions

To use the SSL or TLS-related functions that are associated with this utility, see“SSL and TLS notes” on page 73.

See also

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

idsldapmodify, ldapmodify, idsldapadd, ldapadd

Use the **ldapadd** and **ldapmodify** commands to add and modify entries in directory server.

Description

The **ldapadd** command is an LDAP add-entry tool, and **ldapmodify** 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 | ldapmodify [-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 | ldapadd [-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 Appendix B, "Supported IANA character sets," on page 157. 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 153.

-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 Server 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*. Where, *IDS_LDAP_HOME* is the installation path of the IBM Security Directory Server. The value of the *IDS_LDAP_HOME* variable varies depending on the operating system. The default path on various operating system is listed.

- AIX operating systems: `/opt/IBM/ldap/V6.3.1`
- HP-UX operating systems on Itanium: `/opt/IBM/ldap/V6.3.1`
- Linux operating systems: `/opt/ibm/ldap/V6.3.1`
- Solaris operating systems: `/opt/IBM/ldap/V6.3.1`
- Windows operating systems: `C:\Program Files\IBM\ldap\V6.3.1`

Note: The `C:\Program Files\IBM\ldap\V6.3.1` path is the default installation location. The actual *IDS_LDAP_HOME* is determined during the installation.

For more information about the default key database files and default certificate authorities (CAs), see *Programming Reference* section of the IBM Security Directory Server 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 Server 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 Server 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 Transport Layer Security (TLS). This option is supported only if IBM GSKit is installed.

-Z Specifies to use a secure LDAP connection by using Secure Sockets Layer (SSL). This option is supported only if IBM 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“SSL and TLS notes” on page 73.

See also

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

Examples

Example 1:

Consider a `/tmp/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.of.life.edu
-
add: title
```

```
title: Grand Poobah
-
add: jpegPhoto
jpegPhoto: /tmp/modme.jpeg
-
delete: description
```

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

- Replace the mail attribute of the cn=Modify Me entry with the *modme@student.of.life.edu* value
- Add the title attribute with the *Grand Poobah* value
- Add a file /tmp/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 /tmp/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.of.life.edu
+title=Grand Poobah
+jpegPhoto=/tmp/modme.jpeg
-description
```

Example 3:

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

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

where, /tmp/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.of.life.edu
uid: jdoe
```

Example 4:

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

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

where, /tmp/removeentry.ldif contains:

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

idsldapmodrdn, ldapmodrdn

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

Description

The **ldapmodrdn** 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 | ldapmodrdn [-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 **ldapmodrdn** 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, seeAppendix B, “Supported IANA character sets,” on page 157. 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 153.

-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 Server 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*. Where, *IDS_LDAP_HOME* is the installation path of IBM Security Directory Server. The value of the *IDS_LDAP_HOME* variable varies depending on the operating system. The default path on various operating system is listed.

- AIX operating systems: `/opt/IBM/ldap/V6.3.1`
- HP-UX operating systems on Itanium: `/opt/IBM/ldap/V6.3.1`
- Linux operating systems: `/opt/ibm/ldap/V6.3.1`
- Solaris operating systems: `/opt/IBM/ldap/V6.3.1`
- Windows operating systems: `C:\Program Files\IBM\ldap\V6.3.1`

  Note: The `C:\Program Files\IBM\ldap\V6.3.1` path is the default installation location. The actual *IDS_LDAP_HOME* is determined during the installation.

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

-Z Specifies to use a secure LDAP connection by using Secure Sockets Layer (SSL).

-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, **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, messages are written to the standard error.

Security functions

To use the SSL or TLS-related functions that are associated with this utility, see“SSL and TLS notes” on page 73.

See also

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

Examples

Example 1:

Consider a file /tmp/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 /tmp/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.

idsldapsearch, ldapsearch

Use the **ldapsearch** 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 Server 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 `ldapsearch.c` file. The **`idsldapsearch`** utility rejects any filter size that is larger than 4 KB. If you want to change `ldapsearch.c` to handle a filter larger than 4 KB then change the following line in `ldapsearch.c`. For example, change

  ```
  #define FILTERSIZE 4096
  ```

  to

  ```
  #define FILTERSIZE 16000
  ```

  You must recompile `ldapsearch.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 **`ldapsearch`** 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 Appendix B, “Supported IANA character sets,” on page 157. 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 153.

`-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 Server 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*. Where, *IDS_LDAP_HOME* is the installation path of IBM Security Directory Server. The value of the *IDS_LDAP_HOME* variable varies depending on the operating system. The default path on various operating system is listed.

- AIX operating systems: `/opt/IBM/ldap/V6.3.1`
- HP-UX operating systems on Itanium: `/opt/IBM/ldap/V6.3.1`
- Linux operating systems: `/opt/ibm/ldap/V6.3.1`
- Solaris operating systems: `/opt/IBM/ldap/V6.3.1`
- Windows operating systems: `C:\Program Files\IBM\ldap\V6.3.1`

  Note: The `C:\Program Files\IBM\ldap\V6.3.1` path is the default installation location. The actual *IDS_LDAP_HOME* is determined during the installation.

For information about the default key database files and default certificate authorities (CAs), see *Programming Reference* section of the IBM Security Directory Server 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 Server 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
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 *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 Server 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 Server 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 Transport Layer Security (TLS). The **-Y** option is only supported when IBM 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). The **-Z** option is only supported for SSL component entry, as provided by IBM 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, see"SSL and TLS notes" on page 73.

See also

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

Examples

Some examples of the **ldapsearch** 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.

```
ldapsearch -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.

```
ldapsearch -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.

```
ldapsearch -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*.

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

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

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

idsldaptrace, ldaptrace

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

Description

The **ldaptrace** 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 **ldaptrace** 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 **ldaptrace** command uses system resources and affects the performance of the directory servers.
4. If the **ldaptrace** 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 | ldaptrace [-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 **ldaptrace** 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*. Where, *IDS_LDAP_HOME* is the installation path of IBM Security Directory Server. The value of the *IDS_LDAP_HOME* variable varies depending on the operating system. The default path on various operating system is listed.

- AIX operating systems: `/opt/IBM/ldap/V6.3.1`
- HP-UX operating systems on Itanium: `/opt/IBM/ldap/V6.3.1`
- Linux operating systems: `/opt/ibm/ldap/V6.3.1`
- Solaris operating systems: `/opt/IBM/ldap/V6.3.1`
- Windows operating systems: `C:\Program Files\IBM\ldap\V6.3.1`

  Note: The `C:\Program Files\IBM\ldap\V6.3.1` path is the default installation location. The actual *IDS_LDAP_HOME* is determined during the installation.

For information about the default key database files and default certificate authorities (CAs), see *IBM Security Directory Server Version 6.3.1.5 Programming Reference*.

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 Server 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:

```
C:\>ldtrc info
Trace Version:1.00
Op. System:NT
Op. Sys. Version:4.0
H/W Platform:80x86

Mask: *.*.*.*.*.*
pid.tid to trace: all
cpidto 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 destination file and location of the file is as provided.

On AIX, Linux, Solaris, and HP-UX (Itanium) systems:
`/var/ldap/ibmslapd.trace.dump`

On Windows systems:
installationpath`\var\ibmslapd.trace.dump`

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 153.

-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.

-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 Transport Layer Security (TLS). The **-Y** option is only supported when IBM GSKit is installed.

-Z Specifies to use a secure LDAP connection by using Secure Sockets Layer (SSL). The **-Z** option is only supported for SSL component entry, as provided by IBM 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, see"SSL and TLS notes" on page 73.

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
```

idslink

Use the **idslink** command to create links to LDAP client and server command-line utilities.

Description

The **idslink** command creates links to LDAP client and server command-line utilities. This utility is installed with the client package. During IBM Security Directory Server installation, links for client and server utilities are not set automatically. You can use the **idslink** command to set the links to command-line utilities such as **idsldapmodify** and **idsldapadd**, and libraries such as libibmldap. The links point to the location where the IBM Security Directory Server utilities and libraries are located. *installpath*/bin, *installpath*/sbin, and *installpath*/lib. Where,*installpath* is the IBM Security Directory Server installation location.

Note: The**idslink** utility overwrites any existing links with the updated links.

Synopsis

```
idslink [-i -g -l bits -s mode [-n] [-f] [-q] ] | -v | -h
```

Options

The options to the **idslink** command.

-i
Creates links for client command utilities that begin with *ids*. For example, link from /usr/bin/idsldapsearch to /opt/ibm/ldap/V6.3.1/bin/idsldapsearch.

-g
Creates links for client command utilities that do not begin with *ids*. For example, link from /usr/bin/ldapsearch to /opt/ibm/ldap/V6.3.1/bin/ldapsearch.

-l *bits*
Creates links for 32-bit or 64-bit client library files. The *bits* value can be 32 or 64.

-s *mode*
Creates links for server command-line utilities. The *mode* value can contain one of the following values.

- base to establish links for the base server code for use with the proxy or full server
- fullsrv if the directory server instance is a full server

-n
Specifies to demonstrate the action of the operation without actually running it. The **idslink** command with the **-n** option, lists the links that gets set. With this option, you must also specify one or more of the following options: **-i**, **-g**, or **-l**. After you run the command with the **-n** option, check the /var/idsldap/V6.3.1/idslink.preview file, which might contain any conflicts that are found during the operation.

-f
Specifies the force option. This option overrides existing files or links, and back up any conflicts. For example, /usr/bin/ldapsearch.

If you use the **-f** option, each conflicting link is backed up into a subdirectory with the same name as the file, directory, or link. For example, a conflict for the /usr/bin/ldapsearch command is backed up in a subdirectory called /usr/bin/V6.3.1_idslink_bkup_*timestamp*. Where, *timestamp* is the date and time when the backup was created.

If you do not use this option and conflicts with existing links are found, none of the links in the group are set.

-q
Specifies to run in quiet mode. All output is suppressed except for error messages.

-v Specifies to show the version information of the command.

-h
Specifies to show the help.

Links created by `idslink`

When you run the **idslink** command, it creates links to client utilities, server utilities, and libraries. An example list links created by using the **idslink** command.

Note: The installation path on AIX, Linux, Solaris, and HP-UX (Itanium) systems is /opt/*ibmdir*/ldap/V6.3.1/. Where, the value of the *ibmdir* variable is IBM on AIX, Solaris, and HP-UX (Itanium) systems, and ibm on Linux systems.

Client commands

Links that are created for client commands (that do not begin with ids) for the base client when the **-g** option is specified.

/usr/bin/ldapsearch -> /opt/*ibmdir*/ldap/V6.3.1/bin/ldapsearch

/usr/bin/ldapadd -> /opt/*ibmdir*/ldap/V6.3.1/bin/ldapadd

/usr/bin/ldapmodify -> /opt/*ibmdir*/ldap/V6.3.1/bin/ldapmodify

/usr/bin/ldapdelete -> /opt/*ibmdir*/ldap/V6.3.1/bin/ldapdelete

/usr/bin/ldapmodrdn -> /opt/*ibmdir*/ldap/V6.3.1/bin/ldapmodrdn

/usr/bin/ldapchangepwd -> /opt/*ibmdir*/ldap/V6.3.1/bin/ldapchangepwd

/usr/bin/ldaptrace -> /opt/*ibmdir*/ldap/V6.3.1/bin/ldaptrace

/usr/bin/ldapexop -> /opt/*ibmdir*/ldap/V6.3.1/bin/ldapexop

/usr/bin/ibmdirctl -> /opt/*ibmdir*/ldap/V6.3.1/bin/ibmdirctl

Links that are created for client commands (that begin with ids) for the base client when the **-i** option is specified.

