IBM Tivoli Directory Server

Command Reference

Version 6.1

SC23-7834-00

IBM Tivoli Directory Server

Command Reference

Version 6.1

SC23-7834-00

Note

Before using this information and the product it supports, read the general information under Appendix E, “Notices,” on page 137.

This edition applies to version 6, release 1, of the IBM Tivoli Directory Server and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

About this book

IBM® Tivoli® Directory Server is the IBM implementation of Lightweight Directory Access Protocol for supported Windows®, AIX®, Linux® (xSeries®, zSeries®, pSeries®, and iSeries™), Solaris, and Hewlett-Packard UNIX® (HP-UX) operating systems.

IBM Tivoli Directory Server version 6.1 Command reference describes the syntax and usage of the command-line utilities included with IBM Tivoli Directory Server.

Intended audience for this book

This book is for administrators of IBM Tivoli Directory Server version 6.1.

Readers need to know how to use the operating system on which IBM Tivoli Directory Server will be installed.

Publications

This section lists publications in the IBM Tivoli Directory Server version 6.1 library and related documents. The section also describes how to access Tivoli publications online and how to order Tivoli publications.

IBM Tivoli Directory Server version 6.1 library

The following documents are available in the IBM Tivoli Directory Server version 6.1 library:

- *IBM Tivoli Directory Server Version 6.1 What's New for This Release*, SC23-6539-00

  Provides information about the new features in the IBM Tivoli Directory Server Version 6.1 release.
- *IBM Tivoli Directory Server Version 6.1 Quick Start Guide*, GI11-8172-00

  Provides help for getting started with IBM Tivoli Directory Server 6.1. Includes a short product description and architecture diagram, as well as a pointer to the product Information Center and installation instructions.
- *IBM Tivoli Directory Server Version 6.1 System Requirements*, SC23-7835-00

  Contains the minimum hardware and software requirements for installing and using IBM Tivoli Directory Server 6.1 and its related software. Also lists the supported versions of corequisite products such as DB2® and GSKit.
- *IBM Tivoli Directory Server Version 6.1 Installation and Configuration Guide*, GC32-1560-00

  Contains complete information for installing, configuring, and uninstalling IBM Tivoli Directory Server. Includes information about upgrading from a previous version of IBM Tivoli Directory Server.
- *IBM Tivoli Directory Server Version 6.1 Administration Guide*, GC32-1564-00

  Contains instructions for performing administrator tasks through the Web Administration Tool and the command line.
- *IBM Tivoli Directory Server Version 6.1 Command Reference*, SC23-7834-00

  Describes the syntax and usage of the command-line utilities included with IBM Tivoli Directory Server.
- *IBM Tivoli Directory Server Version 6.1 Server Plug-ins Reference*, GC32-1565-00

Contains information about writing server plug-ins.

- *IBM Tivoli Directory Server Version 6.1 Programming Reference*, SC23-7836-00

  Contains information about writing Lightweight Directory Access Protocol (LDAP) client applications in C and Java™.
- *IBM Tivoli Directory Server Version 6.1 Performance Tuning and Capacity Planning Guide*, SC23-7836-00

  Contains information about tuning the directory server for better performance. Describes disk requirements and other hardware needs 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 rough rules of thumb.
- *IBM Tivoli Directory Server Version 6.1 Problem Determination Guide*, GC32-1568-00

  Contains information about possible problems and corrective actions that can be tried before contacting IBM Software Support.
- *IBM Tivoli Directory Server Version 6.1 Messages Guide*, GC32-1567-00

  Contains a list of all informational, warning, and error messages associated with IBM Tivoli Directory Server 6.1.
- *IBM Tivoli Directory Server Version 6.1 White Pages*, SC23-7837-00

  Describes the Directory White Pages application, which is provided with IBM Tivoli Directory Server 6.1. Contains information about installing, configuring, and using the application for both administrators and users.

Related publications

The following documents also provide useful information:

- *Java Naming and Directory Interface™ 1.2.1 Specification* on the Sun Microsystems Web site at http://java.sun.com/products/jndi/1.2/javadoc/index.html.

  IBM Tivoli Directory Server Version 6.1 uses the Java Naming and Directory Interface (JNDI) client from Sun Microsystems. See this document for information about the JNDI client.

Accessing terminology online

The *Tivoli Software Glossary* includes definitions for many of the technical terms related to Tivoli software. The *Tivoli Software Glossary* is available at the following Tivoli software library Web site:

http://publib.boulder.ibm.com/tividd/glossary/tivoliglossarymst.htm

The IBM Terminology Web site consolidates the terminology from IBM product libraries in one convenient location. You can access the Terminology Web site at the following Web address:

http://www.ibm.com/software/globalization/terminology

Accessing publications online

IBM posts publications for this and all other Tivoli products, as they become available and whenever they are updated, to the Tivoli Information Center Web site at http://publib.boulder.ibm.com/tividd/td/link/tdprodlist.html.

In the Tivoli Information Center window, click **Tivoli product manuals**. Click the letter that matches the first letter of your product name to access your product library. For example, click **M** to access the IBM Tivoli Monitoring library or click **O** to access the IBM Tivoli OMEGAMON® library.

Note: If you print PDF documents on other than letter-sized paper, set the option in the **File → Print** window that allows Adobe Reader to print letter-sized pages on your local paper.

Ordering publications

You can order many Tivoli publications online at http://www.elink.ibmlink.ibm.com/public/applications/publications/cgibin/pbi.cgi.

You can also order by telephone by calling one of these numbers:

- In the United States: 800-879-2755
- In Canada: 800-426-4968

In other countries, contact your software account representative to order Tivoli publications. To locate the telephone number of your local representative, perform the following steps:

1. Go to http://www.elink.ibmlink.ibm.com/public/applications/publications/cgibin/pbi.cgi.
2. Select your country from the list and click **Go**.
3. Click **About this site** in the main panel to see an information page that includes the telephone number of your local representative.

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 additional information, see the *IBM Tivoli Directory Server Version 6.1 Installation and Configuration Guide*.

Tivoli technical training

For Tivoli technical training information, refer to the following IBM Tivoli Education Web site at http://www.ibm.com/software/tivoli/education.

Support information

If you have a problem with your IBM software, you want to resolve it quickly. IBM provides the following ways for you to obtain the support you need:

- IBM Support Assistant: You can search across a large collection of known problems and workarounds, Technotes, and other information at http://www.ibm.com/software/support/isa.
- Obtaining fixes: You can locate the latest fixes that are already available for your product.
- Contacting IBM Software Support: If you still cannot solve your problem, and you need to work with someone from IBM, you can use a variety of ways to contact IBM Software Support.

For more information about resolving problems, see the *IBM Tivoli Directory Server Version 6.1 Problem Determination Guide*.

Conventions used in this book

This book uses several conventions for special terms and actions, operating system-dependent commands and paths, and margin graphics.

Typeface conventions

This book uses the following typeface conventions:

Bold

- Lowercase commands and mixed case commands that are otherwise difficult to distinguish from surrounding text
- Interface controls (check boxes, push buttons, radio buttons, spin buttons, fields, folders, icons, list boxes, items inside list boxes, multicolumn lists, containers, menu choices, menu names, tabs, property sheets), labels (such as **Tip:**, and **Operating system considerations**:)
- Keywords and parameters in text

Italic

- Citations (examples: titles of books, diskettes, and CDs)
- Words defined in text (example: a nonswitched line is called a *point-to-point line*)
- Emphasis of words and letters (words as words example: "Use the word *that* to introduce a restrictive clause."; letters as letters example: "The LUN address must start with the letter *L*.")
- New terms in text (except in a definition list): a *view* is a frame in a workspace that contains data.
- Variables and values you must provide: ... where *myname* represents....

`Monospace`

- Examples and code examples
- File names, programming keywords, and other elements that are difficult to distinguish from surrounding text
- Message text and prompts addressed to the user
- Text that the user must type
- Values for arguments or command options

Operating system-dependent variables and paths

This book uses the UNIX convention for specifying environment variables and for directory notation.

When using the Windows command line, replace **$***variable* with **%** *variable***%** for environment variables and replace each forward slash (**/**) with a backslash (****) in directory paths. The names of environment variables are not always the same in the Windows and UNIX environments. For example, %TEMP% in Windows environments is equivalent to $TMPDIR in UNIX environments.

Note: If you are using the bash shell on a Windows system, you can use the UNIX conventions.

Chapter 1. Command line utilities

This book describes the utilities that can be run from a command prompt.

Command line utilities

The command line utilities are:

Client Utilities

- "idsdirctl, ibmdirctl" on page 3
- "idsldapchangepwd, ldapchangepwd" on page 5
- "idsldapdelete, ldapdelete" on page 10
- "idsldapdiff, ldapdiff" on page 13
- "idsldapexop, ldapexop" on page 23
- "idsldapmodify, ldapmodify, idsldapadd, ldapadd" on page 32
- "idsldapmodrdn, ldapmodrdn" on page 39
- "idsldapsearch, ldapsearch" on page 43
- "idsldaptrace, ldaptrace" on page 54
- "tbindmsg" on page 65

Server Utilities

- "ddsetup" on page 67
- "idsbulkload, bulkload" on page 74
- "idscfgchglg" on page 80
- "idscfgdb" on page 82
- "idscfgsch" on page 83
- "idscfgsuf" on page 84
- "idsdbback, dbback" on page 85
- "idsdbrestore, dbrestore" on page 88
- "idsdb2ldif, db2ldif" on page 89
- "idsdiradm, ibmdiradm" on page 91
- "idsdnpw" on page 92
- "idsgendirksf" on page 94
- "idsicrt" on page 95
- "idsidrop" on page 102
- "idsilist" on page 103
- "idsimigr" on page 105
- "idsldif2db, ldif2db" on page 109
- "idslink" on page 58
- "idslogmgmt" on page 110
- "IDSProgRunner" on page 114
- "idsrunstats, runstats" on page 114
- "idssethost" on page 115
- "idssetport" on page 116
- "idsslapd, ibmslapd" on page 118

- “idssnmp” on page 119
- “idssupport” on page 119
- “idsucfgchglg” on page 119
- “idsucfgdb” on page 120
- “idsucfgsch” on page 122
- “idsucfgsuf” on page 123
- “ldtrc” on page 124
- “idsrun” on page 126

Chapter 2. Client utilities

This section provides a description of the client utilities. The client utilities use the ldap_sasl_bind or ldap_sasl_bind_s API. When bind is invoked, several results can be returned. Following are the bind results using various combinations of user IDs and passwords.

- If specifying the admin DN, the password must be correctly specified or the bind is not successful.
- If a null DN is specified, 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 utilities:

- idsldapchangepwd
- idsldapdelete
- idsldapexop
- idsldapmodify, idsldapadd
- idsldapmodrdn
- idsldapsearch

However, any altered versions of these LDAP utilities are not supported.

idsdirctl, ibmdirctl

The administration daemon control program. The administration daemon (**idsdiradm**) must be running. For more information on the **idsdiradm** utility, refer *Chapter 4, "Directory administration daemon", in the IBM Tivoli Directory Server Version 6.1 Administration Guide.*

Note: Only the administrator may use this utility.

Synopsis

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

where *command:* command to issue to ibmdiradm must be one of {start/stop/restart/status/admstop/startlogmgmt/stoplogmgmt/statuslogmgmt}

start starts the IBM Tivoli Directory Server

stop stops the IBM Tivoli Directory Server

restart stops and starts the IBM Tivoli Directory Server

status displays whether the IBM Tivoli Directory Server is running

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

admstop
stops the IBM Tivoli Directory Server Administration Daemon

startlogmgmt
starts the log management capabilities for the IBM Tivoli Directory Server

stoplogmgmt
stops the log management capabilities for the IBM Tivoli Directory Server

statuslogmgmt
displays whether the log management for the IBM Tivoli Directory Server is running

Description

The administration daemon control program, **ibmdirctl**, is used to start, stop, restart or query the status of the IBM Tivoli Directory Server. It can also be used to stop the administration daemon. If idsslapd options are requested, they must be preceded by the **--**.

To display syntax help for **ibmdirctl**, type `ibmdirctl -?`.

Options

- D adminDN
bind DN. (-d can also be used)

-h hostname
ibmdiradm hostname. (-H can also be used)

-K keyfile
file to use for keys

-N key_name
private key name to use in keyfile

-p port
ibmdiradm port number

-P *key_pw*
keyfile password

-v run in verbose mode

-w *adminPW*
bind password or '?' for non-echoed prompt use backslash '\?' to avoid matching single character filenames (UNIX only)

-W same as -w

-Y use a secure ldap connection (TLS)

-Z use a secure ldap connection (SSL)

-? Displays the syntax format.

Example

To start the server in configuration only mode issue the command:

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

To stop the server issue the command:

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

idsldapchangepwd, ldapchangepwd

The LDAP modify password tool.

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] [-?]
```

Description

Sends modify password requests to an LDAP server.

Notes:

1. idsldapchangepwd cannot be used to change the administrator password or member of administrative group passwords. idsldapchangepwd works only with directory entries.
2. idsldapchangepwd works only on the userpassword attribute.

Options

-C *charset*
: Specifies that the DNs supplied as input to the **idsldapchangepwd** utility are represented in a local character set, as specified by charset. Use **-C** *charset* to override the default, where strings must be supplied in UTF-8. See Appendix B, “IANA character sets supported by platform,” on page 131 for the specific charset values that are supported for each operating system platform. Note that the supported values for charset are the same values 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. This parameter is for use by IBM service personnel. See Chapter 4, “Debugging levels,” on page 127 for additional information on debug levels.

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

-E *token_pw*
: Token password to access the crypto device.

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

-h *ldaphost*
Specify an alternate host on which the LDAP server is running.

-I Crypto device with key storage using PKCS11.

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

A default keyring file that is, ldapkey.kdb, and the associated password stash file that is, ldapkey.sth, are installed in the etc directory under IDS_LDAP_HOME, where IDS_LDAP_HOME is the path to the installed LDAP support. IDS_LDAP_HOME varies by operating system platform:

- AIX operating systems - /opt/IBM/ldap/V6.1
- HP-UX operating systems - /opt/IBM/ldap/V6.1
- Linux operating systems - /opt/ibm/ldap/V6.1
- Solaris operating systems - /opt/IBM/ldap/V6.1
- Windows operating systems - <*local_drive*>:\Program Files\IBM\LDAP\V6.1 (This is the default install location. The actual IDS_LDAP_HOME is determined during installation.)

See the *IBM Tivoli Directory Server Version 6.1 Programming Reference* for more information about default key database files, and default Certificate Authorities.

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 certificate authorities (CAs) that are trusted by the client. These types of X.509 certificates are also known as trusted roots. See the *IBM Tivoli Directory Server version 6.1 Administration Guide* for more information about managing an SSL or TLS key database and for information about SSL and certificates. Also see the “Security functions” on page 8.

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

-m *mechanism*
Use ***mechanism*** to specify the SASL mechanism to be used to bind to the server. The ldap_sasl_bind_s() API will be used. The **-m** parameter is ignored if **-V 2** is set. If **-m** is not specified, simple authentication is used.

-M Manage referral objects as regular entries.

-n *newpassword* **|** **?**
Specifies the new password. Use the ? to generate a password prompt. Using this prompt prevents your password from being visible through the **ps** command.

-N *certificatename*
Specify the label associated with the client certificate in the key database

file. If the LDAP server is configured to perform server authentication only, a client certificate is not required. If the LDAP server is configured to perform client and server Authentication, a client certificate might be required. ***certificatename*** is not required if a default certificate/private key pair has been designated 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 neither **-Z** nor **-K** is specified.

-O *maxhops*
: Specify ***maxhops*** to set the maximum number of hops that the client library takes when chasing referrals. The default hopcount is 10.

-p *ldapport*
: Specify an alternate TCP port where the LDAP server is listening. The default LDAP port is 389. If **-p** is not specified and **-Z** is specified, the default LDAP SSL port 636 is used.

-P *keyfilepw*
: Specify the key database password. This password is required to access the encrypted information in the key database file, which may include one or more private keys. If a password stash file is associated with the key database file, the password is obtained from the password stash file, and the **-P** parameter is not required. This parameter is ignored if neither **-Z** nor **-K** is specified.

-Q *operation*
: Crypto device operation with PKCS11

```
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 that referrals are not to be automatically followed.

-S *token_label*
: Token label of the crypto device.

-U *username*
: Specifies the username. This is required with -m DIGEST-MD5 and ignored when any other mechanism is used. The value ***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 Use verbose mode, with many diagnostics written to standard output.

-V *version*
: Specifies the LDAP version to be used by **ldapdchangepwd** when it binds to the LDAP server. By default, an LDAP V3 connection is established. To explicitly select LDAP V3, specify **-V 3**. Specify **-V 2** to run as an LDAP V2 application. An application, like **ldapdchangepwd**, selects LDAP V3 as the preferred protocol by using ldap_init instead of ldap_open.

-w *passwd* **|** **?**
: Use ***passwd*** as the password for authentication. Use the ? to generate a password prompt. Using this prompt prevents your password from being visible through the **ps** command.

-x Use FIPS mode processing (SSL/TLS only).

-X *lib_path*
Driver path of the crypto device.

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

-Y Use a secure TLS connection to communicate with the LDAP server. The **-Y** option is only supported when IBM's GSKit, is installed.

-Z Use a secure SSL connection to communicate with the LDAP server. The **-Z** option is only supported when the SSL component entry, as provided by IBM's GSKit, is installed.

-? Displays the syntax format.

Examples

The following command,

```
idsldapchangepwd -D "cn=John Doe" -w a1b2c3d4 -n wxyz9876
```

changes the password for the entry named with commonName ″John Doe″ from a1b2c3d4 to wxyz9876

Security functions

To use the SSL or TLS -related functions associated with this utility, see "SSL, TLS notes" on page 65.

Diagnostics

Exit status is 0 if no errors occur. Errors result in a non-zero exit status and a diagnostic message being written to standard error.

See also

idsldapadd, idsldapdelete, idsldapexop, idsldapmodify, idsldapmodrdn, idsldapsearch

idsldapcompare, ldapcompare

The ldapcompare utility sends a compare request to an LDAP server.

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]
```

Description

The ldapcompare utility compares the attribute value of an entry with a user provided value.

The syntax of the ldapcompare command is:

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

where:

- dn: The dn entry for compare.
- attr: The attribute to be used in the compare.

- value: The value to be used in the compare.

Options

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

-d <*level*>
Sets the debug level to <*level*> in the LDAP library.

-D <*dn*>
Specifies the bind dn used to bind to a directory server.

-f <*file*>
Specifies to perform sequence of compares using the values in the *file*.

-G <*realm*>
Specifies the realm used for DIGEST-MD5 bind mechanism.

-h <*host*>
Specifies the LDAP server host name.

-m <*mechanism*>
Specifies the mechanism to be used with the SASL bind to bind to a server.

-n Demonstrates what action would be performed without actually performing it.

Note: This option is useful for debugging when used in conjunction with –v.

-p <*port*>
Specifies the port number on which the LDAP server listens.

-P <*on|off*>
Specifies whether to send password policy controls. The parameter along with -P can be either "on" or "off", which implies:

- 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 DIGEST-MD5 bind mechanism.

-v Specifies to run the command in the verbose mode.

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

-w <*password*>
Specifies the bind password.

-y <*proxydn*>
Specifies to set a proxied id for the proxied authorization operation.

Examples

Consider an example given below:

```
ldapcompare -D <adminDN> -w <adminPWD> -h <localhost> -p <port>
"cn=Bob Campbell, ou=In Flight Systems, ou=Austin, o=sample" postalcode=4502
```

This command compares the entry with an entry existing in the DIT. Now, if the entry cn=Bob Campbell has its postal code as 4502 in the DIT, the above command will return true. Otherwise it returns false.

The same result can be achieved by using an ldif file with the -f option as shown below:

```
ldapcompare -D <adminDN> -w <adminPWD> -h <localhost> -p <port> -f myfile

where myfile contains the following

cn=Bob Campbell, ou=In Flight Systems, ou=Austin, o=sample
postalcode: 4502
```

The –f option is useful when you need to compare more than one entry using a single command.

idsldapdelete, ldapdelete

The LDAP delete-entry tool

Synopsis

```
usage:
 ldapdelete [options] [DNs]
 ldapdelete [options] [-i file]

where:
      dn: one or more items to delete
      file: name of input file containing items to delete
```

Note: If neither dn nor file is specified then items are read from standard input.

Description

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

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

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

```
idsldapdelete -?
```

Options

-c Continuous operation; do not stop processing on error.

-C *charset*
Character set name to use, as registered with IANA. See Appendix B, “IANA character sets supported by platform,” on page 131 for the specific charset values that are supported for each operating system platform. Note that the supported values for charset are the same values supported for the charset tag that is optionally defined in Version 1 LDIF files.

-d <*level***>**
Set debug level in LDAP library. The <*debuglevel*> is a bit mask that controls which output is generated with values up to 65535. This

parameter is for use by IBM service personnel. See Chapter 4, "Debugging levels," on page 127 for additional information on debug levels.

-D *dn* Bind dn

-E *token_pw*
Token password to access the crypto device.

-f *file* Read dn's from a file for deletion, one dn per line.

-G *realm*
Realm used for DIGEST-MD5 bind mechanism.

-h *host* LDAP server host name.

-i *file* Read dn's from a file for deletion, one dn per line.

-I Crypto device with key storage using PKCS11.

-k Use server administration control.

This option sends the Server administration control. See the *IBM Tivoli Directory Server Version 6.1 Programming Reference* for information about this control.

-K *keyfile*
File to use for keys.

A default keyring file that is, ldapkey.kdb, and the associated password stash file that is, ldapkey.sth, are installed in the etc directory under IDS_LDAP_HOME, where IDS_LDAP_HOME is the path to the installed LDAP support. IDS_LDAP_HOME varies by operating system platform:

- AIX operating systems - /opt/IBM/ldap/V6.1
- HP-UX operating systems - /opt/IBM/ldap/V6.1
- Linux operating systems - /opt/ibm/ldap/V6.1
- Solaris operating systems - /opt/IBM/ldap/V6.1
- Windows operating systems - <*local_drive*>:\Program Files\IBM\LDAP\V6.1 (This is the default install location. The actual IDS_LDAP_HOME is determined during installation.)

See the *IBM Tivoli Directory Server Version 6.1 Programming Reference* for more information about default key database files, and default Certificate Authorities.

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 certificate authorities (CAs) that are trusted by the client. These types of X.509 certificates are also known as trusted roots. See the *IBM Tivoli Directory Server version 6.1 Administration Guide* for more information about managing an SSL or TLS key database and for information about SSL and certificates. Also see the "Security functions" on page 8.

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

-l Do not replicate the entry.

This option sends the Do not replicate control. See the *IBM Tivoli Directory Server Version 6.1 Programming Reference* for information about this control.

-m *mechanism*
Perform SASL bind with the given mechanism.

-M Manage referral objects as normal entries.

-n Show what would be done but don't actually do it. Useful for debugging in conjunction with **-v**.

-N *key_name*
Private key name to use in keyfile. This parameter is ignored if neither **-Z** nor **-K** is specified.

-O *maxhops*
Maximum number of referrals to follow in a sequence.

-p *port*
LDAP server port number. If **-p** is not specified and **-Z** is specified, the default LDAP SSL port 636 is used.

-P *key_pw*
Keyfile password. This parameter is ignored if neither **-Z** nor **-K** is specified.

-Q *operation*
Crypto device operation with PKCS11

```
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 Do not chase referrals.

-s Delete subtree.

This option sends the Subtree delete control. See the *IBM Tivoli Directory Server Version 6.1 Programming Reference* for information about this control.

-S *token_label*
Token label of the crypto device.

-U *username*
User name for DIGEST-MD5 bind mechanism. This is required with -m DIGEST-MD5 and ignored when any other mechanism is used. The value ***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 Verbose mode.

-V LDAP protocol version (2 or 3; default is 3).

-w *password*
Bind password or '?'. For non-echoed prompt use backslash '\?' to avoid matching single character filenames (UNIX only).

-x Use FIPS mode processing (SSL/TLS only).

-X *lib_path*
Driver path of the crypto device.

-y *proxydn*
Set proxied id for proxied authorization operation.

-Y Use a secure LDAP connection (TLS).

-Z Use a secure LDAP connection (SSL).

Examples

The following command,

```
idsldapdelete "cn=Delete Me, o=University of Life, c=US"
```

attempts to delete the entry named with commonName ″Delete Me″ directly below the University of Life organizational entry. It might be necessary to supply a ***binddn*** and ***passwd*** for deletion to be allowed (see the **-D** and **-w** options).

Notes

If no DN arguments are provided, the **idsldapdelete** command waits to read a list of DNs from standard input. To break out of the wait, use Ctrl+D. For Windows, use Ctrl+Z.

Security functions

To use the SSL or TLS -related functions associated with this utility, see "SSL, TLS notes" on page 65.

Diagnostics

Exit status is 0 if no errors occur. Errors result in a non-zero exit status and a diagnostic message being written to standard error.

See also

idsldapadd, idsldapchangepwd, idsldapexop, idsldapmodify, idsldapmodrdn, idsldapsearch

idsldapdiff, ldapdiff

The **idsldapdiff** utility identifies differences in a replica server and its master, and can be used to synchronize replicas.

Synopsis

To compare and optionally fix:

```
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]
```

or to compare schema only:

```
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]
```

Description

The **idsldapdiff** command line utility is designed to compare two directory subtrees on two different directory servers to determine if their contents match.

The utility can also optionally synchronize any entries that do not match. The following are two types of differences that might have to be synchronized:

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

The following is a list of 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

Other operational attributes

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

Run the utility when no updates are being made to either of the directory servers. The administrator needs to quiesce or suspend all update activity to the two subtrees being compared. This must be done manually before invoking the compare tool. If the tool is run while updates are being made, it cannot be guaranteed that all discrepancies are accurately reported or fixed.

Note: The tool does not check on startup whether the servers are quiesced. When the tool is used in compare-only mode, the administrator might want to track down a small number of discrepancies as an alternative to stopping updates completely.

Use the tool with the server administration control (**-a** flag), if the fix operation is requested. The server administration control allows the tool to write to a read-only replica, and it also allows it to modify operational attributes such as ibm-entryUuid.

The **idsldapdiff** utility can be used to bring a master and replica server in sync before starting replication. The tool requires that the base DN, which is being compared, exists on both servers. If the base DN does not exist on either of the two servers, the utility gives an error and exits.

The tool traverses each entry in the directory subtree on the supplier server and compares its contents with the corresponding entry on the consumer server. Because information about each entry needs to be read, running the utility can take a long time and can generate lots of read requests to the supplier and consumer servers. Depending on how many differences are found and whether the fix operation is specified, the utility can also generate an equal amount of write requests to the consumer server.

Ideally, the tool is used only once between servers, when replication is initially setup. 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. Then, 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 to the directory on the master servers is propagated to the replicas. However, if a problem occurs, the tool can be run to identify and correct replication problems. This utility is a diagnostic and corrective tool, it is not designed to run as routine maintenance. Depending on the replication-related errors observed in the log files, an administrator might decide to run the utility.

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

```
idsldapdiff -?
```

Note: The **idsldapdiff** utility displays a message after it has finished comparing every 100th entry.

Encryption considerations

idsldapdiff performs ″cn=configuration″ searches to determine the encryption settings on the server. Also, for performing searches and fixes, 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** compare and fix options. Only administrators and administrator group members can run the **idsldapdiff** compare and fix options.

The supplier and consumer servers can have different encryption settings:

- Non-matching one-way
- Two-way and one-way
- Two-way with different stash files

Based on the types of encryption used, different behaviors occur when a password or any other password attribute is encountered.

Non-matching one-way

In this case the servers are using different types of one-way encryption. For example, the master server uses sha and the replica server uses crypt. The consumer values are directly overwritten with the value on the supplier. Running the **idsldapdiff** tool a second time on the same entries does not show any difference.

Two-way and one-way

In this case the one of the servers is using a two-way encryption algorithm like AES and the other server is using one-way encrytpion such as sha. Depending on whether the master server is using two-way or one-way encryption the behavior results are different. In this situation the performance of the **idsldapdiff** utility is degraded.

- When the supplier has a two-way encryption and the consumer has a one-way encryption, the **idsldapdiff** utility shows the two entries as always being different even if the actual values are same. The supplier value is in plain text (decrypted because it is two-way) and consumer value is encrypted (because it is one way). Running the **idsldapdiff** tool a second time on the same entries still shows a difference even though the actual values are the same.

- When the supplier has a one-way encryption and the consumer has a two-way encryption, the consumer values are directly overwritten with the value on the supplier. Running the **idsldapdiff** tool a second time on the same entries does not show any difference.

Two-way encrypted data with different key stash files

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

Notes:

1. The password policy attributes are synchronized by the **idsldapdiff** utility only if password policy is enabled on both of the servers.
2. The **idsldapdiff** utility checks the encryption settings on both of the servers and displays warning messages if the encryption settings are different both of 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 following options apply to the **idsldapdiff** command. There are two subgroupings that apply specifically to either the supplier server or the consumer server.

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

-b *baseDN*

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.

-C *countnumber*

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

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

-j Indicates to not include the following operational attributes in the LDIF file:

- creatorsName
- createTimeStamp
- modifiersName
- modifyTimeStam

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

-L <*filename*>
If the **-F** option is not specified, use this option to generate an LDIF file for output. The LDIF file can be used to update the consumer to eliminate the differences.

-O Displays DNs only 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 using **-S** can be made with any bind DN.

-v Use verbose mode, with many diagnostics written to standard output.

-x Ignore extra entries on the consumer.

idsldapdiff performs two passes to make the servers are in sync. In the first pass, idsldapdiff traverses the Supplier server and does the following:

- Adds any extra entries on the supplier and to the consumer
- Compares and fixes entries that exist on both the servers

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

Options for a replication supplier

The following options apply to the supplier server and are denoted by an initial ’s’ in the option name.

-sD *dn* Use ***dn*** to bind to the LDAP directory. ***dn*** is a string-represented DN.

-sh *host*
Specifies the host name.

-sK *keyStore*
Specify the name of the SSL key database file with default extension of **kdb**. If the key database file is not in the current directory, specify the fully-qualified key database filename. If a key database filename is not specified, this utility will first look for the presence of the SSL_KEYRING environment variable with an associated filename. If the SSL_KEYRING environment variable is not defined, the default keyring file will be used, if present.

A default keyring file that is, ldapkey.kdb, and the associated password stash file that is, ldapkey.sth, are installed in the etc directory under IDS_LDAP_HOME, where IDS_LDAP_HOME is the path to the installed LDAP support. IDS_LDAP_HOME varies by operating system platform:

- AIX operating systems - /opt/IBM/ldap/V6.1
- HP-UX operating systems - /opt/IBM/ldap/V6.1
- Linux operating systems - /opt/ibm/ldap/V6.1
- Solaris operating systems - /opt/IBM/ldap/V6.1
- Windows operating systems - <*local_drive*>:\Program Files\IBM\LDAP\V6.1 (This is the default install location. The actual IDS_LDAP_HOME is determined during installation.)

See the *IBM Tivoli Directory Server Version 6.1 Programming Reference* for more information about default key database files, and default Certificate Authorities.

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 certificate authorities (CAs) that are trusted by the client. These types of X.509 certificates are also known as trusted roots. See the *IBM Tivoli Directory Server version 6.1 Administration Guide* for more information about managing an SSL or TLS key database and for information about SSL and certificates. Also see the "Security functions" on page 8.

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

-sN *keyStoreType*
Specify the label associated with the client certificate in the key database file. If the LDAP server is configured to perform server authentication only, a client certificate is not required. If the LDAP server is configured to perform client and server Authentication, a client certificate might be required. ***keyStoreType*** is not required if a default certificate/private key pair has been designated as the default. Similarly, ***keyStoreType*** is not required if there is a single certificate/private key pair in the designated key database file. This parameter is ignored if neither **-sZ** nor **-sK** is specified.

-sp *ldapport*
Specify an alternate TCP port where the LDAP server is listening. The default LDAP port is 389. If **-sp** is not specified and **-sZ** is specified, the default LDAP SSL port 636 is used.

-sP *keyStorePwd*
Specify the key database password. This password is required to access the encrypted information in the key database file, which may include one or more private keys. If a password stash file is associated with the key database file, the password is obtained from the password stash file, and the **-sP** parameter is not required. This parameter is ignored if neither **-sZ** nor **-sK** is specified.

-st *trustStoreType*
Specify the label associated with the client certificate in the trust database file. If the LDAP server is configured to perform server authentication only, a client certificate is not required. If the LDAP server is configured to perform client and server Authentication, a client certificate might be required. ***trustStoreType*** is not required if a default certificate/private key pair has been designated as the default. Similarly, ***trustStoreType*** is not required if there is a single certificate/private key pair in the designated key database file. This parameter is ignored if neither **-sZ** nor **-sT** is specified.

-sT *trustStore*
Specify the name of the SSL trust database file with default extension of **tdb**. If the trust database file is not in the current directory, specify the fully-qualified trust database filename. If a trust database filename is not specified, this utility will first look for the presence of the SSL_KEYRING environment variable with an associated filename. If the SSL_KEYRING environment variable is not defined, the default keyring file will be used, if present.

A default keyring file that is, ldapkey.tdb, and the associated password stash file that is, ldapkey.sth, are installed in the etc directory under IDS_LDAP_HOME, where IDS_LDAP_HOME is the path to the installed LDAP support. IDS_LDAP_HOME varies by operating system platform:

- AIX operating systems - /opt/IBM/ldap/V6.1
- HP-UX operating systems - /opt/IBM/ldap/V6.1
- Linux operating systems - /opt/ibm/ldap/V6.1
- Solaris operating systems - /opt/IBM/ldap/V6.1
- Windows operating systems - <*local_drive*>:\Program Files\IBM\LDAP\V6.1 (This is the default install location. The actual IDS_LDAP_HOME is determined during installation.)

See the *IBM Tivoli Directory Server Version 6.1 Programming Reference* for more information about default key database files, and default Certificate Authorities.

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 certificate authorities (CAs) that are trusted by the client. These types of X.509 certificates are also known as trusted roots. See the *IBM Tivoli Directory Server version 6.1 Administration Guide* for more information about managing an SSL or TLS key database and for information about SSL and certificates. Also see the “Security functions” on page 8.

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

-sw *password* **|** **?**
Use ***password*** as the password for authentication. Use the ? to generate a password prompt. Using this prompt prevents your password from being visible through the **ps** command.

-sY The password for the trusted database.

-sZ Use a secure SSL connection to communicate with the LDAP server. The **-Z** option is only supported when the SSL component entry, as provided by IBM’s GSKit, is installed.

Options for a replication consumer

The following options apply to the consumer server and are denoted by an initial ′c′ in the option name.

-cD *dn* Use ***dn*** to bind to the LDAP directory. ***dn*** is a string-represented DN.

-ch *host*
Specifies the host name.

-cK *keyStore*
Specify the name of the SSL key database file with default extension of **kdb**. If the key database file is not in the current directory, specify the fully-qualified key database filename. If a key database filename is not specified, this utility will first look for the presence of the SSL_KEYRING environment variable with an associated filename. If the SSL_KEYRING environment variable is not defined, the default keyring file will be used, if present.

A default keyring file that is, ldapkey.kdb, and the associated password stash file that is, ldapkey.sth, are installed in the etc directory under IDS_LDAP_HOME, where IDS_LDAP_HOME is the path to the installed LDAP support. IDS_LDAP_HOME varies by operating system platform:

- AIX operating systems - /opt/IBM/ldap/V6.1
- HP-UX operating systems - /opt/IBM/ldap/V6.1

- Linux operating systems - /opt/ibm/ldap/V6.1
- Solaris operating systems - /opt/IBM/ldap/V6.1
- Windows operating systems - <*local_drive*>:\Program Files\IBM\LDAP\V6.1 (This is the default install location. The actual IDS_LDAP_HOME is determined during installation.)

See the *IBM Tivoli Directory Server Version 6.1 Programming Reference* for more information about default key database files, and default Certificate Authorities.

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 certificate authorities (CAs) that are trusted by the client. These types of X.509 certificates are also known as trusted roots. See the *IBM Tivoli Directory Server version 6.1 Administration Guide* for more information about managing an SSL or TLS key database and for information about SSL and certificates. Also see the "Security functions" on page 8.

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

-cN *keyStoreType*

Specify the label associated with the client certificate in the key database file. If the LDAP server is configured to perform server authentication only, a client certificate is not required. If the LDAP server is configured to perform client and server Authentication, a client certificate might be required. ***keyStoreType*** is not required if a default certificate/private key pair has been designated as the default. Similarly, ***keyStoreType*** is not required if there is a single certificate/private key pair in the designated key database file. This parameter is ignored if neither **-cZ** nor **-cK** is specified.

-cp *ldapport*

Specify an alternate TCP port where the LDAP server is listening. The default LDAP port is 389. If **-cp** is not specified and **-cZ** is specified, the default LDAP SSL port 636 is used.

-cP *keyStorePwd*

Specify the key database password. This password is required to access the encrypted information in the key database file, which may include one or more private keys. If a password stash file is associated with the key database file, the password is obtained from the password stash file, and the **-cP** parameter is not required. This parameter is ignored if neither **-cZ** nor **-cK** is specified.

-ct *trustStoreType*

Specify the label associated with the client certificate in the trust database file. If the LDAP server is configured to perform server authentication only, a client certificate is not required. If the LDAP server is configured to perform client and server Authentication, a client certificate might be required. ***trustStoreType*** is not required if a default certificate/private key pair has been designated as the default. Similarly, ***trustStoreType*** is not required if there is a single certificate/private key pair in the designated key database file. This parameter is ignored if neither **-cZ** nor **-cT** is specified.

-cT *trustStore*

Specify the name of the SSL trust database file with default extension of

tdb. If the trust database file is not in the current directory, specify the fully-qualified trust database filename. If a trust database filename is not specified, this utility will first look for the presence of the SSL_KEYRING environment variable with an associated filename. If the SSL_KEYRING environment variable is not defined, the default keyring file will be used, if present.

A default keyring file that is, ldapkey.tdb, and the associated password stash file that is, ldapkey.sth, are installed in the etc directory under IDS_LDAP_HOME, where IDS_LDAP_HOME is the path to the installed LDAP support. IDS_LDAP_HOME varies by operating system platform:

- AIX operating systems - /opt/IBM/ldap/V6.1
- HP-UX operating systems - /opt/IBM/ldap/V6.1
- Linux operating systems - /opt/ibm/ldap/V6.1
- Solaris operating systems - /opt/IBM/ldap/V6.1
- Windows operating systems - <*local_drive*>:\Program Files\IBM\LDAP\V6.1 (This is the default install location. The actual IDS_LDAP_HOME is determined during installation.)

See the*IBM Tivoli Directory Server Version 6.1 Programming Reference* for more information about default key database files, and default Certificate Authorities.

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 certificate authorities (CAs) that are trusted by the client. These types of X.509 certificates are also known as trusted roots. See the *IBM Tivoli Directory Server version 6.1 Administration Guide* for more information about managing an SSL or TLS key database and for information about SSL and certificates. Also see the "Security functions" on page 8.

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

-cw *password* **|** **?**
Use ***password*** as the password for authentication. Use the ? to generate a password prompt. Using this prompt prevents your password from being visible through the **ps** command.

-cY
The password for the trusted database.

-cZ
Use a secure SSL connection to communicate with the LDAP server. The **-cZ** option is only supported when the SSL component entry, as provided by IBM's GSKit, is installed.

Examples

```
idsldapdiff -b <baseDN> -sh <supplierhostname> -ch <consumerhostname> [options]
```

or

```
idsldapdiff -S  -sh <supplierhostname> -ch <consumerhostname> [options]
```

As an illustration of how the utility works, set up two servers one as a master server and other as a replica server. Assume that Suffix o=ibm, c=us is present on both the servers. Create two LDIF files master.ldif and replica.ldif

master.ldif with entries

```
dn: cn=Entry1,o=sample
 objectclass: inetOrgPerson
 objectclass: organizationalPerson
 objectclass: person
 objectclass: top
 objectclass: ePerson
 sn: entry1
 cn: testEntry

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

replica.ldif with entries

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

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

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 resulting actions are:

1. Entry cn=Entry1,o=sample gets added on 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 Replica server. The value of sn field gets modified to match the value on the master server.
3. Entry cn=Entry3,o=sample get deleted from Replica server. This entry is extra on the replica server that was not on the master server.

Notes

If no DN arguments are provided, the **idsldapdiff** command waits to read a list of DNs from standard input. To break out of the wait, use Ctrl+C or Ctrl+D.

Security functions

To use the SSL or TLS -related functions associated with this utility, see "SSL, TLS notes" on page 65.

Diagnostics

Exit status is 0 if no errors occur. Errors result in a non-zero exit status and a diagnostic message being written to standard error.

idsldapexop, ldapexop

The LDAP extended operation tool.

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]
              -op {acctstatus | cascrepl | clearlog | controlqueue | controlrepl |
              controlreplerr | evaluategroups | effectpwdpolicy | getattributes | getlogsize |
              getusertype | locateEntry | onlineBackup | quiesce | readconfig |
              readlog | repltopology | resumerole | stopserver | unbind | uniqueattr }
```

Description

The **idsldapexop** utility is a command-line interface that provides the capability to bind to a directory and issue a single extended operation along with any data that makes up the extended operation value.

The **idsldapexop** utility supports the standard host, port, SSL, TLS, and authentication options used by all of the LDAP client utilities. In addition, a set of options is defined to specify the operation to be performed, and the arguments for each extended operation

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

```
idsldapexop -?
```

or

```
idsldapexop -help
```

Options

The options for the **idsldapexop** command are divided into two categories:

1. General options that specify how to connect to the directory server. These options must be specified before operation specific options.
2. Extended operation option that identifies the extended operation to be performed.

General options

These options specify the methods of connecting to the server and must be specified before the **-op** option.

-C <*charset*>
Specifies that the DNs supplied as input to the **idsldapexop** 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. See Appendix B, “IANA character sets supported by platform,” on page 131 for the specific charset values that are supported for each operating system platform. Note that the supported values for charset are the same values 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. This parameter is for use by IBM service personnel. See Chapter 4, "Debugging levels," on page 127 for additional information on debug levels.

-D <*binddn*>
Use ***binddn*** to bind to the LDAP directory. ***binddn*** is a string-represented DN. When used with -m DIGEST-MD5, it specifies the authorization ID. It can be either a DN or an authzId string that starts with ″u:″ or ″dn:″.

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

-E *token_pw*
Token password to access the crypto device.

-G <*realm*>
Specify the name of the realm. When used with the -m DIGEST-MD5, the value is passed to the server during the bind.

-h <*ldaphost*>
Specify an alternate host on which the LDAP server is running.

-I Crypto device with key storage using PKCS11.

-help Displays the usage

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

A default keyring file that is, ldapkey.kdb, and the associated password stash file that is, ldapkey.sth, are installed in the etc directory under IDS_LDAP_HOME, where IDS_LDAP_HOME is the path to the installed LDAP support. IDS_LDAP_HOME varies by operating system platform:

- AIX operating systems - /opt/IBM/ldap/V6.1
- HP-UX operating systems - /opt/IBM/ldap/V6.1
- Linux operating systems - /opt/ibm/ldap/V6.1
- Solaris operating systems - /opt/IBM/ldap/V6.1
- Windows operating systems - <*local_drive*>:\Program Files\IBM\LDAP\V6.1 (This is the default install location. The actual IDS_LDAP_HOME is determined during installation.)

See the *IBM Tivoli Directory Server Version 6.1 Programming Reference* for more information about default key database files, and default Certificate Authorities.

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 certificate authorities (CAs) that are trusted by the client. These types of X.509 certificates are also known as trusted roots. See the *IBM Tivoli Directory Server version 6.1 Administration*

Guide for more information about managing an SSL or TLS key database and for information about SSL and certificates. Also see the "Security functions" on page 8.

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

-m <*mechanism*>
Use ***mechanism*** to specify the SASL mechanism to be used to bind to the server. The ldap_sasl_bind_s() API will be used. The **-m** parameter is ignored if **-V 2** is set. If **-m** is not specified, simple authentication is used.

-N <*certificatename*>
Specify the label associated with the client certificate in the key database file. If the LDAP server is configured to perform server authentication only, a client certificate is not required. If the LDAP server is configured to perform client and server Authentication, a client certificate might be required. ***certificatename*** is not required if a default certificate/private key pair has been designated 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 neither **-Z** nor **-K** is specified.

-p <*ldapport* >
Specify an alternate TCP port where the LDAP server is listening. The default LDAP port is 389. If **-p** is not specified and **-Z** is specified, the default LDAP SSL port 636 is used.

-P <*keyfilepw*>
Specify the key database password. This password is required to access the encrypted information in the key database file, which may include one or more private keys. If a password stash file is associated with the key database file, the password is obtained from the password stash file, and the **-P** parameter is not required. This parameter is ignored if neither **-Z** nor **-K** is specified.

-? Displays the syntax format.

-Q *operation*
Crypto device operation with PKCS11

```
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*
Token label of the crypto device.

-U <*username*>
Specifies the username. This is required with -m DIGEST-MD5 and ignored when any other mechanism is used. The value ***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 Use verbose mode, with many diagnostics written to standard output.

-w <*passwd*> **| ?**
Use ***passwd*** as the password for authentication. Use the ? to generate a password prompt. Using this prompt prevents your password from being visible through the **ps** command.

-x Use FIPS mode processing (SSL/TLS only).

-X *lib_path*
Driver path of the crypto device.

-y <*proxyDN*>
Sets a proxied ID for proxied authorization operation.

-Y Use a secure TLS connection to communicate with the LDAP server. The **-Y** option is only supported when IBM's GSKit, is installed.

-Z Use a secure SSL connection to communicate with the LDAP server. The **-Z** option is only supported when the SSL component entry, as provided by IBM's GSKit, is installed.

Extended operations option

The **-op** extended-op option identifies the extended operation to be performed. The extended operation can be one of the following values:

- **acctStatus -d**<*userDN*>: password policy account status extended operation. This operation enables a directory administrator to query the server as to the account status of any entry that contains a userPassword attribute. The *userDN* is the DN of the user account that is being queried. The status for the account is open, locked, or expired.

  Examples:

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

- **cascrepl -action**<*actionvalue*> **-rc**<*contextDN*> [*options*]: cascading control replication extended operation. The requested action is applied to the specified server and also passed along to all replicas of the given subtree. If any of these are forwarding replicas, they pass the extended operation along to their replicas. The operation cascades over the entire replication topology.

  -action {quiesce | unquiesce | replnow | wait}
  This is a required attribute that specifies the action to be performed.

  quiesce
  No further updates are allowed, except by replication.

  unquiesce
  Resume normal operation, client updates are accepted.

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

  wait Wait for all updates to be replicated to all replicas.

  -rc *contextDn*
  This is a required attribute that specifies the root of the subtree.

  options

  -timeout *secs*
  This is an optional attribute that if present, specifies the timeout period in seconds. If not present, or 0, the operation waits indefinitely.

  Example:

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

- **clearlog -log**<*logname*>: clear log file extended operation

  -log {audit | bulkload | cli | slapd | idsdiradm | adminAudit | debug | LostAndFound | config}
  This is a required attribute that specifies the log file to be cleared.

  Example:

  ```
  idsldapexop -D <bindDN> -W <password> -op clearlog -log audit
  ```

- **controlqueue -skip**<*skipvalue*> **-ra**<*agreementDN*>: control queue extended operation

  -skip {all | change-id}
  This is a required attribute.
  - **all** indicates to skip all pending changes for this agreement.
  - **change-id** identifies the single change to be skipped. If the server is not currently replicating this change, the request fails.

  -ra *agreementDN*
  This is a required attribute that specifies the DN of the replication agreement.

  Examples:

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

  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*>**}**: control replication extended operation

  -action {suspend | resume | replnow}
  This is a required attribute that specifies the action to be performed.

  -rc *contextDn* **| -ra** *agreementDn*
  The **-rc** *contextDn* is the DN of the replication context. The action is performed for all agreements for this context. The **-ra** *agreementDn* is the DN of the replication agreement. The action is performed for the specified replication agreement.

  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*>: control replication error extended operation

  -delete *failure-ID* **| all**
  Specifies to remove the failed update, where

  all Specifies to delete all the failed updates for this agreement.

  failure-ID
  Specifies to delete only the failed update specified by the failure-ID for this agreement.

  -retry *failure-ID* **| all**
  Specifies to retry the failed update, where

all Specifies to retry all the failed updates for this agreement.

failure-ID
Specifies to retry only the failed update specified by the failure-ID for this agreement.

-show *failure-ID*
Specifies to show the failed update specified by the failure-ID.

-ra *agreementDn*
The **-ra** *agreementDn* is the DN of the replication agreement. The action is performed for the specified replication agreement.

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...***]** : request evaluategroups extended operation

-d <*specificDN*>
Specifies the DN that is to be evaluated to determine what 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 specified, the specified DN is evaluated for static groups only.

An attribute value pair is an attribute type and attribute value separated by an equal sign. A user's attributes are required for evaluating group membership for dynamic group. When the server receives an evaluate group request with attributes, it is these attributes that are used in the group evaluation.

Example:

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

- **getattributes -attrType**<*type*> **-matches** <*value*>

-attrType {operational | language_tag | attribute_cache | unique | configuration | encryptable | encrypted}
This is a required attribute that specifies type of attribute being requested.

-matches {true | false}
Specifies whether the list of attributes returned matches the attribute type specified by the **-attrType** option.

Example:

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

Returns a list of all attributes that can be defined as unique attributes.

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

Returns a list of all attributes that have been not been defined as unique attributes.

- **getlogsize -log**<*logname*>: request log file size extended operation

-log {audit | bulkload | cli | slapd | idsdiradm | adminAudit | debug | LostAndFound | config}
This is a required attribute that specifies the log file to be queried. The size of the log file, in lines, is written to standard output.

Example:

```
idsldapexop -D <AdminDN> -w <Adminpw> -op getlogsize -log slapd
2000 lines
```

- **effectpwdpolicy -d** < user DN or a group DN>: This extended operation queries the effective password policy of a user or a group.

  Example:

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

- **getusertype**: request user type extended operation

  This extended operation returns the user type based on the bound DN.

  Example:

  ```
  idsldapexop -D <AdminDN> -w <Adminpw> -op getusertype
  ```

  returns:

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

  For an administrative group member who is assigned “ReplicationAdmin” and “ServerStartStopAdmin” roles , the output of the extended operation will be:

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

  If “No Administrator” role is assigned for an administrative group member, the output of this extended operation will be:

  ```
  User    : admin_group_member
  Role(s) : no_administrator
  ```

- **locateEntry**: locate entry extended operation

  –d "DN" | -f "<file Name containing DN list>" [-c]

  This extended operation is used to extract the back-end server details of a given set of entry DNs and provide the details to the client.

  To extract the details of a single entry DN the –d option is used. To extract details of a set of DNs, place the entire set of DNs in a file and use the –f option to pass the file to ldapexop operation.

  Example:

  ```
  idsldapexop -D <binddn> -w <bindpw> -op locateEntry –d "cn=user,o=sample"
  ```

- **onlineBackup**: online backup extended operation

  –path <directoryPath>

  This extended operation performs an online backup of the directory server instance’s DB2 database.

  Example:

  Issue the following command to perform an online backup of the directory server instance’s DB2 database:

  ```
  idsldapexop -D <bindDN> -w <bindpw> -op onlineBackup –path <directoryPath>
  ```

- **quiesce -rc <***contextDN***>[***options***]**: quiesce or unquiesce subtree extended operation

-rc *contextDN*

This is a required attribute that specifies the DN of the replication context (subtree) to be quiesced or unquiesced.

options

-end This is an optional attribute that if present, specifies to unquiesce the subtree. If not specified the default is to quiesce the subtree.

Examples:

```
idsldapexop -op quiesce -rc "o=acme,c=us"

idsldapexop -op quiesce -end -rc "o=sample"
```

- **readconfig -scope<***scopevalue***>**: reread configuration file extended operation

  -scope {entire | single<*entry DN***><***attribute***> | entry <***entry DN***> | subtree <***entry DN***>}**

  This is a required attribute.

  - **entire** indicates to reread the entire configuration file.
  - **single** *entry DN><attribute* means to read the single entry and attribute specified.
  - **entry** <*entry DN*> means to read the entry specified.
  - **subtree** <*entry DN*> means to read the entry and the entire subtree under it.

  Examples:

  ```
  idsldapexop -D <AdminDN> -w <Adminpw> -op readconfig -scope entire

  idsldapexop -D <AdminDN> -w <Adminpw> -op readconfig -scope
    single "cn=configuration" ibm-slapdAdminPW
  ```

- **readlog -log <***logname***> -lines <***value***>**: request lines from log file extended operation

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

  This is a required attribute that specifies the log file to be queried.

  -lines {<*first***><***last***> | all}**

  This is a required attribute that specifies either the first and last lines to be read from the file or all lines. Lines are numbered starting at 0. The specified lines are written to standard output.

  Examples:

  ```
  idsldapexop -D <AdminDN> -w <Adminpw> -op readlog -log audit -lines 10 20

  idsldapexop -op readlog -log slapd -lines all
  ```

- **repltopology -rc<***contextDN***> [***options***]**: replication topology extended operation. This operation replicates the replication topology related entries under the specified context.

  -rc *contextDn*

  This is a required attribute that specifies the root of the subtree.

  options

  -timeout *secs*

  This is an optional attribute that if present, specifies the timeout period in seconds. If not present, or 0, the operation waits indefinitely.

-ra *agreementDn*
The **-ra** *agreementDn* is the DN of the replication agreement. The action is performed for the specified replication agreement. If the -ra option is not specified, the action is performed for all the replication agreements defined under the context.

Example:

```
idsldapexop -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*> : proxy backend server resume role extended operation. This extended operation enables the proxy server to resume the configured role of a back-end server in the distributed directory environment.