/usr/bin/idsldapsearch -> /opt/*ibmdir*/ldap/V6.3.1/bin/idsldapsearch

/usr/bin/idsldapadd -> /opt/*ibmdir*/ldap/V6.3.1/bin/idsldapadd

/usr/bin/idsldapmodify -> /opt/*ibmdir*/ldap/V6.3.1/bin/idsldapmodify

/usr/bin/idsldapdelete -> /opt/*ibmdir*/ldap/V6.3.1/bin/idsldapdelete

/usr/bin/idsldapmodrdn -> /opt/*ibmdir*/ldap/V6.3.1/bin/idsldapmodrdn

/usr/bin/idsldapchangepwd -> /opt/*ibmdir*/ldap/V6.3.1/bin/idsldapchangepwd

/usr/bin/idsldaptrace -> /opt/*ibmdir*/ldap/V6.3.1/bin/idsldaptrace

/usr/bin/idsldapexop -> /opt/*ibmdir*/ldap/V6.3.1/bin/idsldapexop

/usr/bin/idsdirctl -> /opt/*ibmdir*/ldap/V6.3.1/bin/idsdirctl

Client libraries

Note: In the library path, *XX* denotes the library extension, such as .so, .a, or .sl.

Links created when the -l 32 option is specified

The following groups or sets of links are created when the **-l** *bits* option is specified, where *bits* is 32.

Note: Links common to all operating systems and links that are specific to a particular operating system are in one group or set.

Client libraries: Set of links for 32-bit client package

Common links:

/usr/lib/libidsldap.*XX* -> /opt/*ibmdir*/ldap/V6.3.1/lib/libidsldap.*XX*

/usr/lib/libidsldapstatic.*XX* -> /opt/*ibmdir*/ldap/V6.3.1/lib/libidsldapstatic.*XX*

/usr/lib/idsldap_plugin_sasl_digest-md5.*XX* -> /opt/*ibmdir*/ldap/V6.3.1/lib/idsldap_plugin_sasl_digest-md5.*XX*

Operating system-specific links:

/usr/lib/idsldap_plugin_ibm_gsskrb.*XX* -> /opt/*ibmdir*/ldap/V6.3.1/lib/idsldap_plugin_ibm_gsskrb.*XX* (Kerberos library file on AIX)

/usr/lib/libidsldif.*XX* -> /opt/*ibmdir*/ldap/V6.3.1/lib/libidsldif.*XX* (Linux and HP-UX only)

Client libraries: Set of links for 32-bit client package (support for compatibility with earlier versions)

Common links:

/usr/lib/libldap.*XX* -> /opt/*ibmdir*/ldap/V6.3.1/lib/libidsldap.*XX*

/ldap/V6.3.1/lib/libidsldap.*XX* -> /opt/*ibmdir*/ldap/V6.3.1/lib/libidsldap.*XX*

/usr/lib/libibmldapstatic.*XX* -> /opt/*ibmdir*/ldap/V6.3.1/lib/libidsldapstatic.*XX*

/usr/lib/libldapiconv.*XX* -> /opt/*ibmdir*/ldap/V6.3.1/lib/libidsldapiconv.*XX*

/usr/lib/ldap_plugin_sasl_digest-md5.*XX* -> /opt/*ibmdir*/ldap/V6.3.1/lib/idsldap_plugin_sasl_digest-md5.*XX*

Operating system-specific links:

/usr/lib/ldap_plugin_ibm_gsskrb.*XX* -> /opt/*ibmdir*/ldap/V6.3.1/lib/idsldap_plugin_ibm_gsskrb.*XX* (Kerberos library file on AIX)

/usr/lib/libldif.*XX* -> /opt/*ibmdir*/ldap/V6.3.1/lib/libidsldif.*XX* (Linux and HP-UX only)

Links created when the -l 64 option is specified

The following groups or sets of links are created when the **-l** *bits* option is specified, where *bits* is 64.

Client libraries: Set of links with 64 in name for 64-bit client package

Common links:

`/usr/lib/libidsldap64.`*`XX`* -> `/opt/`*`ibmdir`*`/ldap/V6.3.1/lib64/libidsldap.`*`XX`*

`/usr/lib/libidsldapstatic64.`*`XX`* -> `/opt/`*`ibmdir`*`/ldap/V6.3.1/lib64/libidsldapstatic.`*`XX`*

`/usr/lib/idsldap_plugin_sasl_digest-md5_64.`*`XX`* -> `/opt/`*`ibmdir`*`/ldap/V6.3.1/lib64/idsldap_plugin_sasl_digest-md5.`*`XX`*

Operating system-specific links:

`/usr/lib/idsldap_plugin_ibm_gsskrb_64.`*`XX`* -> `/opt/`*`ibmdir`*`/ldap/V6.3.1/lib64/idsldap_plugin_ibm_gsskrb.`*`XX`* (Kerberos library file on AIX)

`/usr/lib/libidsldif64.`*`XX`* -> `/opt/`*`ibmdir`*`/ldap/V6.3.1/lib64/libidsldif.`*`XX`* (Linux and HP-UX only)

Client libraries: Set of links with `64` in name for 64-bit client package (support for compatibility with earlier versions)

Common links:

`/usr/lib/libldap64.`*`XX`* -> `/opt/`*`ibmdir`*`/ldap/V6.3.1/lib64/libidsldap.`*`XX`*

`/usr/lib/libibmldap64.`*`XX`* -> `/opt/`*`ibmdir`*`/ldap/V6.3.1/lib64/libidsldap.`*`XX`*

`/usr/lib/libibmldapstatic64.`*`XX`* -> `/opt/`*`ibmdir`*`/ldap/V6.3.1/lib64/libidsldapstatic.`*`XX`*

`/usr/lib/libldapiconv64.`*`XX`* -> `/opt/`*`ibmdir`*`/ldap/V6.3.1/lib64/libidsldapiconv.`*`XX`*

`/usr/lib/ldap_plugin_sasl_digest-md5_64.`*`XX`* -> `/opt/`*`ibmdir`*`/ldap/V6.3.1/lib64/idsldap_plugin_sasl_digest-md5.`*`XX`*

Operating system-specific links:

`/usr/lib/ldap_plugin_ibm_gsskrb_64.`*`XX`* -> `/opt/`*`ibmdir`*`/ldap/V6.3.1/lib64/idsldap_plugin_ibm_gsskrb.`*`XX`* (Kerberos library file on AIX)

`/usr/lib/libldif64.`*`XX`* -> `/opt/`*`ibmdir`*`/ldap/V6.3.1/lib64/libidsldif.`*`XX`* (Linux and HP-UX only)

Client libraries: Set of links without `64` in name for 64-bit client package

Common links:

`/usr/lib/lib64/libidsldap.`*`XX`* -> `/opt/`*`ibmdir`*`/ldap/V6.3.1/lib64/libidsldap.`*`XX`*

`/usr/lib/lib64/libidsldapstatic.`*`XX`* -> `/opt/`*`ibmdir`*`/ldap/V6.3.1/lib64/libidsldapstatic.`*`XX`*

`/usr/lib/lib64/idsldap_plugin_sasl_digest-md5.`*`XX`* -> `/opt/`*`ibmdir`*`/ldap/V6.3.1/lib64/idsldap_plugin_sasl_digest-md5.`*`XX`*

Operating system-specific links:

`/usr/lib/lib64/idsldap_plugin_ibm_gsskrb.`*`XX`* -> `/opt/`*`ibmdir`*`/ldap/V6.3.1/lib64/idsldap_plugin_ibm_gsskrb.`*`XX`* (Kerberos library file on AIX)

/usr/lib/lib64/libidsldif.*XX* -> /opt/*ibmdir*/ldap/V6.3.1/lib64/libidsldif.*XX* (Linux and HP-UX only)

Client libraries: Set of links without 64 in name for 64-bit client package (support for compatibility with earlier versions)

Common links:

/usr/lib/lib64/libldap.*XX* -> /opt/*ibmdir*/ldap/V6.3.1/lib64/libidsldap.*XX*

/usr/lib/lib64/libibmldap.*XX* -> /opt/*ibmdir*/ldap/V6.3.1/lib64/libidsldap.*XX*

/usr/lib/lib64/libibmldapstatic.*XX* -> /opt/*ibmdir*/ldap/V6.3.1/lib64/libidsldapstatic.*XX*

/usr/lib/lib64/libldapiconv.*XX* -> /opt/*ibmdir*/ldap/V6.3.1/lib64/libidsldapiconv.*XX*

/usr/lib/lib64/ldap_plugin_sasl_digest-md5.*XX* -> /opt/*ibmdir*/ldap/V6.3.1/lib64/idsldap_plugin_sasl_digest-md5.*XX*

Operating system-specific links:

/usr/lib/lib64/ldap_plugin_ibm_gsskrb.*XX* -> /opt/*ibmdir*/ldap/V6.3.1/lib64/idsldap_plugin_ibm_gsskrb.*XX* (Kerberos library file on AIX)

/usr/lib/lib64/libldif.*XX* -> /opt/*ibmdir*/ldap/V6.3.1/lib64/libidsldif.*XX* (Linux and HP-UX only)

Server commands

Links that are created for server commands for base server package when the **-s** base option is specified.

/usr/bin/slapd -> /opt/*ibmdir*/ldap/V6.3.1/sbin/slapd

/usr/bin/ibmslapd -> /opt/*ibmdir*/ldap/V6.3.1/sbin/ibmslapd

/usr/bin/idsslapd -> /opt/*ibmdir*/ldap/V6.3.1/sbin/idsslapd

/usr/bin/ibmdiradm -> /opt/*ibmdir*/ldap/V6.3.1/sbin/ibmdiradm

/usr/bin/idsdiradm -> /opt/*ibmdir*/ldap/V6.3.1/sbin/idsdiradm

/usr/bin/ldtrc -> /opt/*ibmdir*/ldap/V6.3.1/sbin/ldtrc

/usr/bin/ddsetup -> /opt/*ibmdir*/ldap/V6.3.1/sbin/ddsetup

/usr/bin/idsxcfg -> /opt/*ibmdir*/ldap/V6.3.1/sbin/idsxcfg

/usr/bin/idsxinst -> /opt/*ibmdir*/ldap/V6.3.1/sbin/idsxinst

/usr/bin/idsilist -> /opt/*ibmdir*/ldap/V6.3.1/sbin/idsilist

/usr/bin/idsicrt -> /opt/*ibmdir*/ldap/V6.3.1/sbin/idsicrt

/usr/bin/idsidrop -> /opt/*ibmdir*/ldap/V6.3.1/sbin/idsidrop

/usr/bin/idsdnpw -> /opt/*ibmdir*/ldap/V6.3.1/sbin/idsdnpw

/usr/bin/idssetport -> /opt/*ibmdir*/ldap/V6.3.1/sbin/idssetport

/usr/bin/idssethost -> /opt/*ibmdir*/ldap/V6.3.1/sbin/idssethost

/usr/bin/idsimigr -> /opt/*ibmdir*/ldap/V6.3.1/sbin/idsimigr

/usr/bin/idscfgsch -> /opt/*ibmdir*/ldap/V6.3.1/sbin/idscfgsch

/usr/bin/idsucfgsch -> /opt/*ibmdir*/ldap/V6.3.1/sbin/idsucfgsch

/usr/bin/idslogmgmt -> /opt/*ibmdir*/ldap/V6.3.1/sbin/idslogmgmt

/usr/bin/idsgendirksf -> /opt/*ibmdir*/ldap/V6.3.1/sbin/idsgendirksf

Links that are created for server commands for full server package when the **-s** fullsrv option is specified.

```
/usr/bin/bulkload -> /opt/ibmdir/ldap/V6.3.1/sbin/bulkload
/usr/bin/idsbulkload -> /opt/ibmdir/ldap/V6.3.1/sbin/idsbulkload
/usr/bin/ldif2db -> /opt/ibmdir/ldap/V6.3.1/sbin/ldif2db
/usr/bin/idsldif2db -> /opt/ibmdir/ldap/V6.3.1/sbin/idsldif2db
/usr/bin/db2ldif -> /opt/ibmdir/ldap/V6.3.1/sbin/db2ldif
/usr/bin/idsdb2ldif -> /opt/ibmdir/ldap/V6.3.1/sbin/idsdb2ldif
/usr/bin/dbback -> /opt/ibmdir/ldap/V6.3.1/sbin/dbback
/usr/bin/idsdbback -> /opt/ibmdir/ldap/V6.3.1/sbin/idsdbback
/usr/bin/dbrestore -> /opt/ibmdir/ldap/V6.3.1/sbin/dbrestore
/usr/bin/idsdbrestore -> /opt/ibmdir/ldap/V6.3.1/sbin/idsdbrestore
/usr/bin/runstats -> /opt/ibmdir/ldap/V6.3.1/sbin/runstats
/usr/bin/idsrunstats -> /opt/ibmdir/ldap/V6.3.1/sbin/idsrunstats
/usr/bin/idscfgdb -> /opt/ibmdir/ldap/V6.3.1/sbin/idscfgdb
/usr/bin/idsucfgdb -> /opt/ibmdir/ldap/V6.3.1/sbin/idsucfgdb
/usr/bin/idscfgchglg -> /opt/ibmdir/ldap/V6.3.1/sbin/idscfgchglg
/usr/bin/idsucfgchglg -> /opt/ibmdir/ldap/V6.3.1/sbin/idsucfgchglg
/usr/bin/idscfgsuf -> /opt/ibmdir/ldap/V6.3.1/sbin/idscfgsuf
/usr/bin/idsucfgsuf -> /opt/ibmdir/ldap/V6.3.1/sbin/idsucfgsuf
```

idsrmlink

Use the **idsrmlink** command to remove links to the client and server utilities.