  -type {all | partition <*partitionName*> | **server** <*serverName*> | **serverinapartition** <*serverName*> <*partitionName*>}

  options

  all resumes roles for all back-end servers

  partition <*partitionName*>
  resumes the role of all back-end servers in the partition

  server <*serverName*>
  resumes the role of the back-end server for all partitions that it is in

  serverinapartition <*serverName*> <*partitionName*>
  resumes the role of the back-end server in the specified partition

  Example:

  ```
  ldapexop -op resumerole -type all
  ```

- **stopserver**: stop the IBM Tivoli Directory Server

  Example:

  ```
  idsldapexop -D <admindn> -w <adminpw> -op stopserver
  ```

- **unbind {-dn**<*specificDN*> | **-ip**<*sourceIP*> | **-dn**<*specificDN*> **-ip**<*sourceIP*> | **all}**: disconnect connections based on DN, IP, DN/IP or disconnect all connections. All connections without any operations and all connections with operations on the work queue are ended immediately. If a worker is currently working on a connection, it is ended as soon as the worker completes that one operation.

  -dn<*specificDN*>
  Issues a request to end a connection by DN only. This request results in the purging of all the connections bound on the specified DN.

  -ip<*sourceIP*>
  Issues a request to end a connection by IP only. This request results in the purging of all the connections from the specified IP source.

  -dn<*specificDN*> **-ip**<*sourceIP*>
  Issues a request to end a connection determined by a DN/IP pair. This request results in the purging of all the connections bound on the specified DN and 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 the connection from where this request originated. This attribute cannot be used with the -dn or -ip. attributes

Examples:

```
idsldapexop -D <AdminDN> -w <Adminpw> -op unbind -dn cn=john

idsldapexop -D <AdminDN> -w <Adminpw> -op unbind -ip 9.182.173.43

idsldapexop -D <AdminDN> -w <Adminpw> -op unbind -dn cn=john -ip 9.182.173.43

idsldapexop -D <AdminDN> -w <Adminpw> -op unbind -all
```

- **uniqueattr -a** <*attributeType*>: identify all nonunique values for a particular attribute.

  -a <*attribute*>
  Specify the attribute for which all conflicting values are listed.

  Note: Duplicate values for binary, operational, configuration attributes, and the objectclass attribute are not displayed. These attributes are not supported extended operations for unique attributes.

  Example:

  ```
  idsldapexop -D <AdminDN> -w <Adminpw> -op uniqueattr -a "uid"
  ```

  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
  ```

Notes

If no DN arguments are provided, the **ldapdexop** command waits to read a list of DNs from standard input. To break out of the wait, use Ctrl+C or Ctrl+D.

Security functions

To use the SSL or TLS -related functions associated with this utility, see “SSL, TLS notes” on page 65.

Diagnostics

Exit status is 0 if no errors occur. Errors result in a non-zero exit status and a diagnostic message being written to standard error.

See also

idsldapadd, idsldapchangepwd, idsldapdelete, idsldapmodify, idsldapmodrdn, idsldapsearch

idsldapmodify, ldapmodify, idsldapadd, ldapadd

The LDAP modify-entry and LDAP add-entry tools

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]


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]
```

Description

idsldapmodify is a command-line interface to the ldap_modify and ldap_add library calls. **idsldapadd** is implemented as a renamed version of **idsldapmodify**. When invoked as **idsldapadd**, the **-a** (add new entry) flag is turned on automatically.

idsldapmodify opens a connection to an LDAP server, and binds to the server. You can use **idsldapmodify** to modify or add entries. The entry information is read from standard input or from file through the use of the **-i** option.

To display syntax help for **idsldapmodify** or **idsldapadd**, type

```
idsldapmodify -?
```

or

```
idsldapadd -?
```

Options

-a Add new entries. The default action for **idsldapmodify** is to modify existing entries. If invoked as **idsldapadd**, this flag is always set.

-b Assume that any values that start with a `/' are binary values and that the actual value is in a file whose path is specified in place of the valuer.

-B Specifies that a transaction should be rolled back.

-c Continuous operation mode. Errors are reported, but **idsldapmodify** continues with modifications. Otherwise the default action is to exit after reporting an error.

-C *charset*

Specifies that strings supplied as input to the **idsldapmodify** and **idsldapadd** utilities are represented in a local character set as specified by charset, and must be converted to UTF-8. When the **idsldapmodify** and **idsldapadd** records are received from standard input, the specified charset value is used to convert the attribute values that are designated as strings that is, the attribute types are followed by a single colon. If the records are received from an LDIF file that contains a charset tag, the charset tag in the LDIF file overrides the charset value specified on the command-line. See Appendix B, “IANA character sets supported by platform,” on page 131 for the specific charset values that are supported for each operating system platform. Note that the supported values for charset are the same values 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. This parameter is for use by IBM service personnel. See Chapter 4, “Debugging levels,” on page 127 for additional information on debug levels.

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

Note: **-D** *binddn* **-w** *passwd* does not call bind functions on superuser DNs.

-e <*errorfile*>
Specifies the file to which rejected entries are written. This option requires the -c continuous operation option. If the processing of an entry fails, that entry is written to the reject file and the count of rejected entries is increased. If the input to the **idsldapmodify** or **idsldapadd** command is from a file, when the file has been processed, the number of total entries written to the reject file is given.

-E *token_pw*
Token password to access the crypto device.

-f *file*
Read the entry modification information from an LDIF file instead of from standard input. If an LDIF file is not specified, you must use standard input to specify the update records in LDIF format.

Note: This option is deprecated but still supported.

-g
Specifies not to strip the trailing spaces on attribute values.

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

-h *ldaphost*
Specify an alternate host on which the LDAP server is running.

-i *file*
Read the entry modification information from an LDIF file instead of from standard input. If an LDIF file is not specified, you must use standard input to specify the update records in LDIF format.

-I
Crypto device with key storage using PKCS11.

-j
Specifies that a prepare should not be sent.

-k
Specifies to use server administration control.

This option sends the Server administration control. See the *IBM Tivoli Directory Server Version 6.1 Programming Reference* for information about this control.

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

A default keyring file that is, ldapkey.kdb, and the associated password stash file that is, ldapkey.sth, are installed in the etc directory under IDS_LDAP_HOME, where IDS_LDAP_HOME is the path to the installed LDAP support. IDS_LDAP_HOME varies by operating system platform:

- AIX operating systems - /opt/IBM/ldap/V6.1
- HP-UX operating systems - /opt/IBM/ldap/V6.1

- Linux operating systems - /opt/ibm/ldap/V6.1
- Solaris operating systems - /opt/IBM/ldap/V6.1
- Windows operating systems - <*local_drive*>:\Program Files\IBM\LDAP\V6.1 (This is the default install location. The actual IDS_LDAP_HOME is determined during installation.)

See the *IBM Tivoli Directory Server Version 6.1 Programming Reference* for more information about default key database files, and default Certificate Authorities.

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 certificate authorities (CAs) that are trusted by the client. These types of X.509 certificates are also known as trusted roots. See the *IBM Tivoli Directory Server version 6.1 Administration Guide* for more information about managing an SSL or TLS key database and for information about SSL and certificates. Also see the “Security functions” on page 8.

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

-l Do not replicate the entry.

This option sends the Do not replicate control. See the *IBM Tivoli Directory Server Version 6.1 Programming Reference* for information about this control.

-m *mechanism*
Use ***mechanism*** to specify the SASL mechanism to be used to bind to the server. The ldap_sasl_bind_s() API is used. The **-m** parameter is ignored if **-V 2** is set. If **-m** is not specified, simple authentication is used.

-M Manage referral objects as regular entries.

-n Specify the no operation option to enable you to preview the result of the command you are issuing without actually performing the action on the directory. The changes that would be made are preceded by an exclamation mark and printed to standard output. Any syntax errors that are found in the processing of the input file, before the calling of the functions that perform the changes to the directory, are displayed to standard error. This option is especially useful with the **-v** option for debugging operations, if errors are encountered.

-N *certificatename*
Specify the label associated with the client certificate in the key database file. If the LDAP server is configured to perform server authentication only, a client certificate is not required. If the LDAP server is configured to perform client and server Authentication, a client certificate might be required. ***certificatename*** is not required if a default certificate/private key pair has been designated 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 neither **-Z** nor **-K** is specified.

-O *maxhops*
Specify ***maxhops*** to set the maximum number of hops that the client library takes when chasing referrals. The default hopcount is 10.

-p *ldapport*
Specify an alternate TCP port where the LDAP server is listening. The default LDAP port is 389. If **-p** is not specified and **-Z** is specified, the default LDAP SSL port 636 is used.

-P *keyfilepw*
Specify the key database password. This 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, the password is obtained from the password stash file, and the **-P** parameter is not required. This parameter is ignored if neither **-Z** nor **-K** is specified.

-Q *operation*
Crypto device operation with PKCS11

```
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 Replace existing values by default.

-R Specifies that referrals are not to be automatically followed.

-S *token_label*
Token label of the crypto device.

-t Performs the modify in a transaction.

-U *username*
Specifies the username. This is required with -m DIGEST-MD5 and ignored when any other mechanism is used. The value ***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 Use verbose mode, with many diagnostics written to standard output.

-V Specifies the LDAP version to be used by **idsldapmodify** when it binds to the LDAP server. By default, an LDAP V3 connection is established. To explicitly select LDAP V3, specify **-V 3**. Specify **-V 2** to run as an LDAP V2 application. An application, like **idsldapmodify**, selects LDAP V3 as the preferred protocol by using ldap_init instead of ldap_open.

-w *passwd* **|** **?**
Use ***passwd*** as the password for authentication. Use the ? to generate a password prompt. Using this prompt prevents your password from being visible through the **ps** command.

-x Use FIPS mode processing (SSL/TLS only).

-X *lib_path*
Driver path of the crypto device.

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

-Y Use a secure TLS connection to communicate with the LDAP server. The **-Y** option is only supported when IBM's GSKit, is installed.

-Z Use a secure SSL connection to communicate with the LDAP server. The **-Z** option is only supported when the SSL component entry, as provided by IBM's GSKit, is installed.

Input format

The contents of file (or standard input if no **-i** flag is given on the command line) should conform to the LDIF format.

Alternative input format

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

```
Distinguished Name (DN)

attr=value

[attr=value ...]
```

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

By default, values are added. If the **-r** command line flag is given, the default is to replace existing values with the new one. It is permissible for a given attribute to appear more than once, for example, to add more than one value for an attribute. Also note that you can use a trailing `\\' to continue values across lines and preserve new lines in the value itself.

`attr` should be preceded by a - to remove a value. The = and `value` should be omitted to remove an entire attribute.

`attr` should be preceded by a + to add a value in the presence of the **-r** flag.

Examples

Assuming that the file /tmp/entrymods exists and has the following contents:

```
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 command:

```
idsldapmodify -b -r -i /tmp/entrymods
```

will replace the contents of the Modify Me entry's mail attribute with the value modme@student.of.life.edu, add a title of Grand Poobah, and the contents of the file /tmp/modme.jpeg as a jpegPhoto, and completely remove the description attribute. These same modifications can be performed using the older idsldapmodify input format:

```
cn=Modify Me, o=University of Higher Learning, c=US

      mail=modme@student.of.life.edu

      +title=Grand Poobah

      +jpegPhoto=/tmp/modme.jpeg

      -description
```

and the command:

```
idsldapmodify -b -r -i /tmp/entrymods
```

Assuming that the file /tmp/newentry exists and has the following contents:

```
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
```

the command:

```
 idsldapadd -i /tmp/entrymods
```

adds a new entry for John Doe, using the values from the file /tmp/newentry.

Assuming that the file /tmp/newentry exists and has the contents:

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

changetype: delete
```

the command:

```
 idsldapmodify -i /tmp/entrymods
```

removes John Doe's entry.

Notes

If entry information is not supplied from file through the use of the **-i** option, the **idsldapmodify** command will wait to read entries from standard input. To break out of the wait, use Ctrl+C or Ctrl+D.

Security functions

To use the SSL or TLS -related functions associated with this utility, see “SSL, TLS notes” on page 65.

Diagnostics

Exit status is 0 if no errors occur. Errors result in a non-zero exit status and a diagnostic message being written to standard error.

See also

idsldapchangepwd, idsldapdelete, idsldapexop, idsldapmodrdn, idsldapsearch

idsldapmodrdn, ldapmodrdn

The LDAP modify-entry RDN tool

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] [dn newrdn | [-i file]]
```

Description

idsldapmodrdn is a command-line interface to the ldap_rename library call.

idsldapmodrdn opens a connection to an LDAP server, binds, modifies the RDN of an entry and can change the parent of the entry. The entry information is read from standard input, from a file through the use of the **- i** option, or from the command-line pair dn, rdn, or the newSuperior option.

See LDAP Distinguished Names for information about RDNs (Relative Distinguished Names) and DNs (Distinguished Names).

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

```
idsldapmodrdn -?
```

Options

-c
Continuous operation mode. Errors are reported, but **idsldapmodrdn** continues with modifications. Otherwise the default action is to exit after reporting an error.

-C *charset*
Specifies that the strings supplied as input to the **idsldapmodrdn** 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. See Appendix B, “IANA character sets supported by platform,” on page 131 for the specific charset values that are supported for each operating

system platform. Note that the supported values for charset are the same values 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. This parameter is for use by IBM service personnel. See Chapter 4, “Debugging levels,” on page 127 for additional information on debug levels.

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

-E *token_pw*
Token password to access the crypto device.

-f *file* Read entry modification information from specified file.

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

-h *ldaphost*
Specify an alternate host on which the LDAP server is running.

-i *file* Read the entry modification information from file instead of from standard input or the command-line (by specifying rdn and newrdn). Standard input can be supplied from a file, as well (″< file″).

-I Crypto device with key storage using PKCS11.

-k Specifies to use server administration control.

This option sends the Server administration control. See the *IBM Tivoli Directory Server Version 6.1 Programming Reference*.

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

A default keyring file (that is, ldapkey.kdb) and the associated password stash file (that is, ldapkey.sth) are installed in the etc directory under IDS_LDAP_HOME, where IDS_LDAP_HOME is the path to the installed LDAP support. IDS_LDAP_HOME varies by operating system platform:

- AIX operating systems - /opt/IBM/ldap/V6.1
- HP-UX operating systems - /opt/IBM/ldap/V6.1
- Linux operating systems - /opt/ibm/ldap/V6.1
- Solaris operating systems - /opt/IBM/ldap/V6.1
- Windows operating systems - <*local_drive*>:\Program Files\IBM\LDAP\V6.1 (This is the default install location. The actual IDS_LDAP_HOME is determined during installation.)

See the *IBM Tivoli Directory Server Version 6.1 Programming Reference* for more information about default key database files, and default Certificate Authorities.

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 certificate authorities (CAs) that are trusted by the client. These types of X.509 certificates are also known as trusted roots. See the *IBM Tivoli Directory Server version 6.1 Administration Guide* for more information about managing an SSL or TLS key database and for information about SSL and certificates. Also see the "Security functions" on page 8.

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

-l Do not replicate the entry.

This option sends the Do not replicate control. See the *IBM Tivoli Directory Server Version 6.1 Programming Reference* for information about this control.

-m *mechanism*
Use ***mechanism*** to specify the SASL mechanism to be used to bind to the server. The ldap_sasl_bind_s() API is used. The **-m** parameter is ignored if **-V 2** is set. If **-m** is not specified, simple authentication is used.

-M Manage referral objects as regular entries.

-n Show what would be done, but do not modify entries. Useful for debugging in conjunction with **-v**.

-N *certificatename*
Specify the label associated with the client certificate in the key database file. Note that if the LDAP server is configured to perform server authentication only, a client certificate is not required. If the LDAP server is configured to perform client and server Authentication, a client certificate might be required. ***certificatename*** is not required if a default certificate/private key pair has been designated 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 neither **-Z** nor **-K** is specified.

-O *hopcount*
Specify ***hopcount*** to set the maximum number of hops that the client library takes when chasing referrals. The default hopcount is 10.

-p *ldapport*
Specify an alternate TCP port where the LDAP server is listening. The default LDAP port is 389. If not specified and -Z is specified, the default LDAP SSL port 636 is used.

-P *keyfilepw*
Specify the key database password. This password is required to access the encrypted information in the key database file (which may include one or more private keys). If a password stash file is associated with the key database file, the password is obtained from the password stash file, and the **-P** parameter is not required. This parameter is ignored if neither **-Z** nor **-K** is specified.

-Q *operation*
Crypto device operation with PKCS11

```
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 Remove old RDN values from the entry. Default action is to keep old values.

-R Specifies that referrals are not to be automatically followed.

-s *newSuperior*
Specifies the DN of the new superior entry under which the renamed entry is relocated. The newSuperior argument may be the zero-length string (-s "").

-S *token_label*
Token label of the crypto device.

-U *username*
Specifies the username. This is required with -m DIGEST-MD5 and ignored when any other mechanism is used. The value ***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 Use verbose mode, with many diagnostics written to standard output.

-V Specifies the LDAP version to be used by **idsldapmodrdn** when it binds to the LDAP server. By default, an LDAP V3 connection is established. To explicitly select LDAP V3, specify **-V 3**. Specify **-V 2** to run as an LDAP V2 application. An application, like **idsldapmodrdn**, selects LDAP V3 as the preferred protocol by using ldap_init instead of ldap_open.

-w *passwd* **| ?**
Use ***passwd*** as the password for authentication. Use the ? to generate a password prompt. Using this prompt prevents your password from being visible through the **ps** command.

-x Use FIPS mode processing (SSL/TLS only).

-X *lib_path*
Driver path of the crypto device.

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

-Y Use a secure TLS connection to communicate with the LDAP server. The **-Y** option is only supported when IBM's GSKit, is installed.

-Z Use a secure SSL connection to communicate with the LDAP server. The **-Z** option is only supported when the SSL component entry, as provided by IBM's GSKit, is installed.

dn newrdn
See the following section, "Input format for dn newrdn" for more information.

Input format for dn newrdn

If the command-line arguments *dn* and *newrdn* are given, *newrdn* replaces the RDN of the entry specified by the DN, *dn*. Otherwise, the contents of file (or standard input if no **- i** flag is given) consist of one or more entries:

```
Distinguished Name (DN)

Relative Distinguished Name (RDN)
```

One or more blank lines may be used to separate each DN and RDN pair.

Examples

Assuming that the file /tmp/entrymods exists and has the contents:

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

the command:

```
idsldapmodrdn -r -i /tmp/entrymods
```

changes the RDN of the `Modify Me` entry from `Modify Me` to `The New Me` and the old cn, `Modify Me` is removed.

The command:

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

changes the RDN of the Modify Me entry from Modify Me to The New Me. The entry is moved from underneath the University of Life entry to underneath the IBM entry.

Notes

If entry information is not supplied from file through the use of the **-i** option (or from the command-line pair *dn* and *rdn*), the **idsldapmodrdn** command waits to read entries from standard input. To break out of the wait, use Ctrl+C or Ctrl+D.

Security functions

To use the SSL or TLS -related functions associated with this utility, see “SSL, TLS notes” on page 65.

Diagnostics

Exit status is 0 if no errors occur. Errors result in a non-zero exit status and a diagnostic message being written to standard error.

See also

idsldapadd, idsldapchangepwd, idsldapdelete, idsldapexop, idsldapmodify, idsldapsearch

idsldapsearch, ldapsearch

The LDAP search tool and sample program

Synopsis

```
idsldapsearch | ldapsearch [-a deref] [-A] [-b searchbase] [-B] [-c pattern][-C charset]
[-d debuglevel] [-D binddn] [-E token_pw] [-f file] [-F sep] [-G realm] [-h ldaphost]
[-i file] [-I] [-j limit] [-J limit] [-k] [-K keyfile] [-l timelimit] [-L] [-m mechanism]
[-M] [-n] [-N certificatename] [-o attr_type] [-O maxhops] [-p ldapport] [-P keyfilepw]
```

```
[-q pagesize] [-Q operation] [-R] [-s scope ] [-S token_label] [-t] [-T seconds]
[-U username] [-v] [-V version] [-w passwd | ?] [-x] [-X lib_path] [-y proxydn]
[-Y] [-z sizelimit] [-Z] filter [-9 p] [-9 s] [attrs...]
```

Description

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

idsldapsearch opens a connection to an LDAP server, binds, and performs a search using the filter. The filter should conform to the string representation for LDAP filters (see the ldap_search information in the *IBM Tivoli Directory Server Version 6.1 Programming Reference* for more information on filters).

If **idsldapsearch** finds one or more entries, the attributes specified by attrs are retrieved and the entries and values are printed to standard output. If no attrs are listed, all attributes are returned.

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

Note:

- The search filter size limit is set at 4 KB in the ldapsearch.c file. Any filter size larger than 4 KB will be rejected by the **idsldapsearch** utility. If you want to change ldapsearch.c to handle a filter larger than 4 KB (even though an altered version of **idsldapsearch** will not be supported), then change the following line in ldapsearch.c:

  ```
  #define FILTERSIZE 4096
  ```

  to something like the following:

  ```
  #define FILTERSIZE 16000
  ```

  You must recompile ldapsearch.c for these changes to take effect.
- Entries under cn=configuration are not in DIT and, therefore, will not be returned as search results for null based searches.

Options

-a deref
Specify how aliases dereferencing is done. deref should be one of never, always, search, or find to specify that aliases are never dereferenced, always dereferenced, dereferenced when searching, or dereferenced only when locating the base object for the search. The default is to never dereference aliases.

-A
Retrieve attributes only (no values). This is useful when you just want to see if an attribute is present in an entry and are not interested in the specific values.

-b searchbase
Use searchbase as the starting point for the search instead of the default. If **-b** is not specified, this utility will examine the LDAP_BASEDN environment variable for a searchbase definition. If neither is set, the default base is set to "", which is a null search. A null search returns all the entries in the entire Directory Information Tree (DIT). This search requires a **-s** subtree option. Otherwise, an error message is displayed. Be aware that null based search requests consume a lot of resource.

-B Do not suppress display of non-ASCII values. This is useful when dealing with values that appear in alternate characters sets such as ISO-8859.1. This option is implied by the **-L** option.

-c pattern
Performs a persistent search. The pattern format should be ps:changeType[:changesOnly[:entryChangeControls]], where changeType can be add, delete, modify, moddn, and any. The changesOnly and entryChangeControls parameters are Boolean parameters and can be set to TRUE or FALSE.

Note: When alias dereferencing option is 'find', then only the search base object needs to be de-referenced if it is an alias. This means that even if it is a one-level or sub-tree search, the subordinate alias entries under the base are not expected to be de-referenced. However, if it is a persistent search that is reporting changed entries and a changed entry happens to be an alias, then it is de-referenced even though it is subordinate to the search base.

-C charset
Specifies that strings supplied as input to the idsldapsearch utility are represented in a local character set (as specified by charset). String input includes the filter, the bind DN and the base DN. Similarly, when displaying data, **idsldapsearch** converts data received from the LDAP server to the specified character set. Use "-C charset" to override the default, where strings must be supplied in UTF-8. Also, if the **-C** option and the **-L** option are both specified, input is assumed to be in the specified character set, but 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. This is the case because standard LDIF files only contain UTF-8 (or base-64 encoded UTF-8) representations of string data. See Appendix B, "IANA character sets supported by platform," on page 131 for the specific charset values that are supported for each operating system platform. Note that the supported values for charset are the same values 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. This parameter is for use by IBM service personnel. See Chapter 4, "Debugging levels," on page 127 for additional information on debug levels.

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

-e Display the LDAP library version information and exits.

-E *token_pw*
Token password to access the crypto device.

-f *file* Perform sequence of searches using filters in *file*. "%s" must be substituted for the filter.

-F sep Use sep as the field separator between attribute names and values. The default separator is `=', unless the **-L** flag has been specified, in which case this option is ignored.

-G *realm*

Specify the name of the realm. When used with the -m DIGEST-MD5, the value is passed to the server during the bind.

-h ldaphost

Specify an alternate host on which the LDAP server is running.

-i file

Read a series of lines from file, performing one LDAP search for each line. In this case, the filter given on the command line 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 the command, **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.

The command performs a search of the subtree **o=sample** for each of the filters beginning with **cn=*Z**. When that search is completed, the search begins for the next filter **cn=*Z*** and so forth 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

Crypto device with key storage using PKCS11.

-j limit

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

-J limit

Maximum number of total attribute values that can be returned for an entry. The default value is 0 which means unlimited.

-k

Use server administration control on bind.

-K keyfile

Specify the name of the SSL or TLS key database file (with default extension of ″kdb″). If the key database file is not in the current directory, specify the fully-qualified key database filename. If a key database filename is not specified, this utility will first look for the presence of the SSL_KEYRING environment variable with an associated filename. If the SSL_KEYRING environment variable is not defined, the default keyring file will be used, if present.

A default keyring file (that is, ldapkey.kdb) and the associated password stash file (that is, ldapkey.sth) are installed in the etc directory under IDS_LDAP_HOME, where IDS_LDAP_HOME is the path to the installed LDAP support. IDS_LDAP_HOME varies by operating system platform:

- AIX operating systems - /opt/IBM/ldap/V6.1
- HP-UX operating systems - /opt/IBM/ldap/V6.1
- Linux operating systems - /opt/ibm/ldap/V6.1

- Solaris operating systems - /opt/IBM/ldap/V6.1
- Windows operating systems - <*local_drive*>:\Program Files\IBM\LDAP\V6.1 (This is the default install location. The actual IDS_LDAP_HOME is determined during installation.)

See the "Default Keyring and Password" section of the LDAP_SSL API in the *IBM Tivoli Directory Server Version 6.1 Programming Reference* for more information about default key database files, and default Certificate Authorities.

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 certificate authorities (CAs) that are trusted by the client. These types of X.509 certificates are also known as trusted roots. See the *IBM Tivoli Directory Server version 6.1 Administration Guide* for more information about managing an SSL or TLS key database and for information about SSL and certificates. Also see the "Security functions" on page 54 below and LDAP SSL or TLS APIs for more information about SSL and certificates.

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

-l timelimit
Wait at most timelimit seconds for a search to complete.

-L Display search results in LDIF format. This option also turns on the **-B** option, and causes the **-F** option to be ignored.

-m mechanism
Use mechanism to specify the SASL mechanism to be used to bind to the server. The ldap_sasl_bind_s() API will be used. The **-m** parameter is ignored if **-V 2** is set. If **-m** is not specified, simple authentication is used.

-M Manage referral objects as regular entries.

-n Show what would be done, but do not modify entries. Useful for debugging in conjunction with **-v**.

-N certificatename
Specify the label associated with the client certificate in the key database file.

Note: If the LDAP server is configured to perform server authentication only, a client certificate is not required. If the LDAP server is configured to perform client and server Authentication, a client certificate might be required. *certificatename* is not required if a default certificate/private key pair has been designated 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 neither **-Z** nor **-K** is specified.

-o *attr_type*
To specify an attribute to use for sort criteria of search results, you can use the -o (order) parameter. You can use multiple -o parameters to further define the sort order. In the following example, the search results are sorted first by surname (sn), then by given name, with the given name (givenname) being sorted in reverse (descending) order as specified by the prefixed minus sign (-):

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

Thus, the syntax of the sort parameter is as follows:

```
[-]<attribute name>[:<matching rule OID>]
```

where

- `attribute name` is the name of the attribute you want to sort by.
- `matching rule OID` is the optional OID of a matching rule that you want to use for sorting.
- The minus sign (-) indicates that the results must be sorted in reverse order.
- The criticality is always critical.

The default **idsldapsearch** operation is not to sort the returned results.

This option sends the Sorted search results control. See "Sorted search results control" in the *IBM Tivoli Directory Server Version 6.1 Programming Reference*.

-O maxhops
Specify maxhops to set the maximum number of hops that the client library takes when chasing referrals. The default hopcount is 10.

-p ldapport
Specify an alternate TCP port where the LDAP server is listening. The default LDAP port is 389. If not specified and -Z is specified, the default LDAP SSL port 636 is used.

-P keyfilepw
Specify the key database password. This password is required to access the encrypted information in the key database file (which may include one or more private keys). If a password stash file is associated with the key database file, the password is obtained from the password stash file, and the **-P** parameter is not required. This parameter is ignored if neither **-Z** nor **-K** is specified.

-q *pagesize*
To specify paging of search results, two new parameters can be used: -q (query page size), and -T (time between searches in seconds). In the following example, the search results return a page (25 entries) at a time, every 15 seconds, until all the results for that search are returned. The **idsldapsearch** client handles all connection continuation for each paged results request for the life of the search operation.

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

If the -v (verbose) parameter is specified, **idsldapsearch** lists how many entries have been returned so far, after each page of entries returned from the server, for example, **30 total entries have been returned.**

Multiple -q parameters are enabled such that you can specify different page sizes throughout the life of a single search operation. In the following example, the first page is 15 entries, the second page is 20 entries, and the third parameter ends the paged result/search operation:

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

In the following example, the first page is 15 entries, and all the rest of the pages are 20 entries, continuing with the last specified **-q** value until the search operation completes:

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

The default **idsldapsearch** operation is to return all entries in a single request. No paging is done for the default **idsldapsearch** operation.

This option sends the Paged search results control. See ″Paged search results control″ in the *IBM Tivoli Directory Server Version 6.1 Programming Reference*.

-Q *operation*

Crypto device operation with PKCS11

```
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 that referrals are not to be automatically followed.

-s scope

Specify the scope of the search. scope should be one of base, one, or sub to specify a base object, one-level, or subtree search. The default is sub.

Note: If you specify a null search, either by not specifying a **-b** option or specifying **-b** ″″, you must the **-s** option. The default scope is disabled for a null search.

-S *token_label*

Token label of the crypto device.

-t Write retrieved values to a set of temporary files. This is useful for dealing with non-ASCII values such as jpegPhoto or audio.

-T *seconds*

Time between searches (in seconds). The **-T** option is only supported when the **-q** option is specified.

-U *username*

Specifies the username. This is required with -m DIGEST-MD5 and ignored when any other mechanism is used. The value ***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 Use verbose mode, with many diagnostics written to standard output.

-V Specifies the LDAP version to be used by idsldapmodify when it binds to the LDAP server. By default, an LDAP V3 connection is established. To explicitly select LDAP V3, specify ″-V 3″. Specify ″-V 2″ to run as an LDAP V2 application. An application, like idsldapmodify, selects LDAP V3 as the preferred protocol by using ldap_init instead of ldap_open.

-w *passwd* **| ?**

Use ***passwd*** as the password for authentication. Use the ? to generate a password prompt. Using this prompt prevents your password from being visible through the **ps** command.

-x Use FIPS mode processing (SSL/TLS only).

-X *lib_path*

Driver path of the crypto device.

-y *proxydn*

Specifies the DN to be used for proxied authorization.

-Y Use a secure TLS connection to communicate with the LDAP server. The **-Y** option is only supported when IBM’s GSKit, is installed.

-z sizelimit
Limit the results of the search to at most sizelimit entries. This makes it possible to place an upper bound on the number of entries that are returned for a search operation.

-Z Use a secure SSL connection to communicate with the LDAP server. The **-Z** option is only supported when the SSL component entry, as provided by IBM’s GSKit, is installed.

-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 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 is used to specify 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 ending 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 using filters.

See ″RFC 2254, A String Representation of LDAP Search Filters″ for a more complete description of allowable filters.

attrs A whitespace-separated list of attribute type names to be returned for each entry that matches the search filter. Individual attribute type names may be specified. Additionally, the following special notations may be used:

***** An asterisk in the list indicates all attribute types other than operational attributes should be returned.

1.1 Specifies to return no attributes and is used to request that a search return only the matching distinguished names

+ A plus sign indicates that the operational attributes should be returned.

+ibmaci
Returns the access control related operational attributes.

+ibmentry
Returns the operational attributes every entry contains, such as creatorsName, create_Timestamp, and modifiersname to name a few.

+ibmrepl
Returns operational attributes related to replication.

+ibmpwdpolicy
Returns operational attributes related to password policy.

++ Indicates that ALL operational attributes should be included, even those considered expensive to return, such as ibm-allGroups and ibm-replicationPendingChanges.

++ibmaci
Includes ALL access control related operational attributes.

++ibmentry
Includes ALL operational attributes every entry contains, such as numsubordinates, ibm-entryChecksum.

++ibmrepl
Includes ALL operational attributes related to replication.

++ibmpwdpolicy
Includes ALL operational attributes related to password policy.

Output format

If one or more entries are found, each entry is written to standard output in the 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, it will be 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 given, only the ″attributename″ part is written.

Examples

The following command:

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

performs a subtree search (using the default search base) for entries with a commonName of ″john doe″. The commonName and telephoneNumber values is retrieved and printed to standard output. The output might look something like this if 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
```

The command:

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

performs a subtree search using the default search base for entries with user ID of ″jed″. The jpegPhoto and audio values are retrieved and written to temporary files. The output might look like this if 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
```

This command:

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

will perform a one-level search at the c=US level for all organizations whose organizationName begins with university. Search results will be displayed in the LDIF format (see LDAP Data Interchange Format). The organizationName and description attribute values will be retrieved and printed to standard output, resulting in output similar to this:

```
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

 ...
```

This command:

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

performs 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 something like this:

```
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
```

This command:

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

performs a one- level search at the o=sample level and returns the number of entries that would be returned by a one-level search. The count returned does not take into account whether the bound client has authority to read any of the entries that are included in the count, other than the entry containing this value. If you have loaded the example file **sample.ldif** and issued the specified command with the numSubordinates attribute, the result is:

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

The following examples explain the usage of –c option that is used to perform persistent search:

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

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

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

The search command above 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.

The following command displays all password policy attributes for a given entry:

```
ldapsearch -s base -D <adminDN> -w <adminPW> -b "uid=user1,cn=users,o=ibm"
"objectclass=*" +ibmpwdpolicy
```

Security functions

To use the SSL or TLS -related functions associated with this utility, see “SSL, TLS notes” on page 65.

Diagnostics

Exit status is 0 if no errors occur. Errors result in a non-zero exit status and a diagnostic message being written to standard error.

See also

idsldapadd, idsldapchangepwd, idsldapdelete, idsldapexop, idsldapmodify, idsldapmodrdn

idsldaptrace, ldaptrace

The administration tracing utility. This utility is to be used in conjunction with IBM support to solve specific problems.

Notes:

1. Only the administrator or a member of the administrative group can use this utility.
2. Using **idsldaptrace** consumes resources and affects the performance of the server.

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] -?
```

Description

The administration tracing utility, **idsldaptrace**, is used to dynamically activate or deactivate tracing of the Directory Server. This extended operation can also be used to set the message level and specify the name of the file to the output is written. If LDAP trace facility (ldtrc) options are requested, they must be preceded by --.

To display syntax help for **idsldaptrace**, type: `idsldaptrace -?`

Note: While the **idsldaptrace** utility can be used with SSL or TLS , only the simple bind mechanism is supported.

Options

-a *port* Specifies an alternate TCP port where IBM Administration Daemon (idsdiradm), not the Directory Server, is listening. The default port is 3538. If not specified and **-Z** is specified, the default SSL port 3539 is used.

-d *debugLevel*
Debug this program.

-D *adminDn*
Bind DN.

-E *token_pw*
Token password to access the crypto device.

-h *ldaphost*
Specify an alternate host on which the Directory Server and the Administration Daemon are running.

-I Crypto device with key storage using PKCS11.

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

A default keyring file that is, ldapkey.kdb, and the associated password stash file that is, ldapkey.sth, are installed in the etc directory under IDS_LDAP_HOME, where IDS_LDAP_HOME is the path to the installed LDAP support. IDS_LDAP_HOME varies by operating system platform:

- AIX operating systems - /opt/IBM/ldap/V6.1
- HP-UX operating systems - /opt/IBM/ldap/V6.1
- Linux operating systems - /opt/ibm/ldap/V6.1
- Solaris operating systems - /opt/IBM/ldap/V6.1
- Windows operating systems - <*local_drive*>:\Program Files\IBM\LDAP\V6.1 (This is the default install location. The actual IDS_LDAP_HOME is determined during installation.)

See the *IBM Tivoli Directory Server Version 6.1 Programming Reference* for more information about default key database files, and default Certificate Authorities.

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 certificate authorities (CAs) that are trusted by the client. These types of X.509 certificates are also known as trusted roots. See the *IBM Tivoli Directory Server version 6.1 Administration Guide* for more information about managing an SSL or TLS key database and for information about SSL and certificates. Also see the "Security functions" on page 8.

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

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

on Turns on the tracing facility. You can specify any of the following ldtrc options preceded by an extra -.

- [-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 sever 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>] trace only the specified object.
- [-perf] trace only performance records.

Note: The tracing facility must be on for server data to be traced.

off Turns off the tracing facility.

clr Clears the existing trace buffer.

chg The trace must be active before you can use the **chg** option to change the values for the following **ldtrc** 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 sever errors (maxSevereErrors) is reached.
- [-this <thisPointer>] trace 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 is an example of the information that the **info** parameter gives:

```
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
cpid    to trace       : all
```

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

dump Dumps the trace information to a file. This information includes process flow data as well as server debug messages. You can specify the name of the destination file where you want to dump the trace. The default destination files is:

For AIX, Linux, Solaris, and HP-UX systems:
/var/ldap/ibmslapd.trace.dump.

For Windows-based systems:
<*installationpath*>\var\ibmslapd.trace.dump

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

-m <*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. This parameter is for use by IBM service personnel. See Chapter 4, "Debugging levels," on page 127 for additional information on debug levels.

-N *certificatename*
Specify the label associated with the client certificate in the key database file. If the LDAP server is configured to perform server authentication only, a client certificate is not required. If the LDAP server is configured to perform client and server Authentication, a client certificate might be required. ***certificatename*** is not required if a default certificate/private key pair has been designated 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 neither **-Z** nor **-K** is specified.

-o debugfile
Specifies the output file name for the server debug messages.

-p port
Specify an alternate TCP port where the LDAP server is listening. The default LDAP port is 389. If not specified and **-Z** is specified, the default LDAP SSL port 636 is used.

-P *keyfilepw*
Specify the key database password. This password is required to access the encrypted information in the key database file, which may include one or more private keys. If a password stash file is associated with the key database file, the password is obtained from the password stash file, and the **-P** parameter is not required. This parameter is ignored if neither **-Z** nor **-K** is specified.

-Q *operation*
Crypto device operation with PKCS11

```
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*
Token label of the crypto device.

-t [start|stop]

start Starts the collection of server trace data.

stop Stops the collection of server trace data.

-v Specifies to run in verbose mode.

-w *adminPW* **| ?**
Use ***adminPW*** as the password for authentication. Use the ? to generate a password prompt. Using this prompt prevents your password from being visible through the **ps** command.

-x Use FIPS mode processing (SSL/TLS only).

-X *lib_path*
Driver path of the crypto device.

-Z Use a secure LDAP connection (SSL).

-? Displays the syntax format.

Examples

To turn the ldtrc facility on and start the server trace with a 2M trace buffer, issue the command:

```
idsldaptrace -h <hostname> -D <adminDN> -w <adminpw> -l on -t start -- -I 2000000
```

To stop the server trace, issue the command:

```
idsldaptrace -h <hostname> -D <adminDN> -w <adminpw> -t stop
```

To turn off the ldtrc facility, issue the command:

```
idsldaptrace -h <hostname> -D <adminDN> -w <adminpw> -l off
```

See also

"ldtrc" on page 124

idslink

The **idslink** command creates links to LDAP client and server command-line utilities. This utility is installed with the client package. Links for client and server utilities are not set automatically during installation. However, you can use the **idslink** utility to set the links to command-line utilities such as **idsldapmodify** and **idsldapadd** and libraries such as libibmldap. These links point to the location where the IBM Tivoli Directory Server utilities and libraries reside: *installpath*/bin, *installpath*/sbin, and *installpath*/lib. (*installpath* is the directory where IBM Tivoli Directory Server is installed). The syntax for the **idslink** command is as follows

Synopsis

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

where

-h Displays usage help for the command.

-v Displays version information about the command.

-n Pretend option. Displays the links that will be set if you run the command with the options you specify. If specified, you must also specify one or more of the following options: **-i**, **-g**, or **-l**. After running the command with this option, check the /var/idsldap/V6.1/idslink.preview file, which will contain any conflicts that were found.

-i Creates links only for client command utilities that begin with 'ids'. For example, creates the link from /usr/bin/idsldapsearch to /opt/ibm/ldap/V6.1/bin/idsldapsearch

-g Creates links only for client command utilities that do not begin with 'ids'. For example, creates the link from /usr/bin/ldapsearch to /opt/ibm/ldap/V6.1/bin/ldapsearch.

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

-s *mode*

Creates links for server command-line utilities only. *mode* can be **base** to establish links for the base server code to be used by the proxy or full server or **fullsrv** if the directory server instance is a full server.

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

-f Force option. Specifies to override existing files or links, and back up any conflicts. For example, /usr/bin/ldapsearch.

If you use the force option, each conflicting link is backed up into a subdirectory with the same name as the file, directory, or link that had the conflict. For example, a conflict for the /usr/bin/ldapsearch command is backed up in a subdirectory called /usr/bin/V6.1_idslink_bkup_*timestamp*, where *timestamp* is the date and time 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.

Links created by idslink

The following sections show links that are created by the **idslink** command.

Note: /opt/*ibmdir*/ldap/V6.1/ is /opt/IBM/ldap/V6.1/ on AIX, Solaris, and HP-UX systems. On Linux systems, /opt/*ibmdir*/ldap/V6.1/ is /opt/ibm/ldap/V6.1/

Client commands

Links created when -g option is specified: Set of links for client commands (that do not begin with 'ids') for the base client

/usr/bin/ldapsearch —> /opt/*ibmdir*/ldap/V6.1/bin/ldapsearch
/usr/bin/ldapadd —> /opt/*ibmdir*/ldap/V6.1/bin/ldapadd
/usr/bin/ldapmodify —> /opt/*ibmdir*/ldap/V6.1/bin/ldapmodify
/usr/bin/ldapdelete —> /opt/*ibmdir*/ldap/V6.1/bin/ldapdelete
/usr/bin/ldapmodrdn —> /opt/*ibmdir*/ldap/V6.1/bin/ldapmodrdn
/usr/bin/ldapchangepwd —>
  /opt/*ibmdir*/ldap/V6.1/bin/ldapchangepwd
/usr/bin/ldaptrace —> /opt/*ibmdir*/ldap/V6.1/bin/ldaptrace
/usr/bin/ldapexop —> /opt/*ibmdir*/ldap/V6.1/bin/ldapexop
/usr/bin/ibmdirctl —> /opt/*ibmdir*/ldap/V6.1/bin/ibmdirctl

Links created when -i option is specified: Set of links for client commands (that begin with 'ids') for the base client

/usr/bin/idsldapsearch —> /opt/*ibmdir*/ldap/V6.1/bin/idsldapsearch
/usr/bin/idsldapadd —> /opt/*ibmdir*/ldap/V6.1/bin/idsldapadd
/usr/bin/idsldapmodify —> /opt/*ibmdir*/ldap/V6.1/bin/idsldapmodify
/usr/bin/idsldapdelete —> /opt/*ibmdir*/ldap/V6.1/bin/idsldapdelete
/usr/bin/idsldapmodrdn —>
/opt/*ibmdir*/ldap/V6.1/bin/idsldapmodrdn
/usr/bin/idsldapchangepwd —>
/opt/ibm/ldap/V6.1/bin/idsldapchangepwd
/usr/bin/idsldaptrace —> /opt/*ibmdir*/ldap/V6.1/bin/idsldaptrace
/usr/bin/idsldapexop —> /opt/*ibmdir*/ldap/V6.1/bin/idsldapexop
/usr/bin/idsdirctl —> /opt/*ibmdir*/ldap/V6.1/bin/idsdirctl

Client libraries

Note: *XX* is a library extension such as .so, .a, or .sl

Links created when -l 32 option is specified

The following groups or sets of links are created when the **-l** *bits* option is specified and *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/V6.1/lib/libidsldap.*XX*
/usr/lib/libidsldapstatic.*XX* —>
/opt/*ibmdir*/ldap/V6.1/lib/libidsldapstatic.*XX*
/usr/lib/idsldap_plugin_sasl_digest-md5.*XX* —>
/opt/*ibmdir*/ldap/V6.1/lib/idsldap_plugin_sasl_digest-md5.*XX*

Operating system-specific links:

/usr/lib/idsldap_plugin_ibm_gsskrb.*XX* —>
/opt/*ibmdir*/ldap/V6.1/lib/idsldap_plugin_ibm_gsskrb.*XX*
(AIX only, Kerberos library file)
/usr/lib/libidsldif.*XX* —>
/opt/*ibmdir*/ldap/V6.1/lib/libidsldif.*XX*
(Linux and HP_UX only)

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

Common links:

/usr/lib/libldap.*XX* —>
/opt/*ibmdir*/ldap/V6.1/lib/libidsldap.XX
/usr/lib/libibmldap.*XX* —>
/opt/*ibmdir*/ldap/V6.1/lib/libidsldap.*XX*
/usr/lib/libibmldapstatic.*XX* —>
/opt/*ibmdir*/ldap/V6.1/lib/libidsldapstatic.*XX*
/usr/lib/libldapiconv.*XX* —>
/opt/*ibmdir*/ldap/V6.1/lib/libidsldapiconv.*XX*
/usr/lib/ldap_plugin_sasl_digest-md5.*XX* —>
/opt/*ibmdir*/ldap/V6.1/lib/idsldap_plugin_sasl_digest-md5.*XX*

Operating system-specific links:

```
/usr/lib/ldap_plugin_ibm_gsskrb.XX —>
     /opt/ibmdir/ldap/V6.1/lib/idsldap_plugin_ibm_gsskrb.XX
                          (AIX only, Kerberos library file)
/usr/lib/libldif.XX —>
     /opt/ibmdir/ldap/V6.1/lib/libidsldif.XX
                          (Linux and HP_UX only)
```

Links created when -l 64 option is specified

The following groups or sets of links are created when the **-l** *bits* option is specified and *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/V6.1/lib64/libidsldap.XX
/usr/lib/libidsldapstatic64.XX —>
/opt/ibmdir/ldap/V6.1/lib64/libidsldapstatic.XX
/usr/lib/idsldap_plugin_sasl_digest-md5_64.XX —>
/opt/ibmdir/ldap/V6.1/lib64/idsldap_plugin_sasl_digest-md5.XX
```

Operating system-specific links:

```
/usr/lib/idsldap_plugin_ibm_gsskrb_64.XX —>
/opt/ibmdir/ldap/V6.1/lib64/idsldap_plugin_ibm_gsskrb.XX
                          (AIX only, Kerberos library file)
/usr/lib/libidsldif64.XX —>
/opt/ibmdir/ldap/V6.1/lib64/libidsldif.XX
                          (Linux and HP_UX only)
```

Client libraries: Set of links with '64' in name for 64-bit client package (backward compatibility support)

Common links:

```
/usr/lib/libldap64.XX —>
     /opt/ibmdir/ldap/V6.1/lib64/libidsldap.XX
/usr/lib/libibmldap64.XX —>
     /opt/ibmdir/ldap/V6.1/lib64/libidsldap.XX
/usr/lib/libibmldapstatic64.XX —>
     /opt/ibmdir/ldap/V6.1/lib64/libidsldapstatic.XXX
/usr/lib/libldapiconv64.XX —>
     /opt/ibmdir/ldap/V6.1/lib64/libidsldapiconv.XX
/usr/lib/ldap_plugin_sasl_digest-md5_64.XX —>
/opt/ibmdir/ldap/V6.1/lib64/idsldap_plugin_sasl_digest-md5.XX
```

Operating system-specific links:

```
/usr/lib/ldap_plugin_ibm_gsskrb_64.XX —>
     /opt/ibmdir/ldap/V6.1/lib64/idsldap_plugin_ibm_gsskrb.XX
                               (AIX only, Kerberos library file)
/usr/lib/libldif64.XX —>
     /opt/ibmdir/ldap/V6.1/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/V6.1/lib64/libidsldap.XX
/usr/lib/lib64/libidsldapstatic.XX —>
     /opt/ibmdir/ldap/V6.1/lib64/libidsldapstatic.XX
/usr/lib/lib64/idsldap_plugin_sasl_digest-md5.XX —>
/opt/ibmdir/ldap/V6.1/lib64/idsldap_plugin_sasl_digest-md5.XX
```

Operating system-specific links:

```
/usr/lib/lib64/idsldap_plugin_ibm_gsskrb.XX —>
     /opt/ibmdir/ldap/V6.1/lib64/idsldap_plugin_ibm_gsskrb.XX
                                   (AIX only, Kerberos library file)
/usr/lib/lib64/libidsldif.XX —>
     /opt/ibmdir/ldap/V6.1/lib64/libidsldif.XX
                                   (Linux and HP_UX only)
```

Client libraries: Set of links without '64' in name for 64-bit client package (backward compatibility support)

Common links:

```
/usr/lib/lib64/libldap.XX —>
       /opt/ibmdir/ldap/V6.1/lib64/libidsldap.XX
/usr/lib/lib64/libibmldap.XX —>
     /opt/ibmdir/ldap/V6.1/lib64/libidsldap.XX
/usr/lib/lib64/libibmldapstatic.XX —>
     /opt/ibmdir/ldap/V6.1/lib64/libidsldapstatic.XX
/usr/lib/lib64/libldapiconv.XX —>
     /opt/ibmdir/ldap/V6.1/lib64/libidsldapiconv.XX
/usr/lib/lib64/ldap_plugin_sasl_digest-md5.XX —>
/opt/ibmdir/ldap/V6.1/lib64/idsldap_plugin_sasl_digest-md5.XX
```

Operating system-specific links:

```
/usr/lib/lib64/ldap_plugin_ibm_gsskrb.XX —>
     /opt/ibmdir/ldap/V6.1/lib64/idsldap_plugin_ibm_gsskrb.XX
                                 (AIX only, Kerberos library file)
/usr/lib/lib64/libldif.XX —>
     /opt/ibmdir/ldap/V6.1/lib64/libidsldif.XX
                                 (Linux and HP_UX only)
```

Server commands

Links created when -s base option is specified: Set of links for server commands for base server package

```
/usr/bin/slapd —> /opt/ibmdir/ldap/V6.1/sbin/slapd (5.2 legacy)
/usr/bin/ibmslapd —>
     /opt/ibmdir/ldap/V6.1/sbin/ibmslapd (5.2 legacy)
/usr/bin/idsslapd —>
     /opt/ibmdir/ldap/V6.1/sbin/idsslapd
/usr/bin/ibmdiradm —>
     /opt/ibmdir/ldap/V6.1/sbin/ibmdiradm (5.2 legacy)
/usr/bin/idsdiradm —> /opt/ibmdir/ldap/V6.1/sbin/idsdiradm
/usr/bin/ldtrc —> /opt/ibmdir/ldap/V6.1/sbin/ldtrc (5.2 legacy)
```

/usr/bin/ddsetup —> /opt/*ibmdir*/ldap/V6.1/sbin/ddsetup
/usr/bin/idsxcfg —> /opt/*ibmdir*/ldap/V6.1/sbin/idsxcfg
/usr/bin/idsxinst —> /opt/*ibmdir*/ldap/V6.1/sbin/idsxinst
/usr/bin/idsilist —> /opt/*ibmdir*/ldap/V6.1/sbin/idsilist
/usr/bin/idsicrt —> /opt/*ibmdir*/ldap/V6.1/sbin/idsicrt
/usr/bin/idsidrop —> /opt/*ibmdir*/ldap/V6.1/sbin/idsidrop
/usr/bin/idsdnpw —> /opt/*ibmdir*/ldap/V6.1/sbin/idsdnpw
/usr/bin/idssetport —> /opt/*ibmdir*/ldap/V6.1/sbin/idssetport
/usr/bin/idssethost —> /opt/*ibmdir*/ldap/V6.1/sbin/idssethost
/usr/bin/idsimigr —> /opt/*ibmdir*/ldap/V6.1/sbin/idsimigr
/usr/bin/idscfgsch —> /opt/*ibmdir*/ldap/V6.1/sbin/idscfgsch
/usr/bin/idsucfgsch —> /opt/*ibmdir*/ldap/V6.1/sbin/idsucfgsch
/usr/bin/idslogmgmt —> /opt/*ibmdir*/ldap/V6.1/sbin/idslogmgmt
/usr/bin/idsgendirksf —> /opt/*ibmdir*/ldap/V6.1/sbin/idsgendirksf
/usr/bin/idssupport —> /opt/*ibmdir*/ldap/V6.1/sbin/idssupport

Links created when -s fullsrv option is specified: Set of links for server commands for full server package

/usr/bin/bulkload —> /opt/*ibmdir*/ldap/V6.1/sbin/bulkload (5.2 legacy)
/usr/bin/idsbulkload —> /opt/*ibmdir*/ldap/V6.1/sbin/idsbulkload
/usr/bin/ldif2db —> /opt/*ibmdir*/ldap/V6.1/sbin/ldif2db (5.2 legacy)
/usr/bin/idsldif2db —> /opt/*ibmdir*/ldap/V6.1/sbin/idsldif2db
/usr/bin/db2ldif —> /opt/*ibmdir*/ldap/V6.1/sbin/db2ldif (5.2 legacy)
/usr/bin/idsdb2ldif —> /opt/*ibmdir*/ldap/V6.1/sbin/idsdb2ldif
/usr/bin/dbback —> /opt/*ibmdir*/ldap/V6.1/sbin/dbback (5.2 legacy)
/usr/bin/idsdbback —> /opt/*ibmdir*/ldap/V6.1/sbin/idsdbback
/usr/bin/dbrestore —> /opt/*ibmdir*/ldap/V6.1/sbin/dbrestore (5.2 legacy)
/usr/bin/idsdbrestore —> /opt/*ibmdir*/ldap/V6.1/sbin/idsdbrestore
/usr/bin/runstats —> /opt/*ibmdir*/ldap/V6.1/sbin/runstats (5.2 legacy)
/usr/bin/idsrunstats —> /opt/*ibmdir*/ldap/V6.1/sbin/idsrunstats (5.2 legacy)
/usr/bin/idscfgdb —> /opt/*ibmdir*/ldap/V6.1/sbin/idscfgdb
/usr/bin/idsucfgdb —> /opt/*ibmdir*/ldap/V6.1/sbin/idsucfgdb
/usr/bin/idscfgchglg —> /opt/*ibmdir*/ldap/V6.1/sbin/idscfgchglg
/usr/bin/idsucfgchglg —> /opt/*ibmdir*/ldap/V6.1/sbin/idsucfgchglg
/usr/bin/idscfgsuf —> /opt/*ibmdir*/ldap/V6.1/sbin/idscfgsuf
/usr/bin/idsucfgsuf —> /opt/*ibmdir*/ldap/V6.1/sbin/idsucfgsuf

idsrmlink

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

Note: **idsrmlink** does not restore any links previously backed up when **idslink** was run with the force option.

The syntax for the **idsrmlink** command is as follows (*installpath* is the path where IBM Tivoli Directory Server is installed):

Synopsis

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

where

-h Displays usage help for the command.

-v Displays version information about the command.

-n Pretend option. Displays the links that will be removed if you run the command with the options you specify.

-i Removes links only for client command utilities that begin with 'ids'.

-g Removes links only for client command utilities that do not begin with 'ids'.

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

-s *mode*
Removes links for server command-line utilities only. *mode* can be **proxy** if the directory server instance is a proxy server or **fullsrv** if the directory server instance is a full server.

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

idsversion

The Tivoli Directory Server version reporting tool.

Synopsis

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

Description

This utility provides the versions of all Tivoli Directory Server components installed in a machine like base client, servers, proxy servers, webadmin, and language packages.

Options

-? Displays the syntax format.

-b <*outputfile*>
Specifies the absolute path of a file for output redirection.

-t <*tmpOutDir*>
Specifies a directory for storing intermediate data during the processing of the tool.

-d Turns on debugging.

-r Lists the full information about each Tivoli Directory Server component. This is the same as the default option, but the information is printed in a raw format.

Examples

Issue the following command to list the full information about all installed components in a raw format:

```
idsversion –r
```

This command will list the version information for all Tivoli Directory Server installed components in the following format:

```
TDS_CLTJAVA#6.1.0.0
TDS_SRVPROXY#6.1.0.0
TDS_WEBADMIN#6.1.0.0
TDS_CLTBASE#6.1.0.0
TDS_SERVER32#6.1.0.0
TDS_LANGUAGE_EN#6.1.0.0
TDS_CLIENT32#6.1.0.0
```

The above information is generated for Tivoli Directory Server 6.1 installed components. To redirect the version information to another file, issue the following command:

```
idsversion –b <filename>
```

This command will redirect the version information for Tivoli Directory Server installed components to the file specified in the command. For Tivoli Directory Server 6.1 installed components the following output is redirected:

```
TDS java client version:6.1.0.0
32-bit TDS proxy server version:6.1.0.0
TDS Web-admin  server version:6.1.0.0
TDS base client version:6.1.0.0
32-bit TDS server version:6.1.0.0
TDS language(en) package version:6.1.0.0
32-bit TDS client version:6.1.0.0
```

tbindmsg

This utility is used by the server and client script utilities. It is not to be run by an end user.

Synopsis

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

Description

This command line tool is used for fetching a message from a local message catalog and for binding in arguments from the command line. All arguments must be strings.

Options

catalog_name

set_num

msg_num

def_mft

arg

SSL, TLS notes

To use the SSL or TLS -related functions associated with this utility, the SSL or TLS libraries and tools must be installed. The SSL or TLS libraries and tools are provided with IBM's Global Security Kit (GSKit), which includes security software developed by RSA Security Inc.

Note: For information regarding the use of 128-bit and triple DES encryption by LDAP applications, including the LDAP sample programs, see the information about LDAP_SSL in the *IBM Tivoli Directory Server Version 6.1*

Programming Reference. This section describes the steps required to build the sample programs and your applications so they can use SSL with the strongest encryption algorithms available.

See the makefile associated with the sample programs for more information on linking an LDAP application so that it has access to 128-bit and triple-DES encryption algorithms.

The content of a client's key database file is managed with the gsk7ikm utility. The gsk7ikm utility is used to define the set of trusted certification 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 issued by one of the trusted CAs. The gsk7ikm utility can also be used to obtain a client certificate, so that client and server authentication can be performed.

If the LDAP servers accessed by the client use server authentication only, 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 has been issued a certificate by one of the trusted CAs. In addition, 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. For example, if the LDAP server is using a high-assurance VeriSign certificate, you should obtain a CA certificate from VeriSign, import it 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 LDAP server can supply you with a copy of the server's certificate request file. Import the certificate request file into your key database file and mark it as trusted.

If the LDAP servers accessed by the client use client and server authentication, it is necessary to:

- Define one or more trusted root certificates in the key database file. This allows the client to be assured that the target LDAP server has been issued a certificate by one of the trusted CAs. In addition, 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 using gsk7ikm and request a client certificate from a CA. After receiving the signed certificate from the CA, store the certificate in the client key database file.

Chapter 3. Server utilities

This sections describes the server utilities.

Note: The **-I** option for server utilities (except idsicrt and idsidrop) that supports multiple directory instances on a local machine is optional, if you have the IDS_LDAP_INSTANCE environment variable set or if there is only one instance on the machine. If you have more than one instance created on your local machine, you must specify the **-I** option.

Attention: When you create a new directory server instance, be aware of the information that follows. If you want to use replication, use a distributed directory, or import and export LDIF data between server instances, you must cryptographically synchronize the server instances to obtain the best performance.

If you are creating a directory server instance that must be cryptographically synchronized with an existing directory server instance, you must synchronize the server instances ***before*** you do any of the following:

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

See Appendix A, "Synchronizing two-way cryptography between server instances," on page 129 for information about synchronizing directory server instances.

ddsetup

The **ddsetup** command is used to split an LDIF file for loading into a distributed directory. The **ddsetup** tool uses the proxy server's ibmslapd.conf file to partition entries. The data is split using the partition algorithm specified in ibm-slapdDNPartitionPlugin attribute of the configuration file.

Synopsis

```
 ddsetup [[-I Proxy Instance Name] [-B Base DN] [-i Input File]]
| [-f config_file] [-d debug level] [-l output_location]
[-s] [-v] -?
```

Options

-B <*base DN*>
Specifies the base DN or Split DN that should be used by the ddsetup tool

-d <*debuglevel*>
Specifies the debug level that should be used by the ddsetup tool

-f <*configfile*>
Specifies the configuration file that should be used by the ddsetup tool

-I <*instance name*>
Specifies the name of the proxy server instance that should be used by the ddsetup tool

-i <*input file*>
Specifies the input file that should be used by the ddsetup tool

-l <*output_location*>
Specifies the base directory that should be used by the ddsetup tool

-s Specifies that statistics mode should be enabled for the ddsetup tool

-v <*version*>
Specifies version information for the ddsetup tool

-? Displays help

Note: Composite Dn's are not supported by the ddsetup tool.

Examples

In this example, you have an existing database with 5 million entries for the subtree o=ibm,c=us. You want to distribute this data over 5 back-end servers. For this, you export the entries to an LDIF file so that the entries can be distributed among the back-end servers. See "idsdb2ldif, db2ldif" on page 89 for information on how to do this.

Note: In this example it is essential to note that the backends must be cryptographically synchronized. This means that the encryption seed values for the backends must be identical.

1. To create the LDIF file, issue the command:

   ```
   idsdb2ldif  -o mydata.ldif -s o=sample -I <instance_name>
   ```

2. Issue the command:

   ```
   ddsetup –I proxy -B "o=ibm,c=us" -i mydata.ldif
   ```

   where

   proxy: Is the proxy server instance

   The ddsetup command divides the mydata.ldif file into multiple LDIF output files on the basis of the number of partitions defined in the configuration file of the proxy server instance. The first output file corresponds to the partition index 1, the second output file corresponds to the partition index 2, the third output file corresponds to the partition index 3, and so forth.

3. Use idsldif2db or idsbulkload to load the data to the appropriate backend server.
   - 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

   Note: The correct LDIF output must be loaded on to the server with the correct corresponding partition index value, otherwise the proxy server is not able to retrieve the entries.

Similarly you can also split among multiple subtrees. In this example the parent DN (o=ibm,c=us) is split among three subtrees (ou=austin,o=ibm,c=us), (ou=raleigh,o=ibm,c=us), and (ou=poughkeepsie,o=ibm,c=us). The data for each of these subtrees is in turn subdivided:

- ou=austin,o=ibm,c=us - 5 back-end servers
- ou=raleigh,o=ibm,c=us - 3 back-end servers
- ou=poughkeepsie,o=ibm,c=us - 4 back-end servers

1. To create the LDIF file for the existing database, issue the command:

```
idsdb2ldif  -o mydata.ldif -s o=ibm,c=us -I <instance_name>
```

2. Issue the command:

   ```
   ddsetup –I proxy -B "o=ibm,c=us" -i mydata.ldif
   ```

   where

   proxy: Is the 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 of that subtree, the second output file corresponds to the partition index 2, the third output file corresponds to the partition index 3, and so forth. Remember that the partition index number starts at 1 for each subtree that is being distributed.
3. Use **idsldif2db** or **idsbulkload** to load the data to the appropriate backend server.
   - 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

   Note: The correct LDIF output must be loaded on to the server with the correct corresponding partition index value, otherwise the proxy server is not able to retrieve the entries.

The following example describes how to use the ddsetup tool to split the ddsample.ldif file:

1. Create a proxy server instance. Issue the command:

   ```
   idsadduser -u proxy -w proxyPW
   idsicrt -I proxy -x -l <instance location>

   where

   proxy: Is the proxy server instance and also the name of the proxy instance owner
   proxyPW: Is the password of the proxy instance owner (Username is 'proxy' in this case)
   idsadduser: Is used to create the proxy instance owner
   idsicrt: Is used to create the proxy instance
   ```

2. Configure o=sample as a partition base with the proxy server. Issue the command:

   ```
   idscfgsuf -I proxy -s o=sample

   where

   proxy: Is the proxy server instance name
   o=sample: Partition base configured with the proxy server
   ```

3. Set the admin DN and password for the proxy server. Issue the command:

```
idsdnpw -I proxy -u cn=root -p rootpw

where

proxy: Is the proxy server instance name
cn=root: Is the admin DN
rootpw: Is the admin password
```

4. Start the proxy server in configuration-only mode. Issue the command:

   ```
   ibmslapd -I proxy -a

   where

   proxy: Is the proxy server instance name
   ```

5. Add the configuration for splitting o=sample into 3 partitions. Issue the command:

   ```
   ldapadd -D cn=root -w rootpw -p port -f ddibmslapd.conf

   where

   cn=root: Is the admin DN
   rootpw: Is the admin password
   port: Port number on which the proxy is running
   ddibmslapd.conf: Sample configuration file
   ```

6. Run ddsetup with the sample data:

   ```
   ddsetup -I proxy -B o=sample -i ddsample.ldif

   where

   proxy: Is the proxy server instance
   o=sample: partition base
   ddsample.ldif: Sample ldif file
   ```

   Note: Both ddsample.ldif and ddibmslapd.conf are available as part of 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 of that subtree, the second output file corresponds to the partition index 2, and the third output file corresponds to the partition index 3. It is essential to note that the partition index number starts at 1 for each subtree that is being distributed. The following files are generated as a result of the above ddsetup command:

   - sample_1.ldif
   - sample_2.ldif
   - sample_3.ldif
   - default.ldif

   The default.ldif file will contain all the entries that couldn't conform to partitioning rules configured for the proxy server.

7. Use idsldif2db, idsbulkload, or ldapadd to load the data to the appropriate backend server.

   - Server1 (partition index 1) - sample_1.ldif
   - Server2 (partition index 2) - sample_2.ldif
   - Server3 (partition index 3) - sample_3.ldif

   Note: The correct LDIF output must be loaded on to the server with the correct corresponding partition index value, otherwise the proxy server is not able to retrieve the entries.

idsadduser

The **idsadduser** command line utility is used to create the DB2 and directory server instance owner and groups. The idsadduser utility can only be run by root on UNIX or a member of the Administrators group on Windows.

Note:

- If a directory server instance already exists and you attempt to create a user with the same name as the existing directory server instance using the idsadduser command, then an appropriate error message is displayed and the user will not be created.
- If a system user already exists and you attempt to create a user with the same name as the existing system user using the idsadduser command, then a message is displayed indicating that the user already exists. You may then choose to recreate the existing user with modified properties or exit without making any changes.

Synopsis

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

Options

-b <*outputfile*>
Specifies the full path of a file to redirect output. Only errors will be sent to the file if used in conjunction with the -q option. If debugging is turned on, debug output will be sent to this file also.

-d <*debuglevel*>
Sets the debug level. Use in conjunction with the ldtrc command.

-g <*groupname*>
Specifies the user's primary group. This option is valid only on AIX, Linux, Solaris, and HP-UX systems.

-l <*homedir*>
Specify the user's home directory. The default value for a user's home directory on AIX, Linux, and HP-UX is /home/username or /export/home/username on Solaris. This option is valid only on AIX,Linux, Solaris, and HP-UX.

-n Run in no-prompt mode. All output is generated except for messages requiring user interaction. The -w option must be used with this option.

-q Run in quiet mode. All output except errors is suppressed. If the -d option is also specified, trace output is not suppressed.

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

-v Prints the version information for this command.

-? Displays help

Examples

Given below are some examples that explain the usage of the idsadduser command:

1. The following command creates a new user on a UNIX platform with a user name as james, a primary group as staff, home directory at /home/james, and password as james1.

   ```
   idsadduser –u james –g staff –l /home/james –w james1
   ```
2. The following command enables a user to omit the password option so that the password is not visible as clear text on the command line.

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

   On issuing this command, the user is prompted to enter the password and the password is not displayed on the command line when it is entered.
3. To create a new user on Windows, issue the following command:

   ```
   idsadduser –u james –w james1
   ```

idsadscfg

The **idsadscfg** command line utility is used to configure the directory endpoint properties for an associated Tivoli Directory Server instance's AssemblyLines and EventHandlers. The adsync.xml and adsync_cfg.xml files are created at the following location during the time of Tivoli Directory Server installation: opt\IBM\ldap\V6.1\idstools\adsynch.

However, when using the **idsadscfg** command, these files are copied to the instance directory. This means, the files are copied to the following location: <inst_home_dir>\tdisoldir\config. The files at these two locations are independent of each other. Instance specific execution uses files from the instance directory.

Note: Error handling of Active Directory related arguments is done at run-time during idsadsrun, and not during configuration time (idsadscfg). If there are any errors reported during run time then the solution needs to be reconfigured with the correct arguments using idsadscfg.

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

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

-adb AD_Search_Base_DN
Specifies the subtree in Active Directory from which the AD Sync Solution is to propagate changes.

-adD AD_Login_DN
Specifies the Active Directory Login Name.

-adg AD_Group_Container_DN
Specifies the list of Active Directory LDAP subtrees from which groups in Active Directory will be synchronized to Tivoli Directory Server.

-adH AD_LDAP_URL
Specifies the LDAP URL and port for the Active Directory Domain Controller.

-adu AD_User_Container_DN
Specifies the DN of the container in Active Directory that contains the user entries to be synchronized with Tivoli Directory Server.

-adw AD_Login_Password
Specifies the password for the Active Directory Login Name.

-b output_file
Specifies the full path of a file in which to redirect output. If this option is used in conjunction with the -q option, only errors will be sent to the file.

-d debug_level
Sets the debug level. Use in conjunction with the ldtrc command.

-idsg TDS_Group_Container_DN
Specifies the DN of the Tivoli Directory Server container into which groups from this Active Directory will be copied. This container must exist in Tivoli Directory Server.

-idss TDS_Suffix
Used internally.

-idsu TDS_User_Container_DN
Specifies the DN of the container in Tivoli Directory Server into which users will be copied from Active Directory. This container must exist.

-isCfg
Returns a message about the AD Sync solution configuration status for this instance.

-n Run in no-prompt mode. All output is generated except for messages requiring user interaction.

-q Run in quiet mode. All output except errors is suppressed.

-v Prints the version information for this command.

-Z Use an SSL connection to connect to Active Directory.

-? Displays the usage.

idsadsrun

The **idsadsrun** command line utility is used to start execution of a specified instance’s EventHandler.

Synopsis

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

Options

-b *<outputfile>*
Specifies the full path of a file in which to redirect output. If this option is used in conjunction with the -q option, only errors will be sent to the file. If debugging is turned on, debug output will be sent to this file also.

-k Stops the ADSync solution associated with the instance.

-d <*debuglevel*>
Sets the debug level in the LDAP library. Use in conjunction with the ldtrc command.

-I <*instancename*>
Specifies the name of the directory server instance to update.

-n Run in no-prompt mode. All output is generated except for messages requiring user interaction. The -q option must be used with this option.

-q Run in quiet mode. All output except errors are suppressed. If the -d option is also specified, trace output is not suppressed.

-v Prints the version information about the command.

-? Displays the help screen.

idsbulkload, bulkload

The **idsbulkload** utility is used to load the directory data from an LDIF file. It is a faster alternative to **idsldif2db** and is available for bulk-loading large amounts of data in LDIF format.

Attention: If you want to import LDIF data from another server instance, you must cryptographically synchronize the LDIF import file with the server instance that is importing the LDIF file; otherwise any AES-encrypted values in the LDIF file will not be imported. See Appendix A, "Synchronizing two-way cryptography between server instances," on page 129 for information about synchronizing directory server instances.

Notes:

1. The server must be stopped before using the server import utilities.
2. Ensure that no applications are attached to the directory database. If there are applications attached, none of the server utilities will run.
3. All idsbulkload environment variables are no longer supported in IBM Tivoli Directory Server 6.0 and later versions. The ACLCHECK, ACTION, LDAPIMPORT, SCHEMACHECK, and STRING_DELIMITER environment variables are replaced with the command line options -A, -a, -L, -S, -s respectively. The command line switches are now **case sensitive**.

   Note: Because of the **idsbulkload** ACL processing enhancements in the IBM Tivoli Directory Server version 6.0 release, the -A option, while still supported, is deprecated. The following options are also deprecated:
   - -c
   - -C
   - -e
4. To run the **idsbulkload** utility you must have dbadm or sysadm privilege. If you use a Windows system, you must also run the idsbulkload utility within the DB2 command line interpreter (CLI). To start the DB2 CLI, click **Start->Run**, type `db2cmd` and click **OK**.
5. If archival logging is enabled in DB2, the **idsbulkload** utility will fail. Make sure archival logging is disabled before using the **idsbulkload** utility.

   ```
   update database configuration for ldapdb2 using LOGRETAIN OFF USEREXIT OFF
   ```
6. If loading data containing unique attributes, the DB2 unique constraints for the attributes that are going to be modified are dropped. After the data is loaded

the DB2 unique constraints are established for the attributes whose unique constraints were dropped and for any new unique attributes listed in the unique attribute entry in the input file.

Note: If duplicate values are loaded for attributes that are specified as unique attributes, the DB2 unique constraint is not created for that attribute. This information is recorded in the idsbulkload.log file.

7. If loading additional data to a directory already containing data, make sure you have a directory backup before using **idsbulkload** to add entries.

Synopsis