Description

You can use the **idsrmlink** command to remove links to the client and server utilities that were set by the **idslink** command.

Note: The **idsrmlink** command does not restore any links that are previously backed up by **idslink** when run with the force option.

Synopsis

```
installpath/bin/idsrmlink [-i -g -l bits -s mode [-n] [-q]] | -v | -h
```

where, *installpath* is the path where IBM Security Directory Server is installed.

Options

The options to the **idslink** command.

-i
: Removes links for client command utilities that begin with *ids*.

-g
: Removes links for client command utilities that do not begin with *ids*.

-l *bits*
: Removes links for 32-bit or 64-bit client library files. The *bits* value can be 32 or 64.

-s *mode*
Removes links for server command-line utilities. The *mode* value can contain one of the following values.

- proxy if the directory server instance is a proxy server
- fullsrv if the directory server instance is a full server

-n
Specifies to demonstrate the action of the operation without actually implementing it. The **idsrmlink** command with the **-n** option, lists the links that are removed if the command is run.

-q
Specifies to run in quiet mode. All output is suppressed except for error messages.

-v Specifies to show the version information of the command.

-h
Specifies to show the help for the command.

idsversion

Use the **idsversion** to see the version of IBM Security Directory Server components.

Description

The **idsversion** command lists the versions of all IBM Security Directory Server components that are installed on a computer. IBM Security Directory Server components include base, client, servers, proxy server, Web Administration Tool, and language packages.

Synopsis

```
idsversion [[-b outputfile] [-d] [-r] [-t tmpOutDir]]| -v | -?
```

Options

The options to the **idsversion** command.

-b *outputfile*
Specifies the absolute path of a file for redirecting output.

-d Specifies to set the debug option.

-r
Lists the full information about each IBM Security Directory Server component. With the **-r** option, the information is shown in raw format.

-t *tmpOutDir*
Specifies a directory for storing intermediate data during the operation.

-v Specifies to list the version of the command.

-? Specifies to show the syntax format.

Examples

Example 1:
To list the version of the installed IBM Security Directory Server components on a 32-bit architecture in a raw format, run the **idsversion** command with the **-r** option.

```
idsversion –r
```

```
TDS_CLTJAVA#6.3.1.0
TDS_SRVPROXY#6.3.1.0
TDS_WEBADMIN#6.3.1.0
TDS_CLTBASE#6.3.1.0
TDS_SERVER32#6.3.1.0
TDS_LANGUAGE_EN#6.3.1.0
TDS_CLIENT32#6.3.1.0
```

Example 2:

To redirect the version information to another file, run the following command:

```
idsversion –b filename
```

The **idsversion** command with the **-b** option redirects the version information for the installed IBM Security Directory Server components on a 32-bit architecture to a file specified by *filename*. For IBM Security Directory Server installed components, the file that is specified by *filename* might contain the following entries:

```
TDS java client version:6.3.1.0
32-bit TDS proxy server version:6.3.1.0
TDS Web-adminserver version:6.3.1.0
TDS base client version:6.3.1.0
32-bit TDS server version:6.3.1.0
TDS language(en) package version:6.3.1.0
32-bit TDS client version:6.3.1.0
```

tbindmsg

The **tbindmsg** command is used internally by LDAP utilities to get a message from local message catalog.

Synopsis

```
tbindmsg catalog_name set_num msg_num def_fmt [arg ...]
```

Description

The **tbindmsg** command-line tool gets a message from a local message catalog and combines the arguments from the command line. All arguments to the command must be in string format. This command is internally used by LDAP utilities.

Options

catalog_name

set_num

msg_num

def_fmt

arg

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 IBM Global Security Kit (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 *IBM Security Directory Server Version 6.3.1.5 Programming Reference*. 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 3. Server utilities

Understand the behavior and usage of server utilities and the considerations for synchronizing the directory server instances.

For server utilities that support multiple directory instances on a system, the **-I** parameter is optional for any of the following conditions:

- If the *IDS_LDAP_INSTANCE* environment variable is set.
- If there is only one instance on the system.

For the **idsicrt** and **idsidrop** server utilities, you must provide the **-I** parameter.

Attention: When you create a directory server instance, consider the following points.

- You must cryptographically synchronize the directory server instances to obtain better performance, if you are using any of the following features:
  - Replication
  - Distributed directory
  - Import and export LDIF data between server instances
- You must cryptographically synchronize a directory server instance with an existing server before you do any of the following actions:
  - Start the second server instance.
  - Run the **idsbulkload** command from the second server instance.
  - Run the **idsldif2db** command from the second server instance.

For information about synchronizing directory server instances, see Appendix A, “Synchronizing two-way cryptography between server instances,” on page 155.

ddsetup

Use the **ddsetup** to split an LDIF file for loading it in distributed directories.

Description

The **ddsetup** command splits a lightweight directory information 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**, **db2ldif**" on page 106.

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. Create a proxy server instance. Run the **idsicrt** command, for example:

   ```
   idsicrt -I proxy_instance -x -l instance_location -G idsldap -w proxyPW
   ```

   where,

   proxy_instance is the proxy server instance and also the name of the proxy instance owner

   proxyPW is the password of the proxy instance owner

2. 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

3. 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

4. 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

5. 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

6. 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.

7. 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
   ```

idsadduser

Use the **idsadduser** command to create a user account on an operating system.

Description

The **idsadduser** command creates an operating system user with attributes that meet the requirements of a directory server instance owner. The **idsadduser** command can be run by a root user only on UNIX or a member of the Administrators group on Windows.

Note:

- If you create a user with the name of an existing user by using **idsadduser**, a message is shown indicating that the user exists. In such case, you can choose to overwrite an existing user with modified properties or exit.
- On Windows 2008, **idsadduser** adds the instance owner in the default DB2 security groups DB2ADMNS and DB2USERS.

Synopsis

```
idsadduser [–u username [-w password] [ –l homedir ] –g groupname]
          [-d debuglevel] [-b outputfile] [-q] [-n]] | -v | -?
```

Options

-b *outputfile*
: Specifies a file with full path to redirect output. If you use this parameter with the **-q** parameter, error messages are sent to the *outputfile* file. If the debug mode is on, debug output is also sent to this file.

-d *debuglevel*
: Sets the debug level. The **ldtrc** command must be running, when you use this parameter.

-g *groupname*
: Specifies the primary group of the user. This parameter is valid only on AIX, Linux, and Solaris systems.

-l *homedir*
: Specify the home directory of the user. The default value for the home directory on AIX and Linux is /home/username. On Solaris, the default value for the home directory is /export/home/username. This parameter is valid only on AIX, Linux, and Solaris systems.

-n
: Specifies to run in no-prompt mode. All output is generated except for the messages that require user interaction. The **-w** parameter must be used with the **-n** parameter.

-q
: Specifies to run in quiet mode. All output except error is suppressed. If the **-d** parameter is also specified, then the trace output is not suppressed.

-u *username*
Specifies the user name to create on the operating system.

-v
Prints the version information of the command.

-?
Specifies to show the syntax help.

Examples

Example 1:
To create a user on a UNIX system with the user details, run the **idsadduser** command.

- User name: inst1
- Primary group: staff
- Home directory: /home/inst1
- Password: inst123

```
idsadduser –u inst1 –g staff –l /home/inst1 –w inst123
```

Example 2:
To provide password that is not visible as clear text on the command prompt, run the **idsadduser** command of the following format:

```
idsadduser –u inst1 –g staff –l /home/inst1
```

After you run the command, the command prompts you to enter the password.

Example 3:
To create a user on Windows, run the **idsadduser** command of the following format:

```
idsadduser –u inst1 –w inst123
```

idsadscfg

Use the **idsadscfg** command to configure synchronization of Active Directory with IBM Security Directory Server.

Description

From IBM Security Directory Server, version 6.3.1, the Active Directory synchronization solution is deprecated. Use the LDAPSync solution instead.

You can use the **idsadscfg** command to configure the directory endpoint properties for AssemblyLines and EventHandlers associated with a directory server instance. During IBM Security Directory Server installation, the adsync.xml and adsync_cfg.xml files are created at the following location:

On AIX, Linux, and Solaris systems
opt/IBM/ldap/V6.3.1/idstools/adsynch

On Windows systems
DS_install_directory\idstools\adsync

On issuing the **idsadscfg** command, these files are copied to the instance directory. For example, the adsync.xml and adsync_cfg.xml files are copied to the *inst_home_dir*\etc\tdisoldir\config location on Windows systems. The files at

IBM Security Directory Server installation and instance home directory locations are independent of each other. Instance-specific operation uses files from the instance directory.

Note: Error handling actions of Active Directory-related arguments are considered during the run time (**idsadsrun**), and not during the configuration (**idsadscfg**). If any errors are reported during the run time, the solution must be configured again by using **idsadscfg** with the correct arguments.

Synopsis

```
idsadscfg [-I instance_name
          -adb AD_Search_Base_DN        -adD AD_Login_DN
          -adg AD_Group_Container_DN    -adH AD_LDAP_URL
          -adu AD_User_Container_DN     -adw AD_Login_Password
          -idsg TDS_Group_Container_DN  -idss TDS_Suffix
          -idsu TDS_User_Container_DN  [-Z]]
          [-d debug_level] [-b output_file] [-q] [-n]]
          | [-isCfg] | -v | -?
```

Options

The **idsadscfg** command takes the following parameters.

-adb *AD_Search_Base_DN*
Specifies the subtree in Active Directory from which the Active Directory sync solution propagates the changes.

-adD *AD_Login_DN*
Specifies the Active Directory login name.

-adg *AD_Group_Container_DN*
Specifies a list of LDAP subtrees in Active Directory from which groups in Active Directory gets synchronized to IBM Security Directory Server.

-adH *AD_LDAP_URL*
Specifies the LDAP URL and port of the Active Directory domain controller.

-adu *AD_User_Container_DN*
Specifies the DN of a container in Active Directory that contains the user entries for synchronization with IBM Security Directory Server.

-adw *AD_Login_Password*
Specifies a password for the Active Directory login name.

-b *output_file*
Specifies the full path of a file to redirect output. If this parameter is used with the **-q** parameter, only errors are sent to the file.

-d *debug_level*
Sets the debug level. Use this parameter with the **ldtrc** command.

-I *instance_name*
Specifies the name of the directory server instance to synchronize.

-idsg *TDS_Group_Container_DN*
Specifies the container DN for IBM Security Directory Server groups to synchronize and store Active Directory groups. This container must exist in IBM Security Directory Server.

-idss *TDS_Suffix*
Specifies the suffix in IBM Security Directory Server. The value of this parameter is populated internally.

-idsu *TDS_User_Container_DN*
Specifies the container DN for IBM Security Directory Server users to synchronize and store Active Directory users. This container must exist in IBM Security Directory Server.

-isCfg
Returns configuration status of Active Directory synchronize solution for an instance.

-n
Specifies to run in no prompt mode.

-q
Specifies to run in quiet mode. All output messages except errors are suppressed.

-v
Prints the version information for the command.

-Z
Species to use an SSL connection to connect to Active Directory.

-?
Specifies to show the syntax help.

idsadsrun

Use the **idsadsrun** command to synchronize users and groups from Active Directory server to IBM Security Directory Server. You can use the **idsadsrun** command to start `EventHandler` of a specified instance.

Synopsis

```
idsadsrun -I instancename [-d debuglevel] [-b outputfile] [-k] [-q] [-n] | -v | -?
```

Options

Note: From IBM Security Directory Server, version 6.3.1, the Active Directory synchronization solution is deprecated. Use the LDAPSync solution instead.

The **idsadsrun** command takes the following parameters.

-b *outputfile*
Specifies the full path of a file to redirect 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 sent to the *outputfile* file.

-d *debuglevel*
Sets the debug level in the LDAP library. The **ldtrc** command must be running, when you use this parameter.

-I *instancename*
Specifies the directory server instance to update.

-k
Stops the Active Directory sync solution that is associated with the instance.

-n
Specifies to run in no prompt mode. You must use this parameter with the **-q** parameter.

-q
: Specifies to run in quiet mode. All outputs except the errors are suppressed. If the **-d** parameter is specified along with **-q** , trace output is not suppressed.

-v
: Prints the version information of the command.

-?
: Specifies to show the syntax help.

idsbulkload, bulkload

Use the **bulkload** 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 Appendix A, "Synchronizing two-way cryptography between server instances," on page 155.

You must consider the following points before you use the **bulkload** 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 no longer supported in IBM Security Directory Server, version 6.0 and later. 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.

  The ACL processing enhancement in **idsbulkload** with the **-A** parameter is deprecated in IBM Security Directory Server, version 6.0 and later. The following parameters are also deprecated.

  - **-c**
  - **-C**
  - **-e**
- You must run the **idsbulkload** command with dbadm or sysadm privilege. On a Windows system, you must run the **idsbulkload** command within the DB2 command-line interpreter (CLI). To start the DB2 CLI, click **Start** > **Run**, type db2cmd, and click **OK**.
- 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 | bulkload -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>]] | [-?]
```

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 153.

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 `IDS_LDAP_HOME/examples/sample.ldif` file contains sample data in the LDIF format. The *IDS_LDAP_HOME* variable contains the path of IBM Security Directory Server installation location. The value of *IDS_LDAP_HOME* varies depending on the operating system. The default path on various operating system is listed.

- AIX operating systems: `/opt/IBM/ldap/V6.3.1`
- HP-UX operating systems on Itanium: `/opt/IBM/ldap/V6.3.1`
- Linux operating systems: `/opt/ibm/ldap/V6.3.1`
- Solaris operating systems: `/opt/IBM/ldap/V6.3.1`
- Windows operating systems: `C:\Program Files\IBM\ldap\V6.3.1`

  Note: The `C:\Program Files\IBM\ldap\V6.3.1` path is the default installation location. The actual *IDS_LDAP_HOME* is determined during the installation.

-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 path for the temporary storage location varies depending on the operating system.

On AIX, Linux, and Solaris systems
The default location is *instance_home_directory*`/idsslapd-`*instance_name*`/tmp/ldapimport`.

Note: If you log in as root, the **idsbulkload** command fails when you specify the location of the temporary directory by using the **–L** parameter. You must log in as an instance owner to create a temporary directory, and then run the **idsbulkload** command with root privileges. To log in as an instance owner, issue the following command.

```
su instance_name
```

On Windows systems
On Windows systems, the default location is *instance_home_directory*`\idsslapd-`*instance_name*`\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 when the DB2 LOAD tool parses the default **idsbulkload** string delimiter, vertical bar (|), in multi-byte character sets. In such case, reassign the string delimiter to $. To assign a delimiting character, use the following example.

```
idsbulkload -i ldiffile -I instancename -s $
```

-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.

-? 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 default path of the temporary storage location varies depending on the operating system. See the **-L** parameter description for the path names. You can change the path by using the **-L** parameter. For example:

```
idsbulkload -i ldiffile -I instancename -L /newpath
```

Before you run the command, ensure that you set write permission to the directory specified by using the **-L** parameter. You must also ensure that a minimum temporary storage size of 2.5 times the size of the LDIF file is available in the directory. More temporary storage might be required depending on your data. If you receive the following error, for example:

```
SQL3508N Error in accessing a file of type "SORTDIRECTORY" during load
or load query.Reason code: "2".Path: "/u/ldapdb2/sqllib/tmp/".
```

You must set the *DB2SORTTMP* environment variable to point to a directory with more space for usage during the bulkload operation. You can also specify multiple directories that are separated by a comma (**,**). For example:

```
export DB2SORTTMP=/sortdir1,/sortdir2
```

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 `*IDS_LDAP_HOME*/examples/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 varies depending on the operating system.

- On Windows systems, the log file is in the *instance_home_directory*\idsslapd-*instance_name*\tmp\ldapimport directory.

  Note: You can change the default path on Windows systems.
- On AIX, Linux, and Solaris systems the log file is the *instance_home_directory*/idsslapd-*instance_name*/tmp/ldapimport directory.

If the **-L** parameter is specified, the file in the directory that is defined by the **-L** parameter. 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.

- On Windows systems, the file is the *instance_home_directory*\idsslapd-*instance_name*\logs\bulkload_status directory.
- On AIX, Linux, and Solaris systems the file is in the *instance_home_directory*/idsslapd-*instance_name*/logs/bulkload_status directory.

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 Server 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.

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.

Note: Use the **idsicrt** command or the **idsxinst** utility to create a database instance.

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 | -?
```

Options

The **idscfgchglg** 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 the debug mode is set, the debug output is also 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 153.

-f *configfile*
Specifies the full path to the configuration file that is to be updated. If this option is not specified, the default configuration file for the directory server instance is used.

-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.

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 run the **idsicrt** command successfully before you create the database instance. You must also set the database instance owner correctly. Otherwise, the **idscfgdb** command fails. For more information about setting up required users and groups, see the *Installing and Configuring* section in the IBM Security Directory Server 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.

If you configure online backup for an instance by using Instance Administration Tool or Configuration Tool, you can unconfigure it using Configuration Tool or the **idscfgdb** command.

For the reliable results, use Instance Administration Tool or Configuration Tool to administer online backup.

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 instancename [[-w dbadminpw] [-a dbadminid -t dbname -l dblocation
        [-x]]] [-c] [-k backup_dir] [-m ts_type] [-u usr_ts_loc] [-U usr_ts_size]
        [-r ldap_ts_loc] [-R ldap_ts_size] [-z ext_size][-f configfile] [-d debuglevel]
        [-b outputfile] [-q] [-n]] | -v | -?
```

Options

The **idscfgdb** command takes the following parameters.

-a *dbadminid*
Specifies the DB2 administrator ID. The DB administrator must exist on the system and must have the appropriate permissions.

-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 also sent to this file.

-c
Removes the online backup configuration setup of the database.

Note: The **-c** parameter must not be used along with the **-a**, **-t**, and **-l** parameters, if the database is already configured.

-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 153.

-f *configfile*
Specifies the full path of the configuration file that must be updated. If this parameter is not specified, the default configuration file of a directory server instance is used.

-I *instancename*
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*.

-l *dblocation*
Specifies the DB2 database location. On AIX, Linux, or Solaris systems, the location is a directory name, for example `/home/ldapdb2`. On Windows systems, the location must be a drive letter. The database requires a minimum of 80-MB free space. More disk space might be required for as directory entries are added to database.

-m *ts_type*
Specifies the table space type for `USERSPACE1` and `LDAPSPACE`. A valid value for table space type is `DMS` or `SMS`. The default value for the table space type is `DMS`.

-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.

-r *ldap_ts_loc*
Specifies the `LDAPSPACE` container location. The container can be a non-existing file or a raw device. The default container is a file, for example: `[database location]/ldap32kcont_[database name]/ldapspace`.

-R *ldap_ts_size*
Specifies the container size of `LDAPSPACE` table space in pages.

Note: The default page size for the `LDAPSPACE` table space is 32 KB per page. If you change the default value, ensure that you have enough disk space for the values that are specified or the command might fail. The free disk space must be available in the location where you create the database.

-t *dbname*
Specifies the DB2 database name.

-u *usr_ts_loc*
Specifies the `USERSPACE1` container location. The container can be a non-existing file or a raw device. The default container is a file in the default container directory of the database.

-U *usr_ts_size*
Specifies the container size of `USERSPACE1` table space in pages.

Note: The default page size for the `USERSPACE1` table space is 4 KB per page. If you change the default value, ensure that you have enough disk space for the values that are specified or the command might fail. The free disk space must be available in the location where you create the database.

-v
Specifies to show the version information of the command.

-w *dbadminpw*
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 extension is only applicable for DMS cooked table space. The default value for the extension size is 8192 pages.

-?
Specifies to show the syntax format.

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
```

The idscfgdb command creates a DMS table space. 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.

Note: The default minimum disk space that is required for a DMS database is 1 GB. If you have limited disk space and do not plan to have a large directory, configure an SMS database. An SMS database requires a minimum of 150 MB of disk space. These requirements are for an empty database. When you store data in the database, more disk space might be required.

Example 2:
To create a DMS cooked table space of specific size, run the following command:

```
idscfgdb -I instance_name -a db_admin_id -t db_name –w db_admin_pw
-n -l db_location -u usr_ts_loc –U 195 –z 16
```

In this example, the container size that is specified for USERSPACE1 table space is 195 pages and the extension size is 16 pages.

Example 3:
To create SMS table space for directory server data, run the following command:

```
idscfgdb -I instance_name -a db_admin_id -t db_name -w db_admin_pw
-n -l db_location –m SMS
```

Example 4:
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 5:

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 153.

-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 /home/mydir/myschema.oc schema file for a directory server instance, run the following command:

```
idscfgsch -I instance_name –s /home/mydir/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 153.

-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
```

`idsdbback`, `dbback`

Use the **dbback** 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 | dbback -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 153.

-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.
- When you use the **idsdbmaint** command for table space conversion, the **-k** parameter is required.
- The **idsdbmaint** command can be run only by a root user on AIX, Linux, and Solaris systems.

Synopsis

```
idsdbmaint [-I instance_name [-d debuglevel] [-b outputfile]
          [ -t ts_type [ -k working_dir | -l container | -u container ] ] |
          [ -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.

-k *working_dir*
Specifies the directory to use to export and import data from and to the table space.

-l *container*
The container for LDAPSPACE. If not specified, by default LDAPSPACE is created in the *user_home*/ldap32kcont_*database_name* directory. If you provide a directory for the container, the directory must exist. The DB2 instance owner and the primary group of the instance owner must contain read, write, and execute permission on the directory that is assigned as the container.

-r
Specifies to run row compression on the database that is associated with the directory server instance.

-t *ts_type*
Specifies to run table space conversion on the database that is associated with the directory server instance. The valid value for table space type is DMS or SMS.

-u *container*
The container for USERSPACE1. If not specified, by default USPACE is created in the *user_home*/*database_name*/node/SQL00001 directory. If you provide a directory for the container, the directory must exist. The DB2 instance owner and the primary group of the instance owner must contain read, write, and execute permission on the directory that is assigned as the container.

-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.