```
idsbulkload | bulkload -i <ldiffile> [-I <instancename>
[-a <parse_and_load|parseonly|loadonly>] [-A <yes|no>]
[-b] [-c | -C <yes|no>] [-d <number>] [-e drop_index]
[-E <number>] [-f configfile] [-g] [-G] [-k <number>]
[-L <path>] [-n | -N] [-o <filename>]
[-p | -P <yes|no>] [-s <character>] [-R <yes|no>]
[-S <yes|no|only>] [-t <filename>] [-v]
[-W outputfile] [-x|-X <yes|no>]] | [-?]
```

Options

-a <parse_and_load | parseonly | loadonly>
Specifies the load action mode.

-A <yes | no>
Specifies whether to process the ACL information contained in the LDIF file. The default is **yes**. The **no** parameter loads the default acls.

Note: This option is deprecated.

-b Specifies to suppress the progress indicator.

-c | -C <yes | no>
Allows you to skip index recreation. For example, if you are running successive idsbulkloads and you want to skip recreation between loads, you can postpone index creation until the last **idsbulkload**. Issue the final **idsbulkload** with **-c yes**.

-d <*number***>**
Use the **-d** to set the level of the debug mask and to turn debug on. Use this option to find out the data records that might have a problem and cause parsing errors. See Chapter 4, “Debugging levels,” on page 127 for information about debug levels.

Note: Ensure that the **ldtrc** utility is on before using the **-d** option, otherwise no messages are displayed. Issue the command `ldtrc on`.

-e *drop_index*
Drop indexes before load (yes or no).

-E <*number***>**
Specifies the number limit for parsing errors reported. When the limit is reached the **idsbulkload** command exits. The default is infinity.

-f *configfile*
LDAP directory configuration file.

-g Specifies not to strip the trailing spaces on attribute values.

-G Specifies to add members into existing static groups. This option cannot be specified if the **-k** option has been selected.

-i <*ldiffile***>**
Specifies the name of the input file containing the LDIF data to be loaded into the directory. It might include a path. The file <*IDS_LDAP_HOME*>examples/sample.ldif contains some sample data in the correct format. IDS_LDAP_HOME is the path to the installed LDAP support. IDS_LDAP_HOME varies by operating system platform:

- AIX operating systems - /opt/IBM/ldap/V6.1
- HP-UX operating systems - /opt/IBM/ldap/V6.1
- Linux operating systems - /opt/ibm/ldap/V6.1
- Solaris operating systems - /opt/IBM/ldap/V6.1
- Windows operating systems - <*local_drive*>:\Program Files\IBM\LDAP\V6.1 (This is the default install location. The actual IDS_LDAP_HOME is determined during 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** option must be set to **parse_and_load**. This option cannot be specified if the **-G** option has been selected.

-L <*path***>**
Specifies the directory used for storing temporary data. The default path for this temporary storage is:

- AIX,Linux, Solaris, and HP-UX operating systems in <*directory server instance owner home*>/idsslapd-<*directory server instance name*>/tmp/ldapimport

  Note: If you are logged in as root, the **idsbulkload** command fails when you specify the location of the temporary directory using the **–L** option. Hence, you must login as an instance owner to create a temporary directory, and then run the **idsbulkload** command as root. To login as an instance owner, issue the following command:

  ```
  su <instance name>
  ```
- Windows operating systems in <*TDS home directory*>\idsslapd-<*directory server instance name*>\tmp\ldapimport

-n | -N
Specifies that the load is nonrecoverable. With this option, idsbulkload uses less diskspace and runs faster, but if loading of data fails for any reason, all data in the database is lost.

-o <*filename***>**
Specifies to generate an output file to preserve the IBM-ENTRYUUID and the timestamp values created during the parsing phase of **idsbulkload**.

-p | -P <yes|no>
Specifies whether to generate password policy attributes for entries containing the attribute userpassword.

-R <yes|no>
Specifies whether to remove the directory which was used for temporary data. This directory is the default directory or the one specified by the **-L** parameter. Default is **yes**.

Note: Although the default is **yes** , there are two exceptions. If **idsbulkload** ends in a bad state (error condition), the temp files are not deleted on error, because they are needed for recovery, or if the user chooses the **-a parseonly** option the temp files are not deleted because the files are needed for the load phase.

-s <*character*>
Specifies the string delimiting character used for importing

Note: **idsbulkload** might fail to load LDIF files that contain certain UTF-8 characters. This is because of a problem with the DB2 LOAD tool when parsing the default **idsbulkload** string delimiter, vertical bar (|) in multi-byte character sets. In this case, reassign the string delimiter to $.

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

-S <yes|no|only>
Verifies that individual directory entries are valid based on the object class definitions and attribute type definitions found in the configuration files.

Schema checking verifies that all object classes and attributes have been defined, that all attributes specified for each entry comply with the list of ″required″ and ″allowed″ attributes in the object class definition, and that binary attribute values are in the correct 64-bit encoded form.

yes Schema checking is done on the data, before adding it to the directory.

no No schema checking is done on the data before adding it to the directory. This provides much faster performance. This option assumes that the data in the input file is valid. This is the default option.

only Schema checking is done on the data, but it is not added to the directory. This option provides the most feedback and error reporting.

The recommended approach is to use the **-S only** option first to validate the data, and thereafter to use the default **-S no** whenever loading the data into the directory.

-t <*filename*>
Specifies to use the IBM-ENTRYUUID and the timestamp values from the specified input file instead of generating them during the parsing. If these values are also present in the input LDIF file in the form of controls, the controls are ignored.

-v Specifies verbose mode. This option gives you the greatest amount of detail.

-W *outputfile*
Specifies the full path of a file in which to redirect output.

-x | -X <yes|no>
Specifies whether to translate entry data to database code page. Default is **no**.

Note: This parameter is only necessary when using a non-UTF-8 database.

-? Displays the syntax format.

Description

For better performance the **idsbulkload** tool assumes that the data in the input file is correct or that the data has been checked in an earlier loading. The **idsbulkload** tool can, however, perform some basic checks on the input data.

The **idsbulkload** utility cannot run while the directory server (**idsslapd**) is running.

In addition to the disk space required for data storage in the local database directory, the **idsbulkload** tool requires temporary storage for data manipulation before inserting the data into the database. The default path for this temporary storage is platform specific. See the **-L** option description for the path names. You can change the path using the **-L** option:

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

You must have write permission to this directory. You need temporary storage at least 2.5 times the size of the LDIF file that is available in the ldapimport directory. You still might need additional temporary storage depending on your data.

If you receive an error like the following:

```
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 environment variable DB2SORTTMP to a directory (or directories) in a file system with more space to be utilized during the idsbulkload. Multiple directories can be specified separated by a comma (,) as in:

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

The **-o** and **-t** options are useful when adding large amounts of new directory data into existing replication environments. If servers A and B are peer servers and you want to add a large number of new entries to the directory under the current replication context, you can:

1. Generate the LDIF file.
2. Run **idsbulkload** with the **-o** option on server A to load the data and to generate a new file that contains all operational attributes created during bulkload.
3. Copy the operational attributes output file to server B and run **idsbulkload** with the **-i** and **-t** option to import the LDIF file using the same operational attributes.

This ensures that the operational attribute values are preserved across the replicating servers under the same replication context.

The **-G** option is useful when expanding an already existing static group with a large number of new members. The existing entry must have an object class that accepts member or uniquemember as its attribute. For example, if you wanted to add five million new members from static group 1 on the source server of company1 to an existing group, static group A on the target server of companyA, you would:

1. Create the LDIF file from the source server. Use an editor to remove any attributes other than member or uniquemember from the file so that it has the form:

   ```
   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 entry (group) 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. Perform any necessary global changes to the file. In this case, the company name needs to be changed on 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 problems, divide the file into multiple files of a more manageable size. For this example, divide the source file into five files of one million attributes and 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
   ```

   and so forth.

5. Issue the **idsbulkload** command with the **-G** to load each of the files to the target server.

The **idsbulkload** utility verifies that the DN already exists and that its object class allows member or uniquemember as valid attributes before loading the input file.

Note: The **idsbulkload** utility does not check for duplicate attributes.

When running **idsbulkload**, inspect the output messages carefully. If errors occur during execution, the directory might be incompletely populated. You might need to drop all the LDAP tables, or drop the database (recreate an empty database), and start over. If this happens, no data was added to the directory, and the **idsbulkload** must be attempted again. In addition, you will lose any existing data when you drop all the LDAP tables.

The file <*IDS_LDAP_HOME*>/examples/sample.ldif includes sample data. You can use the data in the file to experiment with populating a directory using the **idsbulkload** tool, or you can use the **idsldif2db** command line utility. However, the **idsldif2db** utility might be considerably slower than the **idsbulkload** utility for large amounts of data.

For performance reasons, the **idsbulkload** tool does not check for duplicate entries. Ensure that your input 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 for the reasons. This log file is located for:

- Windows operating systems in <*TDS home directory*>\idsslapd-<*directory server instance name*>\tmp\ldapimport
- AIX, Linux , Solaris, and HP-UX operating systems in <*TDS home directory*>/idsslapd-<*directory server instance name*>/tmp/ldapimport

  Note: The default path on Windows can be changed by the user.

If the **-L** option was specified, find the file in the directory defined by the **-L** option. Correct the problem and rerun **idsbulkload**. **idsbulkload** reloads the files from the last successful load consistency point.

When **idsbulkload** fails, the recovery information is stored in

- Windows operating systems in <*top level drive*>\idsslapd-<*directory server instance name*>\logs\bulkload_status
- AIX, Linux , Solaris, and HP-UX operating systems in <*directory server instance owner home*>/idsslapd-<*directory server instance name*>/logs/bulkload_status

This file is not removed until all of the data is loaded successfully. This insures the data integrity of the directory. If you decide to reconfigure the database and start over, the idsbulkload_status file needs to be removed manually or **idsbulkload** still tries to recover from the last successful load point.

idscfgchglg

Command to configure a change log for a directory server instance.

Synopsis

```
idscfgchglg [-I instancename [-m maxentries] [-y maxdays] [-h maxhours]
           [-f configfile] [-d debuglevel] [-b outputfile] [-q] [-n]] |
            -v | -?
```

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 normal database. The change log information is added to the directory server instance's ibmslapd.conf file. A change log requires only the directory server instance name for which it is being configured. A change log automatically picks up the database instance name that is associated with the directory instance and creates a new database in the same database instance. It is essential that a database instance with the same name as the directory server instance must already exist. Also, a database for the directory server instance must already be created and for UNIX and Linux platforms the local loopback service must be registered in the /etc/services file.

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

You can optionally specify the maximum number of entries to keep in the change log and the maximum age each entry in the change log is kept until it is automatically destroyed. If you do not specify any options, the entries in the change log never expire and the size of the change log is a maximum of 1,000,000 entries.

Options

-b <*outputfile*>
: Specifies the full path of a file to redirect console output into. Only errors are sent to the file if used in conjunction with the **-q** option. If debugging is turned on, that output is sent to this file also.

-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. This parameter is for use by IBM service personnel. See Chapter 4, "Debugging levels," on page 127 for additional information on debug levels.

-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 in hours the maximum amount of time to keep entries in the change log. This option can be used with the **-y** <*maxdays*> to specify the maximum age of a change log entry.

-I <*instancename*>
: Specifies the instance name for the directory server instance that is to be updated.

-n
: Specifies to run no prompt mode. All output is generated, except for messages that require user interaction. This option requires the **-w** option.

-m <*maxentries*>
: Specify the maximum number of entries to keep in the change log. A value of 0 means there is no limit on the number of entries.

-q
: Specifies to run in quiet mode. All output except errors messages are suppressed. If the **-d** option is also specified, trace output is not suppressed.

-y <*maxdays*>
: Specifies in days the maximum amount of time to keep entries in the change log. A value of 0 means that there is no age limit on entries in the change log. This option can be used with the **-h** <*maxhours*> to specify the maximum age of a change log entry.

-v
: Specifies to display version information about the command.

-? Displays the syntax format.

Examples

To configure a change log with no age limit or size limit, issue the command:

```
idscfgchglg –m 0
```

To configure a default change log with a size limit of 1,000,000 and an entry age of 25 hours, issue the command:

```
idscfgchglg –y 1 –h -1
```

Note: After the change log is configured, the **-y**, **-h**, and **-m** options can be used to update the maximum age and maximum size of the entries in the change log.

idscfgdb

Command to configure a database for a directory server instance.

Synopsis

```
idscfgdb [-I instancename [[-w dbadminpw] [-a dbadminid -t dbname -l dblocation
[-x]]] [-f configfile] [-d debuglevel] [-b outputfile] [-q] [-n]] | -v | -?
```

Description

The **idscfgdb** command configures the database for a directory server instance. The **idsicrt** command must have already run successfully to create the database instance. In addition, the database instance owner must be set up correctly. Otherwise, the command fails. (See the *IBM Tivoli Directory Server version 6.1 Installation and Configuration Guide* for information about setting up required users and groups.)

The directory server instance owner specifies a database administrator user ID, a database administrator password, the location to store the database, and the name of the database. The database administrator ID specified must already exist on the system.

After successfully creating the database, the information is added to the ibmslapd.conf file of the directory server instance. The database and local loopback setting are created, if they do not exist. You can specify whether to create the database as a local codepage database or as a UTF-8 database, which is the default.

Options

-a <*dbadminid*>
Specifies the DB2 administrator ID. The DB administrator must already exist on the system and have the proper authority.

-b <*outputfile*>
Specifies the full path of a file to redirect console output into. Only errors are sent to the file if used in conjunction with the **-q** option. If debugging is turned on, that output is sent to this file also.

-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. This

parameter is for use by IBM service personnel. See Chapter 4, “Debugging levels,” on page 127 for additional information on debug levels.

-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.

-I <*instancename*>
Specifies the instance name for the directory server instance that is to be updated.

-l <*dblocation*>
Specifies the DB2 database location. For AIX, Linux, Solaris, or HP-UX systems, this is a directory name (for example, /home/ldapdb2). For Windows systems, this must be a drive letter. The database requires at least 80 MB of free space. Additional disk space is needed to accommodate growth as directory entries are added.

-n Specifies to run no prompt mode. All output is generated, except for messages that require user interaction. This option requires the **-w** option.

-q Specifies to run in quiet mode. All output except errors messages are suppressed. If the **-d** option is also specified, trace output is not suppressed.

-t <*dbname*>
Specifies the DB2 database name.

-v Specifies to display version information about the command.

-w <*dbadminpw*>
Sets the password for the DB2 administrator in the configuration file for the directory server instance. Also sets the password for the change log database owner in the configuration file if the change log is enabled.

This option is required for the **-n** option.

-x <*instancename*>
Specifies to create the DB2 database in a local codepage.

-? Displays the syntax format.

Examples

To configure a database called ldapdb2 in the location /home/ldapdb2 and the DB2 database administrator ID is ldapdb2 with the password of secret, issue the command:

```
idscfgdb –a ldapdb2 –w secret –t ldapdb2 –l /home/ldapdb2
```

If the password is not specified, you are prompted for the password. Your password is not displayed on the command line when you enter it.

idscfgsch

Command to configure a schema file for a directory server instance.

Synopsis

```
idscfgsch [-I instancename -s schemafile [-f configfile] [-d debuglevel]
          [-b outputfile] [-q] [-n]] | -v | -?
```

Description

The **idscfgsch** 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 the file from directory server instance's ibmslapd.conf file.

Options

-b <*outputfile*>
Specifies the full path of a file to redirect console output into. Only errors are sent to the file if used in conjunction with the **-q** option. If debugging is turned on, that output is sent to this file also.

-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. This parameter is for use by IBM service personnel. See Chapter 4, "Debugging levels," on page 127 for additional information on debug levels.

-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.

-I <*instancename*>
Specifies the instance name for the directory server instance that is to be updated.

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

-q Specifies to run in quiet mode. All output except errors messages are suppressed. If the **-d** option is also specified, trace output is not suppressed.

-s <*schemafile*>
Specifies the schema file to add to the directory server instance.

-v Specifies to display version information about the command.

-? Displays the syntax format.

Examples

To configure the schema file /home/mydir/myschema.oc to the directory server instance's ibmslapd.conf file, issue the command:

```
idscfgsch –s /home/mydir/myschema.oc
```

idscfgsuf

Command to configure a suffix for a directory server instance.

Synopsis

```
idscfgsuf [-I instancename -s suffix [-f configfile] [-d debuglevel] [-b outputfile]
[-q] [-n]] | -v | -?
```

Description

The **idscfgsuf** configures a new suffix for a directory server instance. The suffix is added to directory server instance's ibmslapd.conf file. This command fails if the suffix already exists in the configuration file.

Options

-b <*outputfile*>
: Specifies the full path of a file to redirect console output into. Only errors are sent to the file if used in conjunction with the **-q** option. If debugging is turned on, that output is sent to this file also.

-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. This parameter is for use by IBM service personnel. See Chapter 4, "Debugging levels," on page 127 for additional information on debug levels.

-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.

-I <*instancename*>
: Specifies the name of the directory server instance. This option is required if there are additional directory server instances on the local machine.

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

-q Specifies to run in quiet mode. All output except errors messages are suppressed. If the **-d** option is also specified, trace output is not suppressed.

-s <*suffix*>
: Specifies to add a suffix to the directory server instance.

-v Specifies to display version information about the command.

-? Displays the syntax format.

Examples

To configure the suffix o=sample on a machine with a single directory server instance, issue the command:

```
idscfgsuf -s o=sample
```

To configure the suffix o=sample on a machine with a multiple directory server instances, issue the command:

```
idscfgsuf -I <instancename> -s o=sample
```

idsdbback, dbback

The **idsdbback** command is used to backup your directory when the server is offline. It also backs up the database, and configuration, schema and encryption key stash files. You must stop the server before using this command.

Notes:

1. Backing up to 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"."
   ```

   idsdbback or idsdbrestore must be done on a local drive or partition only.
2. The version of DB2 used to back up your database when the server is offline must be the same as the version of DB2 used to restore your database.
3. The **idsdbback** utility does not backup the change log.
4. To know more about online backups, see Appendix G in the *IBM Tivoli directory server 6.1 Administration guide*.

Synopsis

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

Options

-b <*outputfile*>
Specifies the full path of a file to redirect console output into. Only errors are sent to the file if used in conjunction with the **-q** option. If debugging is turned on, that output is sent to this file also.

-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. This parameter is for use by IBM service personnel. See Chapter 4, “Debugging levels,” on page 127 for additional information on debug levels.

-I <*instancename*>
Specifies the name of the directory server instance for which you want to backup the database.

-k <*backupdir*>
Specifies the directory to use to back up the database.

Note: When performing multiple backups, ensure that each backup is in a separate directory. If you have more than one version of the database backup file in the same directory, the idsdbrestore tool only restores the database with the most current timestamp.

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

-q Specifies to run in quiet mode. All output except errors messages are suppressed. If the -d option is also specified, trace output is not suppressed.

-v Specifies to display version information about the command.

-? Displays the syntax format.

Example

The following command can be used to back up a database:

```
idsdbback -I ldapdb2 -k /backupdir
```

idsdbmigr

The **idsdbmigr** tool is used to perform migration of the DB2 instance for an existing IBM Tivoli directory server instance. Using this tool, user data can be migrated from DB2 version 8 while successfully converting it to a fully functioning DB2 9.1 instance and DB2 9.1 database.

Note:

- The directory server instance must be a Tivoli Directory Server 6.1 instance.
- The directory server instance name must be specified using the –I option. This is a required argument.
- After performing migration of a DB2 server for an existing IBM Tivoli directory server instance, if the instance is dropped using the idsidrop command, then temporary files created by DB2 during migration are not deleted.

Synopsis

```
idsdbmigr -I <instance name> [-N <db2_9.1_location>] [-h | -?]
```

Options

-I <*instance name*>
Specifies the name of the directory instance.

-N <*db2_9.1_location*>
Specifies DB2 version 9.1 location.

-h | -? Displays the debug help on the screen.

Examples

Given below are some examples that explain the usage of the idsdbmigr tool. To perform only pre-migration tasks on DB2 version 8 database, issue the following command:

```
idsdbmigr -I <instance name>
```

To perform a complete migration from DB2 version 8 database to a DB2 version 9 database, issue the command given below:

Note: The -N option specifies the location where DB2 version 9 is installed.

```
For windows-based systems :

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

For UNIX systems:

idsdbmigr -I <instance name> -N /opt/IBM/db2/V9.1
```

idsdbrestore, dbrestore

The **idsdbrestore** command is used to restore your database and directory configuration when the server is offline. You must stop the server before using this command.

Notes:

1. Backing up 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"."
   ```

   idsdbback or idsdbrestore must be done on a local drive or partition only.
2. The version of DB2 used to restore your database when the server is offline must be the same as the version of DB2 used to back up your database.
3. You can also run a rollforward command when restoring from an online backup. Following the restore, before starting the server, do the following:

   ```
   db2 rollforward db <dbname> to end of logs and stop
   ```

   You must run this command if you get the following error:

   ```
   SQL1117N A connection to or activation of database <dbname> cannot be
    made because of ROLL-FORWARD PENDING.
   ```
4. The **idsdbrestore** command does not restore the change log.

Synopsis

```
idsdbrestore | dbrestore  -I instancename -k backupdir [-d debuglevel]
             [-b outputfile] [-r] [-q] [-n]] | -v | -?
```

Options

-b <*outputfile*>
: Specifies the full path of a file to redirect console output into. Only errors are sent to the file if used in conjunction with the **-q** option. If debugging is turned on, that output is sent to this file also.

-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. This parameter is for use by IBM service personnel. See Chapter 4, “Debugging levels,” on page 127 for additional information on debug levels.

-I <*instancename*>
: Specifies the name of the directory server instance for which you want to back up the database.

-k <*backupdir*>
: Specifies the directory used to back up the database. The **idsdbrestore** command only restores a database into a database and database instance with the same names and database location as were used for the database backup.

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

-q Specifies to run in quiet mode. All output except errors messages are suppressed. If the **-d** option is also specified, trace output is not suppressed.

-r Specifies not to restore the ibmslapd.conf file.

-v Specifies to display version information about the command.

-? Displays the syntax format.

Example

The following command can be used to restore a database:

```
idsdbrestore -I ldapdb2 -k /backupdir
```

idsdb2ldif, db2ldif

This program is used to dump entries from a directory into a text file in LDAP Directory Interchange Format (LDIF).

Note: This utility can be run at anytime, the server does not need to be stopped.

Attention: If you are exporting data that will be imported into an Advanced Encryption Standard (AES)-enabled server and *if the two servers are not cryptographically synchronized*, select the **Export data for AES-enabled destination server** check box. Then complete the **Encryption seed** and **Encryption salt** fields. See Appendix A, “Synchronizing two-way cryptography between server instances,” on page 129 for information about cryptographic synchronization of servers.

When the source server (the server you are exporting data from) and the destination server (the server into which you are importing the data) are using non-matching directory key stash files, and you specify the encryption seed and salt values of the destination server, any AES-encrypted data is decrypted using the source server’s AES keys, then re-encrypted using the destination server’s encryption seed and salt values. This encrypted data is stored in the LDIF file.

Note: The source server’s SHA-encoded directory encryption seed is written to the LDIF file for reference during import. For parsing purposes, this encryption seed reference is contained in a cn=crypto,cn=localhost pseudo entry that is informational only, and is not actually loaded as part of the import.

The encryption seed is used to generate a set of AES secret key values. These values are stored in a directory stash file and used to encrypt and decrypt directory stored password and secret key attributes. The encryption seed must contain only printable ISO-8859-1 ASCII characters with values in the range of 33 to 126, and must be a minimum of 12 and a maximum of 1016 characters in length. See Appendix C, “ASCII characters from 33 to 126,” on page 133 for information about these characters.

The encryption salt is a randomly generated value that is used to generate AES encryption keys. You can obtain the destination server’s salt value by searching (using the **idsldapsearch** utility) the destination server’s ″cn=crypto,cn=localhost″ entry. The attribute type is ibm-slapdCryptoSalt.

Synopsis

```
idsdb2ldif | db2ldif [-o outputfile -I instancename [-f configfile]
          [-n filterDN] [-c comments] [-k ?|keyseed -t keysalt] [-j]
          [-d debuglevel] [[-s subtreeDN [-x]] | [-p on|off] [-l]]
          [-W] [-x]] | -?
```

Options

All options are case sensitive.

-c <*comments*>
Specifies the comments to be added into output LDIF file.

-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. This parameter is for use by IBM service personnel. See Chapter 4, "Debugging levels," on page 127 for additional information on debug levels.

-f <*configfile*>
Specifies the full path to the configuration file to use. If not specified, the default configuration file for the directory server instance is used.

-I <*instancename*>
Specifies the name of the directory server instance.

-j Indicates that the four operational attributes, createTimestamp, creatorsName, modifiersName, and modifyTimestamp are not to be exported to the LDIF file.

-k ?|<*encryption seed*>
Specifies the encryption seed value used to create the directory key stash file of the destination server. The encryption seed must only contain printable ISO-8859–1 ASCII characters with values in the range of 33 to 126, and must be a minimum of 12 and a maximum of 1016 characters in length. See Appendix C, "ASCII characters from 33 to 126," on page 133. Use the **?** to generate a password prompt. Using this prompt prevents your encryption seed from being visible through the **ps** command. If you specify the **-k** parameter, you must also specify the **-t** parameter because both are necessary to generate encryption keys for re-encryption during export.

-l Exports all suffixes, except the cn=pwdpolicy suffix, in addition to the cn=localhost subtree. This option cannot be used with the **-s** option.

-n <*filterDN*>
Specifies the DN of filter entry used to filter entries before adding into output LDIF file. If this option is specified, all directory entries stored in the database are filtered and then the partial entry is written to the output LDIF file. The filtering is done as per filter specified in filter DN.

-o <*outputfile*>
Specifies the LDIF output file to contain the directory entries in LDIF. All entries from the specified subtree are written in LDIF to the output file. This option is required. If the file is not in the current directory, a full path and file name must be specified.

-p on|off
Exports all suffixes, except cn=localhost subtree, in addition to the cn=pwdpolicy suffix . The default setting is **off**. This option cannot be used with the **-s** option.

-s <*subtree DN*> **[-x]**
The subtree DN identifies the top entry of the subtree that is to be dumped to the LDIF output file. This entry, plus all below it in the directory hierarchy, are written out. If this option is not specified, all directory entries stored in the database are written to the output file based on the suffixes specified in the configuration file except for the suffix cn=localhost. When the **-x** option is specified it means to exclude the subtree specified by the **-s** option. The **-x** option cannot be used with the **-l** or **-p** options.

-t <*encryption salt*>
Specifies the encryption salt value used to create the directory key stash file of the destination server. The salt value can be obtained by searching the destination server's ″cn=crypto,cn=localhost″ entry. The attribute name is ibm-slapdCryptoSalt. The encryption seed must only contain printable ISO-8859–1 ASCII characters with values in the range of 33 to 126, and must be 12 characters in length. See Appendix C, "ASCII characters from 33 to 126," on page 133. If you specify the **-t** parameter, you must also specify the **-k** parameter because both are necessary to generate encryption keys for re-encryption during export.

-W <*outputfile*>
Specifies the full path of a file in which to redirect output.

-x Use FIPS mode processing (SSL/TLS only).

-? Displays the syntax format.

All other command line inputs result in a syntax error message, after which the proper syntax is displayed.

idsdiradm, ibmdiradm

Command to start or stop the administration daemon.

Synopsis

```
idsdiradm | ibmdiradm [-I instancename [-f configfile] [-h debuglevel] [-t]
          [[ [-p port] [-s secureport] [-c]] | -k | -i | -u] ] | -v | -?
          | -h ?
```

Description

The **idsdiradm** command starts or stops the administration daemon.

Options

-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<*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. This parameter is for use by IBM service personnel. See Chapter 4, "Debugging levels," on page 127 for additional information on debug levels.

-h ? Displays the debug help screen.

-I <*instancename*>
Specifies the name of the admin daemon instance to start or stop.

-k Specifies to stop the administration daemon.

-p *port* Specifies the non-SSL port.

-s *secureport*
Specifies the SSL port.

-v Specifies to print the version information.

-? Displays the syntax format.

The following parameters are for Windows systems only:

-i Specifies to install the admin daemon instance as a service.

-u Specifies to remove the admin daemon instance as a service.

The following parameter is for AIX, Linux, Solaris, and HP-UX systems only:

-c Specifies to run the server in console mode.

-t Specifies to tail the server log until final start-up messages are displayed on the console.

Examples

To start the administration daemon, issue the command:

```
idsdiradm -I <instancename>
```

For Windows systems, you can also:

1. Through the Control Panel, open the Services window.
2. Select and right click **IBM Tivoli Directory Admin Daemon V6.1 -** <*instancename*>
3. Click **Start**.

To stop the administration daemon:

- Issue the command (remotely or locally):

  ```
  ibmdirctl -D <AdminDN> -w <Adminpw> -h <hostname> -p <port> admstop
  ```

  or (locally)

  ```
  idsdiradm -k -I <instancename>
  ```
- For Windows systems, you can also:
  1. Through the Control Panel, open the Services window.
  2. Select and right click **IBM Tivoli Directory Admin Daemon V6.1 -** <*instancename*>
  3. Click **Stop**.

idsdnpw

The administration DN and password utility.

Synopsis

```
idsdnpw [-I instancename [[-u userDN] -p password] [-f configfile] [-d debuglevel]
        [-b outputfile] [-q] [-n]] | -v | -?
```

Description

The **idsdnpw** command provides a way to set or change the administrator DN and password for a directory server instance. The command can only be run when the directory server instance is not running. The administrator specifies an administrator password and optionally specifies an administrator DN which the utility writes to the ibmslapd.conf file. The administrator DN is set to cn=root by default.

Options

-b <*outputfile*>
Specifies the full path of a file to redirect console output into. Only errors are sent to the file if used in conjunction with the **-q** option. If debugging is turned on, that output is sent to this file also.

-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. This parameter is for use by IBM service personnel. See Chapter 4, "Debugging levels," on page 127 for additional information on debug levels.

-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.

-I <*instancename*>
Specifies the name of the directory server instance. This option is required if there are additional directory server instances on the local machine.

-n Specifies to run no prompt mode. All output is generated, except for messages that require user interaction. This option requires the **-p** option.

-p <*password*>
Specifies to change the directory administrator password. If an administration DN value is not specified (the **-u** option), the current value of the administrator DN is used. If the administrator DN is not defined, then the the default value cn=root is used. This option is required if the -n option is specified.

-q Specifies to run in quiet mode. All output except errors messages are suppressed. If the **-d** option is also specified, trace output is not suppressed.

-u <*AdminDN*>
Specifies to create or change the directory administrator distinguished name (DN).

-v Specifies to display version information about the command.

-? Displays the syntax format.

Examples

To set the administrator DN to cn=myname and the password to secret, issue the command:

```
idsdnpw –u cn=myname –p secret
```

If the password is not specified, you are prompted for the password. Your password is not displayed on the command line when you enter it.

Note: The administrator's password must conform to the administration password policy requirements, if the administration password policy has been enabled.

idsgendirksf

Command to regenerate a directory key stash file for a directory server instance.

Synopsis

```
idsgendirksf [-s salt [-e encryptseed] -l location
            [-d debuglevel] [-b outputfile] [-q] [-n]] | -v | -?
```

Description

The **idsgendirksf** command uses the encryption seed and salt values that were used when creating the instance to regenerate the instance's directory key stash file. The original encryption seed value is the one that you supplied when you created the instance. The original salt value can be obtained by searching the server instance's ″cn=crypto,cn=localhost″ entry. The attribute value is ibm-slapdCryptoSalt. These two values regenerate the instance's ibmslapddir.ksf file.

When using the **idsgendirksf** utility, if you use a character in the salt value or the encryption seed value that has special meaning to the command shell you are using, the character must be preceded by an escape character so that it will not be interpreted by the command shell. This is true even if the character is in the acceptable character range as documented in Appendix C, "ASCII characters from 33 to 126," on page 133.

For example, on AIX, if you use the ` character when specifying the salt value (using the -s parameter), you must precede the ` character with the \ character.

On AIX, Linux, Solaris, and HP-UX systems only, after you run the **idsgendirksf** utility, 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>.

Options

-b <*outputfile*>
Specifies the full path of a file to redirect console output into. Only errors are sent to the file if used in conjunction with the **-q** option. If debugging is turned on, that output is sent to this file also.

-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. This parameter is for use by IBM service personnel. See Chapter 4, "Debugging levels," on page 127 for additional information on debug levels.

-e ?|<*encryption seed*>
Specifies the encryption seed value that was used to create the original directory key stash file of the server. The encryption seed must only contain printable ISO-8859–1 ASCII characters with values in the range of

33 to 126, and must be a minimum of 12 and a maximum of 1016 characters in length. See Appendix C, “ASCII characters from 33 to 126,” on page 133. Use the **?** to generate a password prompt. Using this prompt prevents your encryption seed from being visible through the **ps** command.

Note: The encryption seed has the following requirements:

- Minimum number of alphabetic characters
- Minimum number of numeric and special characters
- Maximum number of repeated characters

-l <*location*>
Specifies the location to create the directory key stash file in.

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

-q Specifies to run in quiet mode. All output except errors messages are suppressed. If the **-d** option is also specified, trace output is not suppressed.

-s <*encryptionsalt*>
Specifies the encryption salt value used to create the directory key stash file of the server. The salt value can be obtained by searching the server’s ″cn=crypto,cn=localhost″ entry. The attribute value is ibm-slapdCryptoSalt. The encryption seed must only contain printable ISO-8859–1 ASCII characters with values in the range of 33 to 126, and must be 12 characters in length. See Appendix C, “ASCII characters from 33 to 126,” on page 133.

-v Specifies to display version information about the command.

-? Displays the syntax format.

Examples

To regenerate the key stash file for the directory server instance, myinstance, issue the command:

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

Then copy the generated **ibmslapddir.ksf** file and paste it in the idsslapd-myinstance/etc directory.

idsicrt

Command to create a directory server instance.

Synopsis

```
idsicrt [–I instancename [–e encryptionseed] [-g encryptsalt] [-p port]
        [-s secureport] [-a admport] [-t dbinstance] [-c admsecureport]
        [-i ipaddress] [-l instlocation][-r description] [-C] [-d debuglevel]
        [-b outputfile] [-q] [-n] [-x]] | -v | -?
```

Description

The idsicrt command can only be run by root on AIX, Linux, Solaris, or HP-UX platforms, or a member of the Administrators group on Windows platforms. The administrator specifies a directory server instance name and optionally can specify the port, secure port, admin daemon port, admin and daemon secure port. If these ports are not specified, then the first available port starting from #389 to #636 is selected for Tivoli Directory Server and the secure port, where # takes values from 1 to 65. For admin daemon, ports that are in the range 3538 to 65535 are selected. The **-e** option does not have to be specified, however, the encryption seed is required and you are prompted to supply one. On Windows, the administrator must specify the location to store the directory server instance. On AIX, Linux, Solaris, or HP-UX platforms, specifying the location is optional.

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

If a DB2 database instance already exists on the system, that DB2 instance is used. However, if the DB2 database instance is used by another directory server instance, the command will fail. This can be checked via the directory server instance repository and then looking at each directory server instance's configuration file.

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

Note: No database instance is created if the server component (RDBM) is not installed.

Attention: When you create a new 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 directory server instance, you must synchronize the encryption keys on the server instances ***before*** you do any of the following because the server will generate 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

See Appendix A, "Synchronizing two-way cryptography between server instances," on page 129 for information about synchronizing directory server instances.

Options

-a <*adminport*>
Specifies the port that the IBM directory server instance's administration daemon listens on. Specify a positive number that is greater than 0 and less than 65535. The ports specified must not cause a conflict with ports being used by other applications or operating systems, or any 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 into. Only errors are sent to the file if used in conjunction with the **-q** option. If debugging is turned on, that output is sent to this file also.

-c <*adminsecureport*>
Specifies the secure port that the IBM directory server instance's administration daemon listens on. Specify a positive number that is greater than 0 and less than 65535. The ports specified must not cause a conflict

with ports being used by other applications or operating systems, or any 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 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. This parameter is for use by IBM service personnel. See Chapter 4, "Debugging levels," on page 127 for additional information on debug levels.

-e <*encryptseed*>
Specifies the seed to be used to create the key stash files for a particular directory server instance. This option is required for use with the -n option. If not specified, you will be prompted for an encryption seed. The encryption seed must only contain printable ISO-8859–1 ASCII characters with values in the range of 33 to 126, and must be a minimum of 12 and a maximum of 1016 characters in length. See Appendix C, "ASCII characters from 33 to 126," on page 133.

Note: The encryption seed has the following requirements:

- Minimum number of alphabetic characters
- Minimum number of numeric and special characters
- Maximum number of repeated characters

-g <*encryptsalt*>
Specifies the encryption salt value. Providing an encryption salt value is useful if you want to use replication, use a distributed directory, or import and export LDIF data between server instances. You can obtain better performance if the two directory server instances have the same encryption salt value. Therefore, if the directory server instance you are migrating will be used in one of these ways, set the encryption salt value to the encryption salt value of the directory server instances with which it will be involved in these activities.

If you do not specify an encryption salt, the command randomly generates one.

The encryption salt must have exactly 12 characters and can contain only printable ISO-8859-1 ASCII characters in the range from 33 to 126 inclusive. For information about the characters that can be used, see Appendix C, "ASCII characters from 33 to 126," on page 133.

-i <*ipaddress*>
Specifies the IP address that the directory server instance binds to. If more than one IP address is specified, the comma separator is required with no spaces. Spaces are allowed only if the entire argument is surrounded in quotes. Use the key word ″all″ to specify to use all available IP addresses. All available IP addresses is the default setting, if you do not specify the **-i** option.

-I <*instancename*>
Specifies the instance name to be created for the directory server instance. The instance name must be an existing user ID on the machine and must be no greater than 8 characters in length.

-l <*instancelocation*>
Specifies the location to store the directory server instance's configuration files and logs. On Windows systems, this option is required and a drive letter must be specified. This location needs to have at least 30 MB of free space. Additional disk space needs to be available to accommodate growth as the directory server log files increase.

-n Specifies to run no prompt mode. All output 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 ports specified must not cause a conflict with ports being used by other applications or operating systems, or any other directory server instance that is bound to a particular host name or IP address.

-q Specifies to run in quiet mode. All output except errors messages are suppressed. If the **-d** option is also specified, 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 ports specified must not cause a conflict with ports being used by other applications or operating systems, or any 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 display version information about the command.

-x Create a proxy directory server instance. If this option is not given, then a full directory server instance with a DB2 instance will be created.

-? Displays the syntax format.

Examples

- To create a new directory server instance called **myinst** that has a port of 389, a secure port of 636, an encryption seed of **mysecretkey!**, an encryption salt of **mysecretsalt**, and a DB2 instance with the name **myinst**, issue the command:

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

  If the directory server instance already existed, this command would fail. If you did not specify the encryption salt, the command would randomly generate an encryption salt. If you did not specify the encryption seed, you would be prompted for the seed. In the following example, you are prompted to enter an encryption seed. The encryption seed is not displayed 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 response is:

  ```
  Enter encryption seed:
  ```

- To create the same instance so that it binds to a particular IP address, issue the command:

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

- To create a new directory server instance called **myinst** that has a port of 389, a secure port of 636, an encryption seed of **mysecretkey!**, and a DB2 instance with the name **mydbin**, use the following command:

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

  In this case, the command will randomly generate an encryption salt value.

idsideploy

Command to create a directory server instance from an existing directory server instance.

Synopsis

```
idsideploy [options]
```

Description

You can use the **idsideploy** command to create a directory server instance that uses an existing directory server instance (on the local computer or on another computer) as a template. When you do this, the configuration settings and schema files from the source directory server instance are duplicated and the directory key stash files are also synchronized. The new directory server instance can be configured as a replica or a peer to the source directory server instance if it is in an existing replication deployment, as a full directory server instance that is not participating in replication, or as an additional proxy server. Requirements are:

- The source directory server instance must be running IBM Tivoli Directory Server version 6.1; it cannot be running an earlier version of IBM Tivoli Directory Server, and it cannot be running another version of LDAP.
- The source directory server instance must be running, 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 the directory server instance you are creating will be a peer or replica, there must be a replication context 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 already have at least one replication context, replication group, and replication subentry defined. If a replica is being configured, the source directory server instance must already have the initial replication topology defined, including an agreement to at least one other server. If a peer is being configured, the source server must be defined as a master for one or more of the subentries in the replication configuration.
- If the directory server instance you are creating will be a peer or replica, a new replication subentry will be created under ibm-replicaGroup=default,<*replContext*> DN. If this DN is not present, the instance cannot be copied.

The new directory server instance will be created on the computer where the **idsideploy** command is running. If the source directory server is on a different computer, the operating systems of the two computers need not be the same. 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 will also copy the key database files if the source directory server is running under SSL mode and the **idsideploy** command is connected to the source directory server using SSL communication.

If the directory server instance you are copying is a proxy server, the new directory server instance will also be a proxy. If the directory server instance you are copying is a full server, the new directory server instance will also be a full server, and you can choose whether or not you want to copy the data from the existing directory server instance.

Note: If you want to copy the data from the existing directory server instance while creating the new directory server 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, however, can be different.
- The source directory server instance must be configured to allow for online backups.
- An initial offline backup must have been taken on the source directory server instance at some time before you use the **idsideploy** command to copy the directory server instance. The path you specify must contain only one backup image.
- The path where the backup images are stored must be accessible to both the source directory server instance and the new directory server instance.

See the *IBM Tivoli Directory Server Version 6.1 Installation and Configuration Guide* for information about preparing the source instance for copying the data.

Options

-a
Specifies the instance owner’s password. This will be used for database configuration and is required when the –x option is not given. This option is not allowed when the –x option is given.

-b <*outputfile*>
Specifies the full path of a file in which to redirect output. If this option is used in conjunction with the -q option, only errors will be sent to the file. If debugging is turned on, debug output will be sent to this file also.

-d <*debuglevel*>
Sets the debug level in the LDAP library. Use in conjunction with the ldtrc command.

-e <*encryptseed*>
Specifies the encryption seed for the new directory server instance. This must match the value given for the source directory server instance.

-D <*DN*>
The directory administrator distinguished name (DN) for the new directory server instance.

-I <*instancename*>
Specifies the name of the directory server instance to create. The instance name must be an existing userID on the machine and must be no greater than 8 characters in length.

-l <*instlocation*>
Specifies the location to store the directory server instance’s configuration

files and logs. On Windows, this option is required and a drive letter must be specified. There must be at least 30 MB free. If the directory server is not a proxy, this location will also be the DB2 database location and must have at least 80 MB free. Additional disk space must be available to accommodate growth as directory entries are added.

-L <*directoryPath*>
Specifies to load the data from the source directory server instance into the newly created directory server and gives the directory path for the backup images. This option is not allowed if the -x option is given and must be given with the -r and -p options.

-K <*keyfile*>
Specifies the file to use for keys for an SSL connection.

-n Run in no-prompt mode. All output is generated except for messages requiring user interaction.

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

-p Perform the restore of the database on the new directory server instance. If the instance name given by the -I option already exists, the idsideploy command must have been run before to create this instance and back up the source server. The -L option is required if this option is specified.

-q Run in quiet mode. All output except errors are suppressed. If the -d option is also specified, trace output is not suppressed.

-r <*peer* | *replica*>
Specifies to configure the new directory server instance in a replication environment as either a peer or replica. This option is not allowed if the –x option is given. The only valid values with this option are 'peer' and 'replica'.

-sU <*LDAP URL*>
Specifies the LDAP URL for the source directory server instance.

-sD <*DN*>
Specifies the adminDN for the source directory server instance.

-sw <*pw*>
Specifies the admin password for the source directory server instance.

-v Prints the version information about the command.

-w <*password*>
Specifies the password for the new directory server instance admindDN.

-x Specifies to create a proxy directory server instance, without a DB2 database instance. The source server must also be configured as a proxy server. This option is not allowed with the -a, -L, -p or -r options.

-? Displays the syntax format.

Note: If the **idsideploy** command is run only for the purpose of restoring a database using the – p option, then it is essential to set the DB2INSTANCE environment variable to the database instance name attached to the Tivoli Directory Server instance. Otherwise, idsideploy will fail.

Examples

Given below is an example of how a new directory server instance is created from an existing directory server instance:

```
idsideploy -sU ldap://<host>:<port> -sD <admin DN> -sw <adminPWD>
-e <encryptseed> -I <instname> -a inst123 -D cn=<adminDN> -w <adminPWD> -l <instlocation>
–b <outputfile> -q -L <directory path>
```

To create a standalone directory server instance from an existing directory server instance issue the following command. This command clones the database also.

```
idsideploy -sU ldap://<host>:<port> -sD <adminDN> -sw <adminPWD>
-e <encryptseed> -I <instname> -a inst123 -D <adminDN> -w <adminPWD> -l <instlocation>
–b <outputfile> -q -L <directory path>
```

To create a standalone directory server instance from an existing directory server instance issue the following command. This command does not clone the database.

```
idsideploy -sU ldap://<host>:<port> -sD <adminDN> -sw <adminPWd>
-e <encryptseed> -I <instname> -a inst123 -D <adminDN> -w <adminPWD> -l <instlocation>
–b <outputfile>
```

To create a peer in an existing replication setup, issue the following command:

```
idsideploy -sU ldap://<host>:<port> -sD <adminDN> -sw <adminPWD>
-e <encryptseed> -I <instname> -a inst123 -D <adminDN> -w <adminPWD> -l <instlocation>
–b <outputfile> -L <directory path> -r peer
```

To deploy a proxy instance under SSL mode, issue the following command:

```
idsideploy -sU ldaps://<host>:<port> -sD <adminDN> -sw <adminPWD>
-e <encryptseed> -I <instname> -K <kdb file> -P <kdb file pwd> -N <certificate name>
-D <adminDN> -w <adminPWD> -x -l <instlocation>
```

idsidrop

Command to delete a directory server instance.

Synopsis

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

Description

The **idsidrop** command can only be run by root on UNIX or a member of the Administrators group on Windows. 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 that directory server instance is stopped.

Options

-b <*outputfile*>
Specifies the full path of a file to redirect console output into. Only errors are sent to the file if used in conjunction with the **-q** option. If debugging is turned on, that output is sent to this file also.

-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. This

parameter is for use by IBM service personnel. See Chapter 4, "Debugging levels," on page 127 for additional information on debug levels.

-I <*instancename*>
Specifies the name of the directory server instance. This option is required if there are additional directory server instances on the local machine.

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

-q Specifies to run in quiet mode. All output except errors messages are suppressed. If the **-d** option is also specified, trace output is not suppressed.

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

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

-v Specifies to display version information about the command.

-? Displays the syntax format.

Examples

To remove a directory server instance and retain the associated database instance, issue the command:

```
idsidrop -I <instancename>
```

To remove a directory server instance and destroy the associated database instance, issue the command:

```
idsidrop -I <instancename> -r
```

To unconfigure the associated database instance without removing a directory server instance, issue the command:

```
idsidrop -I <instancename> -R
```

idsilist

Command to list directory server instances on the machine.

Synopsis

```
idsilist [[-a | -r] [-d debuglevel] [-b outputfile]] | -v | -?
```

Description

The **idsilist** command can only be run by root on UNIX or a member of the Administrators group on Windows by default. The command lists all of the directory server instances that exist on the machine. The command can also retrieve detailed information about each instance.

Note: You may manually change the permissions on the directory instance repository files to allow the command to be run by other users. However, only users with the ability to read all of the ibmslapd.conf files of all directory server instances on the machine are able to run the command successfully.

Options

-a <*outputfile*>
: Specifies to list the full information about each instance. This option cannot be used with the **-r** option.

-b <*outputfile*>
: Specifies the full path of a file to redirect console output into. If debugging is turned on, that output is sent to this file also.

-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. This parameter is for use by IBM service personnel. See Chapter 4, "Debugging levels," on page 127 for additional information on debug levels.

-r Specifies to list the full information about each instance. This provides the same information as the **-a** option, but the information is printed in a raw format. The information about each instance is printed on an individual line and each data item is separated by a number sign (#). This option cannot be used with the **-a** option.

-v Specifies to display version information about the command.

-? Displays the syntax format.

Examples

To get a list of directory server instances (in this example two) residing on the machine, issue the command:

```
idsilist
```

The output is:

```
Directory server instances:
myinst1
myinst2
```

To obtain information about each instance, issue the same command with the -a or -r options

```
idsilist -a
```

This command lists the directory server instances with their respective versions:

```
Instance 1:

Name: myinst1
Version: 6.1
Location: c:
Description: IBM Tivoli Directory Server Instance V6.1
IP Addresses: All available
Port: 389
Secure Port: 636
Admin Daemon Port: 3538
Admin Daemon Secure Port: 3539
Type: Directory Server

Instance 2:

Name: myinst2
Version: 6.1
Location: c:
Description: IBM Tivoli Directory Server Instance V6.1
```

```
IP Addresses: All available
Port: 389
Secure Port: 636
Admin Daemon Port: 3538
Admin Daemon Secure Port: 3539
Type: Proxy Server
```

```
idsilist -r
```

The output is:

```
Directory server instances:
myinst1#6.1#c:#IBM Tivoli Directory Server Instance V6.1#All available
#389#636#3538#3539#Directory Server
myinst2#6.1#c:#IBM Tivoli Directory Server Instance V6.1#All available
#389#636#3538#3539#Proxy Server
```

Notes:

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″ means that the directory server instance binds to all IP addresses. If there directory server instance only binds to certain IP addresses, a list is presented, separated by commas. For example,

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

idsimigr

Synopsis

The syntax for the **idsimigr** command is as follows:

```
idsimigr [–I instancename] [-t dbinstance] [-u backupdir]
 [-e encryptseed] [-g encryptsalt][-p port] [-s secureport]
[-a admport] [-c admsecureport] [-i ipaddress] [-r description]
[-b outputfile] [-d debuglevel] [-l instlocation] [–q] [-n] | [-v] | [-?]
```

Description

The **idsimigr** migration utility migrates the schema and configuration files from an earlier release to IBM Tivoli Directory Server 6.1 versions of these files and creates a directory server instance with the migrated information. This directory server instance is the upgraded version of your previous server. If required, can use the Instance Administration Tool, specifying that you want to migrate from a previous release. For more information about Instance Administration Tool, see "Creating and administering instances" in the *IBM Tivoli Directory Server Version 6.1 Installation and Configuration Guide*.