Example 3:
To convert table spaces from system managed table space (SMS) to database managed table space (DMS) and use the directory to store the exported data, run the **idsdbmaint** command.

```
idsdbmaint –I instance_name -t DMS -k /disk/data
```

Example 4:

To specify a file container for LDAPSPACE table spaces when you convert it from SMS to DMS and to store the exported data in a directory, run the **idsdbmaint** command.

```
idsdbmaint –I instance_name -t DMS -l /disk/32K_ldapspace_container/ldapspace \
-k /disk/data
```

Example 5:

To specify a file container for USERSPACE1 table spaces when you convert it from SMS to DMS and to store the exported data in a directory, run the **idsdbmaint** command.

```
isdbmaint –I instance_name -t DMS -u /disk/container/userspace1 –k /disk/data
```

Example 6:

To specify a file container for LDAPSPACE and USERSPACE1 table spaces when you convert it from SMS to DMS and to store the exported data in a directory, run the **idsdbmaint** command.

```
idsdbmaint –I instance_name -t DMS -l /disk/32K_ldapspace_container/ldapspace \
-u /disk/container/userspace1 -k /disk/data
```

Example 7:

To convert table spaces from DMS to SMS and to use a directory to store the exported data, run the **idsdbmaint** command.

```
idsdbmaint –I instance_name -t SMS -k /disk/data
```

Example 8:

To specify a container path for LDAPSPACE table spaces when you convert it from DMS to SMS and to use a directory to store the exported data, run the **idsdbmaint** command.

```
idsdbmaint –I instance_name -t SMS -l /disk/32K_ldapspace_container/ -k /disk/data
```

Example 9:

To specify a container path for USERSPACE1 table spaces when you convert it from DMS to SMS and to use a directory to store the exported data, run the **idsdbmaint** command.

```
idsdbmaint –I instance_name -t SMS -u /disk/userspace1_container/ –k /disk/data
```

Example 10:

To specify a file container for LDAPSPACE and USERSPACE1 table spaces when you convert it from DMS to SMS and to use a directory to store the exported data, run the **idsdbmaint** command.

```
idsdbmaint –I instance_name -t SMS -l /disk/32K_ldapspace_container/ \
-u /disk/userspace1_container/ -k /disk/data
```

idsdbmigr

Use the **idsdbmigr** command to migrate the database instance for an existing directory server instance.

Description

The **idsdbmigr** command does migration of the DB2 database instance for an existing directory server instance. By using this command, user data can be migrated from DB2 version 8 while successfully converting it to a fully functioning DB2 9 instance and database. Here, DB2 version can be DB2, version 9.5 or DB2, version 9.7. You can use this command to migrate user data from DB2 version 9.1 while successfully converting it to a fully functioning DB2 9.7 instance and database.

Note:

- The version of the directory server instance must be IBM Security Directory Server, version 6.1 or later.
- The directory server instance name must be specified by using the **-I** parameter, which is a required parameter.
- If you drop an existing directory server instance by using the **idsidrop** command after you migrate a database instance, the temporary files that are created by DB2 during migration are not deleted.

Synopsis

```
idsdbmigr [-I instance_name [-N db2_install_location]]|-h | -?
```

Options

-I *instance_name*
Specifies the directory server instance name.

-N *db2_install_location*
Specifies the DB2 installation location for the migration and postmigration tasks.

-h | -?
Specifies to show the syntax help.

Examples

Example 1:
To run only premigration tasks on DB2, version 8 database, run the following command:

```
idsdbmigr -I instance name
```

Example 2:
To run a complete migration from DB2 version 8 database to a DB2 version 9 database, run the **idsdbmigr** command.

Note: The **-N** parameter specifies the DB2 version 9 installation location.

On Windows systems

```
idsdbmigr -I instance name -N "C:\Program Files\IBM\SQLLIB"
```

On AIX, Linux, and Solaris systems

```
idsdbmigr -I instance name -N /opt/IBM/db2/V9.7
```

idsdbrestore, dbrestore

Use the **dbrestore** command to restore a database and configuration files for a directory server instance.

Description

The **dbrestore** 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 **dbrestore** 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 | dbrestore -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 153.

-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
```

idsdb2ldif, db2ldif

Use the **db2ldif** command to output directory server entries to an LDIF file.

Description

The **db2ldif** 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 Appendix A, "Synchronizing two-way cryptography between server instances," on page 155.

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 Appendix C, "ASCII characters from 33 to 126," on page 159. 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 | db2ldif [-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 **db2ldif** 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 **db2ldif** 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
```

idsdiradm, 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

```
idsdiradm | ibmdiradm [-I instance_name [-f config_file] [-h debug_level] [-t]
          [[ [-p port] [-s secure_port] [-c]] | -k | -i | -u] ] | -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 153.

-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.

The following parameters are applicable only on Windows systems.

-i
Specifies to install the administration server instance as a service.

-u
Specifies to remove the administration server instance as a service.

The following parameters are applicable only on AIX, Linux, and Solaris systems.

-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:

```
idsdiradm -I instance_name
```

On Windows system, you can also start the administration server that is associated with an instance with the following steps:

1. Open **Start** > **Control Panel** > **Administrative Tools** > **Services**.
2. On the **Services** windows, right-click **IBM Security Directory Admin Server V6.3.1 -** ***instance_name***.
3. Click **Start**.

Example 2:
To stop the administration server that is associated with an instance, run the following command:

```
idsdiradm -I instance_name -k
```

On Windows system, you can stop the administration server that is associated with an instance with the following steps:

1. Open **Start** > **Control Panel** > **Administrative Tools** > **Services**.
2. On the **Services** windows, right-click **IBM Security Directory Admin Server V6.3.1 -** ***instance_name***.
3. Click **Stop**.

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 only when the directory server instance is in stopped state. 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]] | -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 153.

-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.

-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.

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 is 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 Appendix C, “ASCII characters from 33 to 126,” on page 159.

For example, on AIX, if you use the ` character for the salt value by using the **-s** parameter, you must precede the ` character with the \ character.

On AIX, Linux, and Solaris systems, after you run the **idsgendirksf** command, the ownership of the `ibmslapddir.ksf` file is `root:system`. You must change the ownership of this file to *directory_server_instance owner*:*instance_owner_group*.

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 is generated with values from 1 to 65535. For more information about debug levels, see Chapter 4, “Debugging levels,” on page 153.

-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, seeAppendix C, “ASCII characters from 33 to 126,” on page 159. 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, myinst, run the following command. For example:

```
idsgendirksf -e mysecretseed –s mysecretsaltvalue -l /home/mydir/tmp
```

After you generate the key stash file, copy the ibmslapddir.ksf file to the idsslapd-*myinst*/etc directory.

idsicrt

Use the **idsicrt** command to create a directory server instance.

Description

The **idsicrt** command can be run only by root on AIX, Linux, or Solaris systems, or a member of the Administrator group on Windows systems. The administrator specifies a directory server instance name and optionally can specify the port, secure port, admin server port, admin server secure port. If these ports are not specified, then the first available port from #389 to #636 is selected for directory server and the secure port. Where, # takes values from 1 to 65. For admin server, ports that are in the range 3538 - 65535 are selected. It is not a must to specify the **-e** parameter. However, the encryption seed is required and you are prompted to supply a value for the encryption seed. On Windows, the administrator must specify the location to store the directory server instance. On AIX, Linux, or Solaris systems, specifying the location is optional.

If an operating system user corresponding to an instance does not exist, then the **idsicrt** command creates the user by internally issuing the **idsadduser** command. To create a user, you must provide the primary group name to associate with the user by using the **-G** parameter. The values for **-u**, **-w**, and **-g** parameters of **idsadduser** is taken from the values of **-I**, **-w**, and **-G** parameters of **idsicrt**.

If an operating system user exists, and the parameter values are specified then you can run **idsicrt** in prompt mode or no prompt mode. In no prompt mode, the properties of the existing user are overwritten.

Note: On Windows 2008 Longhorn system, if DB2 is installed with operating system security set for DB2 objects then the default security groups DB2ADMNS and DB2GROUPS are created. In such a case, if an instance is created by using **idsicrt**, then the instance owner must be a member of the DB2 security groups.

If the **idsicrt** command is used with the **-w** parameter, then the instance owner is added as a member of the DB2 security groups. If the **-w** parameter is not used, then you must manually add the instance owner as a member of the DB2 security groups.

By default, the DB2 database instance name (DB database instance owner) is assumed to have the same name as the directory server instance name. The DB2 instance name can be overwritten by using the **-t** parameter, if a DB2 instance owner ID exists on the operating system.

If a DB2 database instance exists on a system, then that DB2 instance is used. However, if the DB2 database instance is being used by another directory server instance, then the command fails. To verify whether the DB2 instance name is in use, check the directory server instance repository and then check configuration file of each directory server instance.

By default, the directory server instance listens on all available IP addresses.

Note: No database instance is created if the server component of IBM Security Directory Server is not installed.

Attention: When you create a directory server instance, be aware of the information that follows. If you want to use replication, you must synchronize the encryption keys of the server instances to obtain the best performance.

If you are creating a directory server instance that must be cryptographically synchronized with an existing instance, you must synchronize the encryption keys of the instances. You must synchronize before you do any of the following steps because the directory server instance generates the server encryption keys.

- Start the second server instance.
- Run the **idsbulkload** command from the second server instance.
- Run the **idsldif2db** command from the second server instance.

For more information about synchronizing directory server instances, seeAppendix A, "Synchronizing two-way cryptography between server instances," on page 155.

Synopsis

```
idsicrt [-I instance_name [-e encrypt_seed] [-g encrypt_salt] [-p port] [-s secureport]
        [-a admin_port] [-c admin_secureport] [-t db_instance] [-C]
        [-i ipaddress] [-l inst_location] [-r description]
        [-d debug_level] [-b output_file] [-G group_name]
        [-w user_password] [-q] [-n] [-x]] | -v | -?
```

Options

The **idsicrt** command takes the following parameters.

-a *admin_port*
Specifies the port that the administration server associated with a 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 used by other applications, operating systems. The port must not be in use by other directory server instance that is bound to a particular 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 *admin_secureport*
Specifies the secure port that the administration server associated with a 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 used by other applications, operating systems. The port must not be in use by other directory server instance that is bound to a particular host name or IP address.

-C
Specifies to configure a database instance for an existing directory server instance.

-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 153.

-e *encrypt_seed*

Specifies the encryption seed to use for creating the key stash files for a directory server instance. This parameter is required if you use the **-n** parameter. If this parameter is not specified, you are prompted for an encryption seed. 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 the characters that can be used, seeAppendix C, "ASCII characters from 33 to 126," on page 159.

-g *encrypt_salt*

Specifies the encryption salt value. If you want to use replication, use a distributed directory, or import and export LDIF data between server instances, providing an encryption salt value is useful. You can obtain better server performance if two interacting directory server instances have the same encryption salt value.

The encryption salt value must have exactly 12 characters and can contain only printable ISO-8859-1 ASCII characters in the range from 33 to 126. For more information about the characters that can be used, seeAppendix C, "ASCII characters from 33 to 126," on page 159.

If you do not specify an encryption salt, the **idsicrt** command randomly generates a value.

-G *group_name*

Specifies the name of primary group of the user. This parameter is valid only on AIX, Linux, and Solaris systems and is required on these systems if you want to create user.

-i *ipaddress*

Specifies the IP address of the system 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 *instancename*

Specifies the directory server instance name to create. The instance name must be an existing user ID on the system and must not be greater than eight characters in length.

-l *instancelocation*

Specifies the location to store the configuration files and logs of a directory server instance. On Windows systems, this parameter is required and a drive letter must be specified. This location must have a minimum of 30 MB of free space. More disk space must be available to accommodate growth as the directory server log files increase.

-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 used by other applications, operating systems. The port must not be in use by other directory server instance that is bound to a particular 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.

-r *description*
Specifies a description of the directory server instance.

-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 used by other applications, operating systems. The port must not be in use by other directory server instance that is bound to a particular host name or IP address.

-t *db2instance*
Specifies the DB2 database instance name. The database instance name is also the DB2 instance owner ID. By default, the database instance name is assumed to be the same as the directory server instance owner ID.

-v
Specifies to show the version information of the command.

-w
Specifies the password of the user. This parameter is required if you want to create the user.

-x
Creates a proxy directory server instance. If this parameter is not specified, then a full directory server instance with a DB2 instance is created.

-?
Specifies to show the syntax format.

Examples

Example 1:

To create a directory server instance, with the following details run the **idsicrt** command.

- Instance name: myinst
- Port: 389
- Secure port: 636
- Encryption seed: mysecretkey!
- Encryption salt: mysecretsalt
- DB2 instance: myinst

```
idsicrt -I myinst –p 389 –s 636 –e mysecretkey! -g mysecretsalt
```

If the directory server instance exists, then this command fails. If you do not specify the encryption salt, the command generates an encryption salt. If you do not specify the encryption seed, the command prompts for the seed. The encryption seed is not shown on the command line when you enter it. After you type the encryption seed and press Enter, the command attempts to create the directory server instance.

```
idsicrt -I myinst –p 389 –s 636
```

The command prompts for the following response:

```
Enter encryption seed:
```

Example 2:

To create an instance that binds to a particular IP address, run the following command:

```
idsicrt –I myinst –p 389 –s 636 –e mysecretkey! -g mysecretsalt –i 1.9.86.566
```

Example 3:

To create a directory server instance with the following details, run the **idsicrt** command. In this example, the command randomly generates an encryption salt value.

- Instance name: `myinst`
- Port: `389`
- Secure port: `636`
- Encryption seed: `mysecretkey!`
- DB2 instance: `mydbin`

```
idsicrt -I myinst –p 389 –s 636 –e mysecretkey! -t mydbin
```

Example 4:

To create an instance when the corresponding operating system user does not exist, run the following command:

```
idsicrt –I instance_name –e encryptionseed –l instlocation \
 –G group_name –w password
```

idsideploy

Use the **idsideploy** command to create a copy of an existing directory server instance.

Description

You can use the **idsideploy** command to create a directory server instance by using an existing instance on a local or remote computer as a template. When you run **idsideploy**, the configuration settings and schema files from the source instance are duplicated and the directory key stash files are synchronized. The target directory server instance can be configured as a replica or peer of the source instance if it is in an existing replication deployment. You can also configure the target instance as a full directory server instance that is not participating in replication or as proxy server. The following requirements must be met for using the **idsideploy** command:

- The source directory server instance must be running IBM Security Directory Server, version 6.1 or later. You must not use an earlier version of IBM Security Directory Server, and cannot be running another version of LDAP.
- The source directory server instance must be running in normal mode, and it cannot be running in configuration only mode.
- The source directory server instance must be accessible from the computer where you are running the command.
- If you are creating the target instance as a replica or peer, then a replication context must be defined on the source directory server instance. You cannot use the **idsideploy** command to set up the first replica or peer in a replication topology. The source directory server instance must contain at least one replication context, replication group, and replication subentry defined. If you are configuring a replica server, the source instance must contain the initial replication topology, including an agreement to at least one other server. If you are configuring a peer server, the source instance must be defined as a master for one or more subentries in the replication configuration.

- If you are creating the target instance as a replica or peer, a replication subentry is created under the ibm-replicaGroup=default, *replContext* DN. If this DN entry is not present, the instance cannot be duplicated.
- If the operating system user corresponding to the target instance does not exist, the **idsideploy** command creates the user by internally running the **idsadduser** command. However, you must provide the value for primary group name by using the **-G** parameter. The values for **-u**, **-w**, and **-g** parameters of **idsadduser** are taken from values of **-I**, **-a**, and **-G** parameters of **idsideploy**.

The target directory server instance is created on the computer where you run the **idsideploy** command. If the source directory server is on a different computer, the operating systems of the two computers can be different. For example, on a Windows system, you can make a copy of a directory server instance that is running on a Linux system.

The **idsideploy** command also copies the key database files if the source directory server is running in SSL mode. To copy the key database files, the **idsideploy** command must be connected to the source instance over SSL.

If the source instance is a proxy server, then the target instance that gets created is a proxy server. If the source instance is a full directory server, then the target instance that gets created is a full directory server. If the source instance is a full directory server, you can choose whether to copy the data or not to the target instance.

Note: If you want to copy the data from the source instance while you create the target instance, the following requirements must be met:
- The version of DB2 must be the same for both directory server instances. Both instances must use DB2 v8 or DB2 v9. The fix pack levels can be different.
- The source directory server instance must be configured for online backup.
- An initial offline backup of the source instance must be taken before you use the **idsideploy** command to copy the instance. The path that you specify must contain only one backup image.
- The path where the backup image is stored must be accessible to both the source instance and the target instance.

For information about preparing the source instance for copying the data, see the *Installing and Configuring* section of the IBM Security Directory Server documentation.

Synopsis

```
idsideploy [-I instance_name -e encrypt_seed -D admin_DN
           -w admin_Pw -su LDAP_URL -sD admin_DN -sw admin_Pw
           [-l inst_location] [-L directory] [-r peer|replica]
           [-K key_file -N key_name -P key_pw]
           [-d debug_level] [-b output_file] [-G group_name]
           [-a password] [-x] [-q] [-n]] | -v | -?
```

Options

The **idsideploy** command takes the following parameters.

-a *password*
: Specifies the instance owner password. This password is used during the user creation if the user does not exist, and is also used for the database configuration. On AIX, Linux, and Solaris systems, this parameter is required

when the **-G** parameter is specified. On Windows systems, this parameter is required when a new user is created for the target instance.

-b *outputfile*
Specifies the full path of a file in which to redirect 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 also sent to this file.

-d *debuglevel*
Sets the debug level in the LDAP library. Set debug mode when you use the **ldtrc** command.

-D *admin_DN*
Specifies the directory administrator distinguished name (DN) for the target directory server instance.

-e *encrypt_seed*
Specifies the encryption seed for the target directory server instance. This value must match with the value provided for the source directory server instance.

-G
Specifies the name of primary group of the user that is associated with the target instance. This parameter is valid only on AIX, Linux, and Solaris systems and is required on these systems to create the user.

-I *instance_name*
Specifies the name of the directory server instance to create. The instance name must be an existing user ID on the system and must not be greater than eight characters in length.

-l *inst_location*
Specifies the location to store the configuration files and logs of a directory server instance. On Windows systems, this parameter is required and a drive letter must be specified. This location must have a minimum of 30 MB of free space. More disk space must be available to accommodate growth as the directory server log files increase. For a full directory server, a minimum of 80 MB is required to also store DB2 database.

-L *directoryPath*
Specifies the directory path of the backup image of the source instance from where to load data into the target instance. This parameter must be specified with the **-r** and **-p** parameters. The **-L** parameter must not be specified when the **-x** parameter is specified.

-K *keyfile*
Specifies the key file to use for an SSL connection.

-n
Specifies to run in no prompt mode. All output from the command is generated, except for messages that require user interaction.

-N *key_name*
Specifies the private key name to use in the key file for an SSL connection.

-p
Specifies to restore database on the target instance. To use **-p** parameter, the instance that is specified with the **-I** parameter must exist and back up of the source instance must be taken. The **-L** parameter is required with the **-p** 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.

-r peer | replica
Specifies to configure the target instance in a replication environment as a peer or replica. This parameter must not be specified with the **-x** parameter. The only valid values with this parameter are `peer` and `replica`.

-sD *admin_DN*
Specifies the directory administrator DN of the source instance.

-sU *LDAP_URL*
Specifies the LDAP URL of the source instance.

-sw *pw*
Specifies the administrator password of the source instance.

-v
Specifies to show the version information of the command.

-w *password*
Specifies the administrator password for the target instance.

-x
Specifies to create a proxy server instance. The source instance must also be configured as a proxy server. This parameter must not be specified with the **-L**, **-p**, or **-r** parameter.

-?
Specifies to show the syntax format.

Note:

- If **idsideploy** is run with the **-p** parameter to restore a database, then you must set the `DB2INSTANCE` environment variable. The variable must point to the database instance name associated with the directory server instance. Otherwise, **idsideploy** might fail.
- If the source instance is created with raw DMS table space, then backup or restore by using the **idsideploy** command might fail.

Examples

Example 1:
To create a target instance with data from an existing source instance, run the **idsideploy** command of the following format:

```
idsideploy -sU ldap://host:port -sD adminDN -sw adminPWD \
-e encrypt_seed -I inst_name -a user_pwd -D adminDN -w adminPWD \
-l inst_location –b outputfile -q -L directory_path
```

Example 2:
To create a stand-alone target instance without data from an existing source instance, run the following command:

```
idsideploy -sU ldap://host:port -sD adminDN -sw adminPWD \
-e encrypt_seed -I inst_name -a user_pwd -D adminDN -w adminPWD \
-l inst_location –b outputfile
```

This command does not clone the database.

Example 3:

To create a target instance as a peer in an existing replication setup, run the following command:

```
idsideploy -sU ldap://host:port -sD adminDN -sw adminPWD \
-e encrypt_seed -I inst_name -a user_pwd -D adminDN -w adminPWD \
-l inst_location –b outputfile -L directory_path -r peer
```

Example 4:

To deploy a proxy instance in SSL mode, run the following command:

```
idsideploy -sU ldaps://host:sec_port -sD adminDN -sw adminPWD \
-e encrypt_seed -I inst_name -K kdb_file -P kdb_file_pwd \
-N certificate_name -D adminDN -w adminPWD -x -l inst_location
```

Example 5:

To create a target instance when the corresponding operating system user does not exist, run the following command:

```
idsideploy -I instance_name -a inst_owner_PWD -D adminDN \
-w adminPWD -e encryption_seed -l inst_location –G group_name \
-sU ldap_URL -sD adminDN -sw adminPWD -L directoryPath
```

idsidrop

Use the **idsidrop** command to drop a directory server instance.

Description

The **idsidrop** command can be run only by root on UNIX based systems, or a member of the Administrators group on Windows systems. The administrator specifies a directory server instance name and optionally can specify whether to delete the database instance. The command does not delete the directory server instance owner. The command does not delete the directory server instance until the directory server instance is stopped.

Synopsis

```
idsidrop [-I instancename [-r] [-R] [-s] [-d debuglevel] [-b outputfile]
        [-q] [-n]] | -v | -?
```

Options

The **idsidrop** command takes the following parameters.

-b *outputfile*
: Specifies the full path of a file in which to redirect 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 also 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 153.

-I *instancename*
: Specifies the directory server instance name to drop. This parameter is required when there are directory server instances on the local 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.

-r
Specifies to delete the database instance that is associated with the directory server instance. It also deletes all the databases that are contained in the database instance.

-R
Specifies to unconfigure the database instance and to retain the directory server instance.

-s
Removes the backup for the directory server instance if configured.

-v
Specifies to show the version information of the command.

-?
Specifies to show the syntax format.

Examples

Example 1:
To remove a directory server instance and to retain the associated database instance, run the following command:

```
idsidrop -I instancename
```

Example 2:
To remove a directory server instance and delete the associated database instance, run the following command:

```
idsidrop -I instancename -r
```

Note: If you run the **idsidrop** command with the **-r** parameter against a proxy server instance, then this parameter is ignored.

Example 3:
To unconfigure the database instance without removing the directory server instance, run the following command:

```
idsidrop -I instancename -R
```

Example 4:
To drop a directory server instance and remove the backup, run the following command:

```
idsidrop -I instancename -r -s -n
```

idsilist

Use the **idsilist** command to list directory server instances on the system.

Description

The **idsilist** command can be run only by root on UNIX systems or a member of the Administrators group on Windows systems. Based on the parameter that is used, the command lists a directory server instance or all directory server instances that exist on the system. The command retrieves detailed information about each instance on a system.

Note: You can manually change the permissions on the directory instance repository files to allow the command to be run by other users. Users with permissions to read the `ibmslapd.conf` file of all directory server instances on a system can run the command successfully.

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 153.

-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, `myinst1`, run the following command:

```
idsilist -I myinst1
```

The command generates the following output:

```
Directory server instance(s):
myinst1
```

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 myinst1 -a** or **idsilist -I myinst1 -r**.

Example 2:

To list complete details about instances, run the **idsilist** command along with the **-I** and **-a** parameters. For example:

```
idsilist -I myinst1 -a

Directory server instance(s):

--------------------------------------
Name: myinst1
Version: 6.3.1
Location: /home/myinst1
Description: IBM Security Directory Server Instance V6.3.1
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 myinst1 -r

Directory server instance(s):
myinst1#6.3.1#/home/myinst1# IBM Security Directory Server Instance V6.3.1#
All available #4389#4636#3544#3545#Directory Server
```

Example 4:

To list all directory server instances on a system, run the following command:

```
idsilist
```

The command generates the following output:

```
Directory server instance(s):
myinst1
myinst2
```

Example 5:

To obtain information about each instance, run the command with the **-a** or **-r** parameter.

```
idsilist -a
```

This command lists the directory server instances with their versions:

```
Directory server instance(s):
-----------------------------
Instance 1:

Name: myinst1
Version: 6.3.1
Location: /home/myinst1
Description: IBM Security Directory Server Instance V6.3.1
IP Addresses: All available
Port: 389
Secure Port: 636
admin server Port: 3538
admin server Secure Port: 3539
```

```
Type: Directory Server

Instance 2:

Name: myinst2
Version: 6.3
Location: /home/myinst2
Description: IBM Security Directory Server Instance V6.3
IP Addresses: All available
Port: 389
Secure Port: 636
admin server Port: 3538
admin server Secure Port: 3539
Type: Proxy Server
```

```
idsilist -r
```

The command generates the following output:

```
Directory server instance(s):
myinst1#6.3.1#/home/myinst1#IBM Security Directory Server Instance V6.3.1#
All available#389#636#3538#3539#Directory Server
myinst2#6.3#/home/myinst2#IBM Security Directory Server Instance V6.3#
All available#389#636#3538#3539#Proxy Server
```

Note:

1. The directory server types are Proxy Server, Directory Server, or Unknown. If a description is not set for a directory server instance, it is not shown.
2. The IP address `All available` indicates that the directory server instance binds to all available IP addresses. If the directory server instance binds only to certain IP addresses, a list is presented, separated by comma. For example,

   ```
   IP Addresses: 1.3.45.333,1.2.45.222
   ```

idsimigr

Use the **idsimigr** command to migrate schema and configuration files from an earlier version of IBM Security Directory Server to the latest general availability (GA) level.

Description

The **idsimigr** command is a migration utility. This command migrates the schema and configuration files from an earlier release to IBM Security Directory Server 6.3.1 versions of these files. After you migrate the schema and configuration files, the command creates a directory server instance with the migrated information. The created directory server instance is the upgraded version of your previous server. You can also use Instance Administration Tool to migrate from a previous version. For more information about Instance Administration Tool, see *Creating and administering instances* in the *Installation and Configuration* section of the IBM Security Directory Server documentation.

Synopsis

The syntax for the **idsimigr** command.

```
idsimigr [–I instancename] [-u backupdir] [-p port]
        [-s secure_port] [-a adm_port] [-c adm_secureport] [-t dbinstance]
        [-i ipaddress] [-l inst_location] [-r description] [-G group_name]
        [–w password] [-d debuglevel] [-b outputfile] [–q] [-n] | [-v]
        | [-?]
```

Options

The **idsimigr** command takes the following parameters.

-a *adm_port*
Specifies the port on which the administration server for the directory server instance listens on.

Note: If you have two or more instances that listen on an IP address, be sure that those instances do not use the same port.

-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 *adm_secureport*
Specifies the secure port on which the administration server for the directory server instance listens on. Specify a positive number that is greater than 0 and lesser than or equal to 65535. The port that is specified must a conflict with ports in use by other instances that are 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 153.

-G *group_name*
Specifies the operating system group name for the user. This parameter is required when you create an operating system user on a system. This parameter is valid only on AIX, Linux, and Solaris systems.

-i *ipaddress*
Specifies the IP address to which the directory server instance binds to. If more than one IP address is specified, the IP addresses must be separated by a comma with no spaces. Spaces can be used only if the entire argument is enclosed in quotation marks ("). If you want to use all available IP addresses, use the key word `all`. If you do not specify the **-i** parameter, all available IP addresses is the default setting.

-I *instancename*
Specifies the directory server instance name to be create or migrate. The instance name must be an existing user ID on the system and must not be greater than eight characters in length. If there is no corresponding user ID for the directory server instance name, the command fails. For more information about requirements for the instance name, see *Setting up users and groups: directory server instance owner, database instance owner, and database owner* in the *Installing and Configuring* section of the IBM Security Directory Server documentation.

-l *inst_location*
Specifies the location in which to store the configuration files and logs for the directory server instance. On Windows systems, you must specify this

parameter and a drive letter must be provided. The location must have a minimum of 30 MB of free disk space. More disk space must be available to accommodate growth as directory server log files increase in size.

-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 on which the directory server instance listens on. Specify a positive number that is greater than 0 and lesser than or equal to 65535. The port that is specified must not conflict with ports in use by other instances that are 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.

-r *description*
Specifies a description of the directory server instance.

-s *secure_port*
Specifies the secure port that the directory server instance listens on. Specify a positive number that is greater than 0 and lesser than or equal to 65535. The port that is specified must not conflict with ports in use by other instances that are bound to a host name or IP address.

-t *dbinstance*
Specifies the DB2 database instance name. By default, the database instance name is assumed to be the same as the directory server instance owner ID.

-u *backupdir*
Specifies the directory in which the schema and configuration files to migrate is saved.

If all the necessary files are not found in the directory, the command fails. These files include the server configuration files and the schema files: `V3.ibm.at`, `V3.ibm.oc`, `V3.system.at`, `V3.system.oc`, `V3.user.at`, `V3.user.oc`, and `V3.modifiedschema`.

-v
Specifies to show the version information of the command.

-w *password*
Specifies the password of the user. To create a user, you must also specify this parameter.

-?
Specifies to show the syntax format.

Examples

Example 1:
To upgrade an IBM Security Directory Server, version 6.2 instance, `myinst`, to IBM Security Directory Server 6.3, issue the following command:

```
idsimigr –I myinst -n
```

ldif

Use the **ldif** command to convert arbitrary data values to LDAP data interchange format (LDIF).

Description

The **ldif** command is a shell-accessible tool that converts arbitrary data values to LDIF. The command reads values from standard input and produces entries appropriate for use in an LDIF file.

Synopsis

```
ldif [-b ] attrname
```

Options

The **ldif** command takes the following parameters. All values are case-sensitive.

-b
Specifies the value is a single raw binary value. Output is a base64 encoded value.

attrname
Specifies the attribute name for which values to convert. If the **-b** parameter is not specified, the command interprets each line of standard input to be a separate value of the attribute.

Examples

Example 1:
To create the LDIF format for the sn attribute with smith as value, run the following command:

```
ldif sn
smith
```

The command generated the following output:

```
sn: smith
```

Example 2:
To create binary value with the **ldif** command, use the **-b** parameter. Run the following command to generate the binary value:

```
ldif -b sn
smith
```

On Windows systems, press Ctrl+Z, and on UNIX based systems, press Ctrl+D to generate the following output:

```
sn:: c21pdGgNCg==
```

idsldif2db, ldif2db

Use the **ldif2db** command to load entries from an LDIF file to a database.

Description

You can run the **ldif2db** 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 Appendix A, “Synchronizing two-way cryptography between server instances,” on page 155.

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 | ldif2db [-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 153.

-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:
On AIX, Linux, or Solaris systems, to load the *IDS_LDAP_HOME*/examples/sample.ldif file that is included with IBM Security Directory Server, run the following command:

```
idsldif2db -i IDS_LDAP_HOME/examples/sample.ldif
```

The *IDS_LDAP_HOME* variable contains the path of the IBM Security Directory Server installation location.

Example 2:
On Windows systems, to load the *IDS_LDAP_HOME*\examples\sample.ldif, run the following command:

```
idsldif2db -i IDS_LDAP_HOME\examples\sample.ldif
```

idslogmgmt

Use the **idslogmgmt** command to limit the log file size.

Description

An administrator can use the **idslogmgmt** command to limit the size of log files. The **idslogmgmt** command activates every 15 minutes, checks the log files sizes, and moves log files that exceed the maximum log size threshold to an archive file. The number of archived logs can also be limited. The configuration settings are in the ibmslapd.conf file in most cases, except for administrative tools and the idslogmgmt log settings. You can also configure the log management settings by using Web Administration Tool. To use the **idslogmgmt** command, IBM Security Directory Integrator must be installed.

You must run the **idslogmgmt** command by using a system startup script or manually run the command. To run the command, type the following command at a command prompt.

```
idslogmgmt
```

To specify the settings for the administrative tools log, idsadm.log, you can set the following environment variables.

- To set the threshold size, use the *IDSADM_SIZE_THRESHOLD* variable. The default threshold size is 10 MB. For example: *IDSADM_SIZE_THRESHOLD*=10.
- To set the maximum number of archive files, use the *IDSADM_ARCHIVES* variable. The default value for the maximum number of archive files is 3. For example: *IDSADM_ARCHIVES*=3.

The archived log files are in the following directories and have the file name *timestamp*_idsadm.log.

- UNIX path: /var/idsldap/V6.3.1
- Windows path: *DS_install_directory*\var

To specify the settings for the **idslogmgmt** tool log, idslogmgmt.log, you can set the following environment variables.

- To set the threshold size, use the *IDSLMG_SIZE_THRESHOLD* variable. The default threshold size is 10 MB. For example: *IDSLMG_SIZE_THRESHOLD*=10.
- To set the maximum number of archive files, use the *IDSLMG_ARCHIVES* variable. The default value for the maximum number of archive files is 3. For example: *IDSLMG_ARCHIVES*=3.

The archived log files are in the following directories and have the file name *timestamp*_idslogmgmt.log.

- UNIX path: /var/idsldap/V6.3.1
- Windows path: *DS_install_directory*\var

In addition to the main log file, idslogmgmt.log, the other log file that is produced by the IBM Security Directory Integrator tool is ibmdi.log. If the directories are not created, then the additional logs are placed in the current working directory. The ibmdi.log file is overwritten each time the **idslogmgmt** command is run. As a result, the size of this log file remains small.

Synopsis

The syntax for the **idslogmgmt** command.

```
idslogmgmt [–I instancename [-k]] [-t threshold_size
-a archives -p archive_path] [-?]
```

Options

The **idslogmgmt** command takes the following parameters.

-a *archives*
: Specifies the maximum number of archived log files for IBM Security Directory Server Instance Administration Tool.

-I *instance_name*
: Specifies the instance name for which the command manages the logs. If you specify the **-I** parameter, then the **-t**, **-a**, and **-p** parameters must not be specified with it.

-k
: Stops the Log Management feature for the directory server instance. You must use the **-k** parameter with the **-I** parameter.

-p *archive_path*
: Specifies the path where the archived log files of IBM Security Directory Server Instance Administration Tool is stored.

-t *threshold_size*
: Specifies the threshold size for the log file of IBM Security Directory Server Instance Administration Tool to trigger archiving.

-?
: Specifies to show the help for the command.

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
```

IDSProgRunner

The **IDSProgRunner** command is called internally by the **idsxinst** and **idsxcfg** commands to run long tasks in the background.

The **IDSProgRunner** command is called from the **idsxinst** and **idsxcfg** commands to create a long-running task to run in the background. The **idsxinst** command then exits, and other processes that include other instances of **idsxcfg** query the state and progress of the task during and after its running.

The **IDSProgRunner** command is used instead of creating the task directly for two reasons:

- The **IDSProgRunner** command obtains the exit code of the process that is running. The only way to get the exit code from a process is for another process, **IDSProgRunner**, to be waiting for it at the time the task exits.
- **IDSProgRunner** creates almost any process to run in the background. It also maintains the start and stop time, and the PID of the process so that the task can be signaled or ended.

idsrunstats, runstats

Use the **runstats** 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 | runstats [-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 153.

-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

The **idssethost** command can be run only by a root user on UNIX or a member of the Administrators group on Windows systems by default. You can manually change the permissions on the instance repository files to allow other users to run the command. Users with the access to read the `ibmslapd.conf` files of all directory server instances on the system can run the command.

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.

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 153.

-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

The **idssetport** command can be run only by root on AIX, Linux, Solaris systems, or a member of the Administrators group on Windows systems by default. You can manually change the permissions on the instance repository files to allow other users to run the command. Users with the access to read the `ibmslapd.conf` files of all directory server instances on the system can run the command.

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 153.

-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, Solaris, and HP-UX (Itanium) 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.

idsslapd, ibmslapd

Use the **idsslapd** or **ibmslapd** command to start or stop the directory server process.

Description

The **ibmslapd** command changes the working directory to *instance_home*/idsslapd-*instance*/workdir. Therefore, relative paths are considered as relative to *instance_home*/idsslapd-*instance*/workdir.

Synopsis

```
idsslapd | ibmslapd [-I instancename [-f configfile] [-h debuglevel] [-t]
[[ [-p port] [-s secureport] [-R ServerID] [-c] [-a | -n] ]
| -k | -i | -u] ] | -v | -? | -h ?
```

Options

The **idsslapd** 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 153.

-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.

The following parameters are applicable only on Windows systems.

-i
Specifies to install the directory server instance as a service.

-u
Specifies to remove the directory server instance as a service.

The following parameters are applicable only on AIX, Linux, and Solaris systems.

-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:

```
idsslapd -I myinst
```

Example 2:
To stop the directory server process for the instance, `myinst`, run the following command:

```
idsslapd -I myinst -k
```

idssnmp

Use the **idssnmp** command to start the `idssnmp` process.

Description

You must install a supported version of IBM Security Directory Integrator to use the **idssnmp** command. The **idssnmp** command starts the `idssnmp` process.

Options

The **idssnmp** command takes the following parameter.

-q
Specifies not to show the log messages on the standard output. This parameter is optional.

-v
Specifies to print the version number of the **idssnmp** command. This parameter is optional.

-?
Specifies to show the command usage. This parameter is optional.

When IBM Security Directory Integrator stops, it returns one of the following exit codes:

0
Start IBM Security Directory Integrator with **-v** parameter (show information and exit).

1

- Cannot open log file (**-l** parameter)

- Cannot open configuration file
- Stopped by admin request

2 Exit after auto run. When you start IBM Security Directory Integrator by specifying the **-w** parameter, the directory integrator runs the AssemblyLine specified by the **-r** parameter and then exits.

9 License is expired or invalid.

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 | -?
```

Options

The **idsucfgchglg** 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 information about debug levels, see Chapter 4, “Debugging levels,” on page 153.

-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.

-v
Specifies to show the version information of the command.

-?
Specifies to show the syntax format.

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 | -?
```

Options

The **idsucfgdb** 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 153.

-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.

-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.

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 153.

-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 `/home/mydir/myschema.oc` schema file from the `ibmslapd.conf` file of an instance, run the following command:

```
idsucfgsch –s /home/mydir/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 153.

-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`

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. You can specify the name of the destination file where you want to dump the trace. The default location fro the files is listed.

For AIX, Linux, and Solaris systems
`/var/ldap/ibmslapd.trace.dump`

For Windows systems:
`installation_path``\var\ibmslapd.trace.dump`

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.

```
C:\>ldtrc info
Trace Version:1.00
Op. System:NT
Op. Sys. Version:4.0
H/W Platform:80x86

Mask: *.*.*.*.*.*
pid.tid to trace: all
cpidto 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
```

idsrun

The **idsrun** command is used internally by commands to start another process.

The **idsrun** command is used on AIX, Linux, and Solaris systems. It is similar to **IDSProgRunner**, but it does not track the process that it creates. Instead, it starts the executable and then exits. This program is used by the **idsdiradm** command to start a directory server, and is used by the **idsicrt** command to start **idsdiradm**.

idsxcfg

Use the **idsxcfg** command to start Configuration Tool for a directory server instance.

Description

The **idsxcfg** command starts IBM Security Directory Server Configuration Tool for a directory server instance. You can use Configuration Tool to configure a directory server instance.

Synopsis

```
idsxcfg [-I instanceName] | -?
```

Options

The **idsxcfg** command takes the following parameters.

-I *instanceName*
Specifies the directory server instance name to configure.

-?
Specifies to show the syntax help.

idsxinst

Use the **idsxinst** command to start Instance Administration Tool.

Description

The **idsxinst** command starts IBM Security Directory Server Instance Administration Tool. You can use Instance Administration Tool to manage directory server instances on the system.

Synopsis

```
idsxinst [-migrate backupdir] | -?
```

Options

The **idsxinst** command takes the following parameters.

-migrate *backupdir*
Specifies the path of the stored schema and configuration files for migration.

-? Specifies to show the syntax help.

migbkup

Use the **migbkup** command to a backup of the schema files, configuration files, key stash files, and key database files of a directory server instance.

Description

You can run the **migbkup** command to back up the schema files, configuration files, key stash files, and key database files of a directory server instance. This command does not back up the data in the directory database. The following files are backed up by the **migbkup** command from the *instance_home*/etc directory:

- ibmslapd.conf
- V3.ibm.at
- V3.ibm.oc
- V3.system.at
- V3.system.oc
- V3.user.at
- V3.user.oc
- V3.modifiedschema
- V3.config.at
- V3.config.oc
- V3.ldapsyntaxes
- V3.matchingrules

- `ibmslapdcfg.ksf`
- `ibmslapddir.ksf`
- `ibmdiradmService.cmd` (On Windows only)
- `ibmslapdService.cmd` (On Windows only)

The **migbkup** command creates the following file.

- `db2info`

Synopsis

```
migbkup instance_home backup_directory | -?
```

Options

The **migbkup** command takes the following parameters.

instance_home
Specifies the home directory of the directory server instance. For example, *user_home_dir*/idsslapd-*instance_name*. This parameter is required.

backup_directory
Specifies the directory where the files must be copied. This parameter is required.

-?
Specifies to show the syntax help.

Examples

Example 1:
On a Linux system, run the **migbkup** command to take a backup of files:

```
migbkup /home/idsinst/idsslapd-idsinst/ /home/tdsbkup
```

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
```

Appendix A. Synchronizing two-way cryptography between server instances

You must synchronize two-way cryptography between directory server instances to reduce the time that is required to encrypt and decrypt data during server communications.

Before you begin

To synchronize directory server instances by using two-way cryptography, you must have two or more instances.

You must synchronize the servers before you do any of the following operations:

- Starting the second server instance.
- Running the **idsbulkload** command from the second server instance.
- Running the **idsldif2db** command from the second server instance. When you import an LDIF data that is not cryptographically synchronized, AES encrypted entries in the file are not imported.

About this task

If you want to use replication, distributed directory, or import and export LDIF data between server instances, you must cryptographically synchronize the instances for better performance.

Although, in the procedure two server instances are used. You might need a group of server instances that are cryptographically synchronized.

Procedure

To cryptographically synchronize two server instances, assuming that you created the first server instance do the following steps.

1. Create the second server instance, but do not start the server instance.
2. Run the **idsbulkload** command, or run the **idsldif2db** command on the second server instance.
3. Copy the `ibmslapddir.ksf` file (the key stash file) from the first server instance to the second server instance. The file is in the `idsslapd-`*`instance_name`*`\etc` directory on Windows systems, or in the `idsslapd-`*`instance_name`*`/etc` directory on AIX, Linux, and Solaris systems. The *instance_name* is the name of the server instance.
4. Run the **idsgendirksf** command to create the `ibmslapddir.ksf` file from the source server instance.
5. Replace the `ibmslapddir.ksf` file of the target server instance with the `ibmslapddir.ksf` file of the source server instance. For more information about the **idsgendirksf** command, see"**idsgendirksf**" on page 112.
6. Run any one of the following operations:
   - Start the second server instance.
   - Run the **idsbulkload** command from the second server instance.
   - Run the **idsldif2db** command from the second server instance.

Results

After the directory server instances are cryptographically synchronized, AES encrypted data gets loaded correctly.

Related reference:

"`idsgendirksf`" on page 112
Command to regenerate a directory key stash file for a directory server instance.

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/assignments/character-sets .

Table 2. IANA defined character sets

Character	Locale					DB2 code page	
Set Name	HP-UX	Linux, Linux_390,	NT	AIX	Solaris	UNIX	NT
`ISO-8859-1`	X	X	X	X	X	819	1252
`ISO-8859-2`	X	X	X	X	X	912	1250
`ISO-8859-5`	X	X	X	X	X	915	1251
`ISO-8859-6`	X	X	X	X	X	1089	1256
`ISO-8859-7`	X	X	X	X	X	813	1253
`ISO-8859-8`	X	X	X	X	X	916	1255
`ISO-8859-9`	X	X	X	X	X	920	1254
`ISO-8859–15`	X	n/a	X	X	X		
`IBM437`	n/a	n/a	X	n/a	n/a	437	437
`IBM850`	n/a	n/a	X	X	n/a	850	850
`IBM852`	n/a	n/a	X	n/a	n/a	852	852
`IBM857`	n/a	n/a	X	n/a	n/a	857	857
`IBM862`	n/a	n/a	X	n/a	n/a	862	862
`IBM864`	n/a	n/a	X	n/a	n/a	864	864
`IBM866`	n/a	n/a	X	n/a	n/a	866	866
`IBM869`	n/a	n/a	X	n/a	n/a	869	869
`IBM1250`	n/a	n/a	X	n/a	n/a		
`IBM1251`	n/a	n/a	X	n/a	n/a		
`IBM1253`	n/a	n/a	X	n/a	n/a		
`IBM1254`	n/a	n/a	X	n/a	n/a		
`IBM1255`	n/a	n/a	X	n/a	n/a		
`IBM1256`	n/a	n/a	X	n/a	n/a		

Table 2. IANA defined character sets (continued)

Character	Locale					DB2 code page	
Set Name	HP-UX	Linux, Linux_390,	NT	AIX	Solaris	UNIX	NT
`TIS-620`	n/a	n/a	X	X	n/a	874	874
`EUC-JP`	X	X	n/a	X	X	954	n/a
`EUC-KR`	n/a	n/a	n/a	X	X*	970	n/a
`EUC-CN`	n/a	n/a	n/a	X	X	1383	n/a
`EUC-TW`	X	n/a	n/a	X	X	964	n/a
`Shift-JIS`	n/a	X	X	X	X	932	943
`KSC`	n/a	n/a	X	n/a	n/a	n/a	949
`GBK`	n/a	n/a	X	X	n/a	1386	1386
`Big5`	X	n/a	X	X	X	950	950
`GB18030`	n/a	X	X	X	X		
`HP15CN`	X (with non-`GB18030`)						

Appendix C. 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 3. 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
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				

Index

Notices

This information was developed for products and services offered in the U.S.A. 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 give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

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

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law :

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 states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement might 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 Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply 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 Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.

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.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All 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 the names and addresses used by an actual business enterprise 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. 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 IBM's application programming interfaces.

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_. All rights reserved.

If you are viewing this information in softcopy form, the photographs and color illustrations might not be displayed.

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.

Adobe, Acrobat, PostScript and all Adobe-based trademarks are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Printed in USA

SC27-2753-02