Attention: When you create a new directory server instance, be aware of the information that follows.

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

   If you are creating a directory server instance that must be cryptographically synchronized with an existing directory server instance, you must synchronize the server instances ***before*** you do any of the following:

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

   You can synchronize the server instances by ensuring that the encryption salt value for the server instance you are creating is the same as that of the existing server instance instance. You can obtain the destination server's salt value by searching for the ibm-slapdCryptoSalt attribute value (using the **idsldapsearch** utility) in the destination server's 'cn=crypto,cn=localhost' entry. See Appendix A, "Synchronizing two-way cryptography between server instances," on page 129 for information about synchronizing directory server instances.
2. After you create a directory server instance and configure the database, use the **idsdbback** utility to create a backup of the directory server instance. The configuration and directory key stash files are archived along with the associated configuration and directory data. You can then use the **idsdbrestore** utility to restore the key stash files if necessary. (You can also use the **idsdbback** utility after you load data into the database. See "Backing up the database" and "Backing up, restoring, and optimizing the database" in the *IBM Tivoli Directory Server Version 6.1 Installation and Configuration Guide* for information about backing up the database.)

Options

-? Displays usage help for the command.

-a *admport*

Specifies the port on which the administration daemon for the directory server instance will listen.

Note: If you have two or more directory server instances listening on the same IP address (or set of IP addresses), be sure that those directory server instances do not use any of the same port numbers.

-b *outputfile*

Specifies the full path of a file to redirect output into. If used in conjunction with the **-q** option, only errors are written to the file. If debugging is turned on, debugging information is also sent to the file.

-c *admsecureport*

Specifies the secure port on which the administration daemon for the directory server instance listens. Specify a positive number that is greater than 0 and less than or equal to 65535. The port specified must not cause a conflict with ports being used by any other directory server instance that is bound to a particular hostname or IP address.

-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. This

parameter is for use by IBM service personnel. See Chapter 4, "Debugging levels," on page 127 for additional information on debug levels.

-e *encryptseed*

Specifies the seed to be used to create the key stash files for the directory server instance. This option is required if you use the **-n** option. If it is not specified, you will be 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, and must be a minimum of 12 and a maximum of 1016 characters in length. For information about the characters that can be used, see Appendix C, "ASCII characters from 33 to 126," on page 133.

This encryption seed is used to generate a set of Advanced Encryption Standard (AES) secret key values. These values are stored in a directory stash file and used to encrypt and decrypt directory stored password and secretkey attributes. There is one encryption seed string for each directory server instance.

Record the encryption seed in a secure location; you might need it if you export data to an LDIF file (the **idsdb2ldif** command) or regenerate the key stash file (the **idsgendirksf** command.)

-g *encryptsalt*

Specifies the encryption salt value. Providing an encryption salt value is useful if you want to use replication, use a distributed directory, or import and export LDIF data between server instances. You can obtain better performance if the two directory server instances have the same encryption salt value. Therefore, if the directory server instance you are migrating will be used in one of these ways, set the encryption salt value to the encryption salt value of the directory server instances with which it will be involved in these activities.

If you do not specify an encryption salt, the command randomly generates one.

The encryption salt must have exactly 12 characters and can contain only printable ISO-8859-1 ASCII characters in the range from 33 to 126 inclusive. For information about the characters that can be used, see Appendix C, "ASCII characters from 33 to 126," on page 133.

-i *ipaddress*

Specifies the IP address that the directory server instance binds to. If more than one IP address is specified, a comma separator is required with no spaces. Spaces are allowed only if the entire argument is enclosed in quotation marks ("). Use the key word "all" to specify that you want to use all available IP addresses. If you do not specify the **-i** option, all available IP addresses is the default setting.

-I *instancename*

Specifies the name of the directory server instance to be created or migrated. The instance name must be an existing user ID on the computer and must be no greater than 8 characters in length. If there is no corresponding user ID for the directory server instance name, the command fails. See " Setting up users and groups: directory server instance owner, database instance owner, and database owner" in the *IBM Tivoli Directory Server Version 6.1 Installation and Configuration Guide* for information about additional requirements for the instance name.

-l *instlocation*
Specifies the location in which to store the configuration files and logs for the directory server instance. On Windows systems, this option is required and a drive letter must be specified. The location must have at least 30 MB of free disk space. Additional disk space must be available to accommodate growth as directory server log files increase in size.

-n Specifies that you want the command to run without prompting. All output is generated except for messages that require user interaction.

-p *port* Specifies the port on which the directory server instance listens. Specify a positive number that is greater than 0 and less than or equal to 65535. The port specified must not cause a conflict with ports being used by any other directory server instance that is bound to a particular hostname or IP address.

-q Specifies to run in quiet mode. All output is suppressed except error messages. If the **-d** option is also specified, 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 or equal to 65535. The port specified must not cause a conflict with ports being used by any other directory server instance that is bound to a particular hostname or IP address.

-t *dbinstance*
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.

-u *backupdir*
Specifies the name of the directory in which the schema and configuration files to be migrated have been saved.

If all the necessary files are not found in the specified directory, the command will fail. These files include the server configuration file and the following schema files: V3.ibm.at, V3.ibm.oc, V3.system.at, V3.system.oc, V3.user.at, V3.user.oc, and V3.modifiedschema.

-v Prints version information about the command.

Examples

For example, you want to migrate from IBM Tivoli Directory Server 5.2 to IBM Tivoli Directory Server 6.1 and:

- You saved the configuration and schema files in a directory named /tmp/ITDS52
- You want to create an instance called **myinst** with an encryption seed of **my_secret_key!** and an encryption salt of **mysecretsalt**

Use the following command:

```
idsimigr –I myinst –u /tmp/ITDS52 –e my_secret_key! -g mysecretsalt
```

On Windows, you must specify a location for the directory server instance using the **-l** option. The following example creates a c:\idsslapd-myinst directory for the directory server instance being migrated.

```
idsimigr –I myinst –u c:\temp –l c: -e my_secret_key!
```

idsldif2db, ldif2db

Command to load LDIF file entries into a database.

Synopsis

```
idsldif2db | ldif2db [-i inputfile -I instancename [-f configfile] [-d debuglevel]
        [-r yes | no] [-g] [-W]] | [?]
```

Description

This program is used to load entries specified in text LDAP Directory Interchange Format (LDIF) into a directory. The database must already exist. **idsldif2db** can be used to add entries to an empty directory database or to a database that already contains entries.

Notes:

1. The server must be stopped before using the server import utilities.
2. Ensure that no applications are attached to the directory database. If there are applications attached, none of the server utilities will run.
3. If you have installed Tivoli Directory Server 6.0 or later versions over a 5.2, 5.1, or a 4.1 server, you must initially start the server before using the **idsldif2db** utility so that one-time migration processing can be completed.
4. When records are added using **idsldif2db**, the master server must be stopped and then restarted immediately.
5. The **idsldif2db** utility recognizes the operational attributes **creatorsname**, **modifiersname**, **modifytimestamp**, and **createtimestamp** if they are in plain text format.

All other command line inputs result in a syntax error message, after which the correct syntax is displayed.

Attention: If you want to import LDIF data from another server instance, you must cryptographically synchronize the LDIF import file with the server instance that is importing the LDIF file; otherwise any AES-encrypted entries in the LDIF file will not be imported. See Appendix A, “Synchronizing two-way cryptography between server instances,” on page 129 for information about synchronizing directory server instances.

Note: If the file was created using **idsdb2ldif**, the source server’s SHA-encoded directory encryption seed was written to the LDIF file for reference during import. For parsing purposes, this encryption seed reference is contained in a cn=crypto,cn=localhost pseudo entry that is informational only, and is not actually loaded as part of the import.

Options

All options are case insensitive.

-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. This parameter is for use by IBM service personnel. See Chapter 4, “Debugging levels,” on page 127 for additional information on debug levels.

-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.

-g Specifies not to strip the trailing spaces on attribute values.

-i <*inputfile*>
: Specify the name of the LDIF input file, containing directory entries in LDIF format. This option is required. If the file is not in the current directory, a full path and file name must be specified.

-I <*instancename*>
: Specifies the instance name for the directory server instance that is to be used.

-r [yes|no]
: Specifies whether to replicate. The default is **yes** which means entries are put into the Change table and are replicated when the server restarts.

-W <*outputfile*>
: Specifies the full path of a file in which to redirect output.

-? Displays the syntax format.

Examples

On AIX, Linux, Solaris, or HP-UX platforms, to load the sample.ldif included with the IBM Tivoli Directory Server from the IDS_LDAP_HOME/sbin directory, issue the command:

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

On Windows systems the command is:

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

idslogmgmt

This feature enables the IBM Tivoli Directory Server administrator to limit the size of log files. The **idslogmgmt** utility activates every 15 minutes, checks the log files sizes, and moves log files that exceed the maximum log size threshold into an archive file. The number of archived logs can also be limited. The configuration settings are located in the ibmslapd.conf configuration file in most cases, the exception being the administrative tools and the idslogmgmt log settings. This enables the log management settings to be configured via the Web Administration Tool. The **idslogmgmt** utility requires IBM Tivoli Directory Integrator to be installed. See the *IBM Tivoli Directory Server version 6.1 Installation and Configuration Guide* for more information about installing IBM Tivoli Directory Integrator.

Synopsis

The syntax for the **idslogmgmt** command is as follows:

```
idslogmgmt [–I instancename] [-t threshold size]
[-a archives][-p archive path]|[-?]
```

Description

You must launch **idslogmgmt** using a system startup script or manually activate the tool. Type the following at a command prompt:

```
idslogmgmt
```

After that, the **idslogmgmt** tool will activate automatically to do its job.

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

- Threshold size: IDSADM_SIZE_THRESHOLD
- Number of archives: IDSADM_ARCHIVES

The following values are the defaults:

- The default threshold is 10MB (IDSADM_SIZE_THRESHOLD=10)
- The maximum number of archive files is 3 (IDSADM_ARCHIVES=3)

The archived log files are located in the following directories and have the filename <*timestamp*>_idsadm.log:

- UNIX path: `/var/idsldap/V6.1`
- Windows path: `<`*`ldap_install_directory`*`>\var`

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

- Threshold size: IDSLMG_SIZE_THRESHOLD
- Number of archives: IDSLMG_ARCHIVES

The following values are the defaults:

- The default threshold is 10MB (IDSLMG_SIZE_THRESHOLD=10)
- The maximum number of archive files is 3 (IDSLMG_ARCHIVES=3)

The archived log files are located in the following directories and have the filename <*timestamp*>_idslogmgmt.log:

- UNIX path: `/var/idsldap/V6.1`
- Windows path: `<`*`ldap_install_directory`*`>\var`

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

- Threshold size: IDSLMG_SIZE_THRESHOLD
- Number of archives: IDSLMG_ARCHIVES

The following values are the defaults:

- The default threshold is 10MB (IDSLMG_SIZE_THRESHOLD=10)
- The maximum number of archive files is 3 (IDSLMG_ARCHIVES=3)

The archived log files are located in the following directories and have the filename <*timestamp*>_idslogmgmt.log:

- UNIX path: `/var/idsldap/V6.1`
- Windows path: `<`*`ldap_install_directory`*`>\var`

In addition to the tool's main log file, idslogmgmt.log file, there are two additional log files produced by the IBM Tivoli Directory Integrator tool:

- ibmdi.log
- idslogmgmtinit.log

If the directories mentioned previously are not created, then the additional logs are placed in the current working directory. The ibmdi.log and idslogmgmtinit.log are overwritten each time the idslogmgmt tool is executed. As a result, the size of these two log files can remain small.

Options

-? Displays usage help for the command.

-a <*archives*>
Specifies the maximum number of Tivoli Directory Server admin tool's archived log files.

-I <*instance name*>
Specifies the name of the Tivoli Directory Server instance that the tool will manage the logs for.

Note: If –I is specified then –t, –a and –p options cannot be specified and vice-versa.

-p <*archive path*>
Specifies the path where the archived Tivoli Directory Server admin tool's log files will be placed.

-t <*threshold size*>
Specifies the size threshold of the Tivoli Directory Server admin tools' log file that will trigger archiving.

idsperftune

The idsperftune command is used to tune directory server performance.

Synopsis

```
 idsperftune [-I instance name] –i <property file> [-s] [–B] [-A] [-m ] [-o] [-p port]
[-h host] [-f configfile] [-b outputfile] [-d debuglevel] [-u update] [-E entrycache size]
[-F filter cache size] [–v | -? ]
```

Description

The idsperftune command helps administrators achieve higher directory server performance by tuning various caches, db2 buffer pools, and db2 parameters. It may be used in basic mode (with -B option) at any time including before a new directory instance has been used or after a directory server has been in use for a long time and previous tuning has been done. The advanced mode (-A option) should be only after the directory instance has been subjected to a typical workload. The advanced tuning analyzes db2 performance metrics and makes recommendations for fine tuning database parameters.

Options

-i <*property file*>
Specifies the property file from which to read input tuning parameters.

-I <*name*>
Specifies the name of directory server instance to be tuned.

-s Sets the default values for total number of entries expected in the directory (TDS_TOTAL_ENTRY) and average entry size (TDS_AVG_ENTRY_SZ) in ldaptune_input.conf file based on current directory contents. If –s is specified with –B or –A then these default values will be set and used as input for tuning the directory. If –s is specified without –A or –B, then the default values will be set, but no tuning will be done.

-b <***output file***>
Specifies the full path of a file in which to redirect output.

-B Performs basic tuning of directory server caches and db2 buffer pools. If –B is supplied with –u then the new values computed by the tool for the ldap caches and db2 buffer pools will be updated in the server and database instance respectively. If –B is specified without –u then the recommended settings will be updated in the perftune_stat.log file only.

-A Performs advanced tuning of db2 configuration parameters. Advanced tuning depends on runtime data gathered using the db2 monitor facility, so it should not be used during initial tuning of a new directory instance. This option will collect db2 snapshot data to compute recommended adjustments for db2 configuration parameters. This option monitors the number of "Select SQL statements executed". If this value is less than the threshold value i.e 10,000, then an appropriate warning message will be displayed, indicating that advanced tuning operation can not be performed until the directory has been more heavily used. due to insufficient data available. If –A is used without the –u option then the recommendations health status of for the db2 configuration parameters will be logged in perftune_stat.log file. If –A is supplied with the –u option then the new db2 configuration parameter values as provided by the administrator in the perftune_input.conf file will be updated in the database instance. If –A option is supplied with –m option, db2 monitor switches BUFFERPOOL and SORT will be enabled to gather additional db2 runtime data and the advanced tuning operation will be performed after a time interval of 5 min. Monitor switches will be turned off after completion of the advanced tuning operation.

-u Performs update of db2 and directory server cache configuration settings. If –u is not specified then recommended settings are recorded in the ldaptune_stat.log property file and no configuration files are updated.

-m Turns on db2 monitor switches for BUFFERPOOL and SORT. Turning on these switches allows collection of data that enables more complete tuning of db2. Some degradation in performance will be observed while these switches are on. If –m is specified with –A then a database snapshot will be captured after a time interval of 5 minutes and the switches will then be turned off.

-o Turns off monitor switches for BUFFERPOOL and SORT.

-p *<port>*
LDAP server port number. This will be used to update the directory cache parameter using ldap exop operation.

-h *<host>*
LDAP server host name.

-d *<debuglevel>*
Sets debug level in LDAP library.

-f *<configfile>*
Specifies the full path of the configuration file to update. If not specified, the default configuration file for the specified directory server instance will be used.

-E *<entrycache size>*
This option may be used to override the target for percentage of total number of entriesy that could reside in the entry cache. By default 80% is used.

-F *<filtercache size>*
This option may be used to override the default filter cache size (number of search filters to be cached). The default value is 1000.

-v Print the version information about the command.

-? Displays the syntax format.

Examples

To perform basic tuning of the directory server, issue the following command:

```
idsperftune –I myinst –i  <property file> -B –u
```

Since the above command is specified using the –u option, the recommended ldap cache and db2 buffer pool values are updated in the server and database instance respectively. If specified without the –u option, then the recommended settings are updated in the perftune_stat.log file only.

To perform advanced tuning of the directory server, issue the following command:

```
idsperftune –I myinst –i  <property file>  -A –m
```

By using the –m option, monitor switches for BUFFERPOOL and SORT are turned on.

IDSProgRunner

The **IDSProgRunner** is called from the **idsxinst** and **idsxcfg** commands to spawn a long-running task to run in the background. The **idsxcfg** utility then exits, and other processes (including other instances of the **idsxcfg** utility) query the state and progress of the task during and after its running.

The **IDSProgRunner** is used instead of simply spawning the task directly for two reasons:

1. **IDSProgRunner** 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 (the **IDSProgRunner**) to be waiting for it at the time the task exits.
2. IDSProgRunner enables almost any process to run in the background. It also maintains the start and stop time, and PID of the process so that the task can be signaled or ended.

idsrunstats, runstats

Command to optimize the database for a directory server instance.

Synopsis

```
idsrunstats | runstats [-I instancename [-f configfile] [-d debuglevel]] | -v | -?
```

Description

The **idsrunstats** command updates statistics about the physical characteristics tables and the associated indexes in the database of the directory server instance. These characteristics include number of records, number of pages, and average record length. The optimizer uses these statistics when determining access paths to the data. This utility should be called when a table has had many updates, or after reorganizing a table.

Options

-I <*instancename*>
: Specifies the instance name for the directory server instance that is to be updated.

-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. This parameter is for use by IBM service personnel. See Chapter 4, "Debugging levels," on page 127 for additional information on debug levels.

-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.

-v Specifies to display version information about the command.

-? Displays the syntax format.

Examples

```
idsrunstats -I <instancename>
```

idssethost

Command to set the IP addresses a directory server instance binds to.

Synopsis

```
idssethost [-I instancename –i ipaddress [-d debuglevel] [-b outputfile] [-q]
          [-n]] | -v | -?
```

Description

The **idssethost** command can only be run by root on UNIX or a member of the Administrators group on Windows by default. You may manually change the permissions on the directory instance repository files to allow the command to be run by other users. However, only users with the ability to read all of the ibmslapd.conf files of all directory server instances on the machine are able to run the command successfully.

This command sets the IP addresses that a particular directory server binds to. The administrator specifies a directory server instance name and a list of IP addresses. The directory server instance and the admin daemon of the directory server instance being updated is running must be stopped. The **idssethost** does not allow the IP addresses to be changed, if another directory server instance is using any of the same ports on the specified IP addresses. The command replaces all of the current IP addresses 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.

Options

-b <*outputfile*>
: Specifies the full path of a file to redirect console output into. Only errors are sent to the file if used in conjunction with the **-q** option. If debugging is turned on, that output is sent to this file also.

-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. This parameter is for use by IBM service personnel. See Chapter 4, “Debugging levels,” on page 127 for additional information on debug levels.

-i <*ipaddress*>
Specifies the IP address that the directory server instance binds to. If more than one IP address is specified, the comma separator is required with no spaces. Spaces are allowed only if the entire argument is surrounded in quotes. Use the key word ″all″ to specify to use all available IP addresses. All available IP addresses is the default setting, if you do not specify the **-i** option.

-I <*instancename*>
Specifies the instance name for the directory server instance that is to be updated.

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

-q Specifies to run in quiet mode. All output except errors messages are suppressed. If the **-d** option is also specified, trace output is not suppressed.

-v Specifies to display version information about the command.

-? Displays the syntax format.

Examples

To update the IP addresses of the directory server instance myinst to only bind to 1.3.45.668, issue the command:

```
idssethost -I myinst –i 1.3.45.668
```

To update the IP addresses of the directory server instance myinst to bind to all available IP addresses, issue the command:

```
idssethost -I myinst –i all
```

Note: You can also change the host name using the **idsldapmodify** command or the Web Administration tool. However, the modify command does fail, if the IP address specified is not valid on the machine. To ensure that there are no conflict with other ports on particular IP addresses, the IP address updates are done by the root administrator on the machine.

idssetport

Command to set the ports that a directory server instance binds to.

Synopsis

```
idssetport [-I instancename
         [-p port] [-s secureport] [-a admport] [-c admsecureport]
         [-d debuglevel] [-b outputfile] [-q] [-n]] | -v | -?
```

Description

The **idssetport** command can only be run by root on UNIX or Linux operating systems, or a member of the Administrators group on Windows by default. You may manually change the permissions on the directory instance repository files to allow the command to be run by other users. However, only users with the ability to read all of the ibmslapd.conf files of all directory server instances on the machine are able to run the command successfully.

The command sets the specified ports that a particular directory server binds to. The administrator specifies a directory server instance name and the ports to update. The directory server instance that is being updated must be stopped. If the admin daemon instance is running and an admin daemon instance port is changed, you must restart the admin daemon.

Options

-a <*adminport*>
Specifies the port that the IBM directory server instance's administration daemon listens on. Specify a positive number that is greater than 0 and less than 65535. The ports specified must not cause a conflict with ports being used by other applications or operating systems, or any 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 into. Only errors are sent to the file if used in conjunction with the **-q** option. If debugging is turned on, that output is sent to this file also.

-c <*adminsecureport*>
Specifies the secure port that the IBM directory server instance's administration daemon listens on. Specify a positive number that is greater than 0 and less than 65535. The ports specified must not cause a conflict with ports being used by other applications or operating systems, or any other directory server instance that is bound to a particular host name or IP address.

-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. This parameter is for use by IBM service personnel. See Chapter 4, "Debugging levels," on page 127 for additional information on debug levels.

-I <*instancename*>
Specifies the instance name for the directory server instance that is to be updated.

-n
Specifies to run no prompt mode. All output 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 ports specified must not cause a conflict with ports being used by other applications or operating systems, or any other directory server instance that is bound to a particular host name or IP address.

-q
Specifies to run in quiet mode. All output except errors messages are suppressed. If the **-d** option is also specified, trace output is not suppressed.

-s <*secureport*>
Specifies the SSL port.

-v Specifies to display version information about the command.

-? Displays the syntax format.

Examples

To update port of the directory server instance myinst to 555, issue the command:

```
idssetport -I myinst –p 555
```

Notes:

1. By default, all ports between 1 and 1024 including ports 389 and 636 can only be used by the root administrator on AIX, Linux, Solaris, and HP-UX platforms.
2. You can also change the host name using the idsldapmodify command or the Web Administration tool. However, the modify command does fail, if the IP address specified is not valid on the machine. To ensure that there are no conflict with other ports on particular IP addresses, the IP address updates are done by the root administrator on the machine.

idsslapd, ibmslapd

Command to start or stop the directory server daemon

Synopsis

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

Description

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

Options

-a Specifies to start the server in configuration only mode.

-c <*adminsecureport*>
Specifies the secure port that the IBM directory server instance's administration daemon listens on. Specify a positive number that is greater than 0 and less than 65535. The ports specified must not cause a conflict with ports being used by other applications or operating systems, or any other directory server instance that is bound to a particular host name or IP address.

-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 <*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. This parameter is for use by IBM service personnel. See Chapter 4, "Debugging levels," on page 127 for additional information on debug levels.

-h ? Displays the debug help screen.

-I <*instancename*>
Specifies the name of the directory server instance.

-k Specifies to stop the directory server deamon.

-n Specifies not to start the server in configuration only mode, if an error is encountered.

-p <*port*>
Specifies the non-SSL port.

-R *serverID*
Use serverID as the server ID while running this directory server instance.

-s <*secureport*>
Specifies the SSL port.

-v Specifies to print the version information.

-? Displays the syntax format.

The following parameters are for Windows systems only:

-i Specifies to install the admin daemon instance as a service.

-u Specifies to remove the admin daemon instance as a service.

The following parameter is for AIX, Linux, Solaris, and HP-UX systems only.

-c Specifies to run the server in console mode.

-t Specifies to tail the server log until final start-up messages are displayed on the console.

Examples

To start the directory server for the directory server instance, myinstance, issue the command:

```
idsslapd -I myinstance
```

To stop the directory server for the directory server instance, myinstance, issue the command:

```
idsslapd -I myinstance -k
```

idssnmp

For information about the idssnmp utility, see Appendix D, “Using the command line – idssnmp,” on page 135.

idssupport

For information about the idssupport utility, see ″IBM Tivoli Directory Server Support Tool″ in the *IBM Tivoli Directory Server Version 6.1 Problem Determination Guide* .

idsucfgchglg

Command to unconfigure a change log for a directory server instance.

Synopsis

```
idsucfgchglg [-I instancename [-f configfile] [-d debuglevel]
             [-b outputfile] [-q] [-n]] | -v | -?
```

Description

The **idsucfgchglg** command unconfigures a change log for a directory server instance. A change log must be currently configured in the ibmslapd.conf file. The directory server instance owner does not have to specify any parameters to have the change log removed and the change log information removed from the ibmslapd.conf file. The directory server instance owner is prompted to confirm the action before the change log is deleted.

Options

-b <*outputfile*>
: Specifies the full path of a file to redirect console output into. Only errors are sent to the file if used in conjunction with the **-q** option. If debugging is turned on, that output is sent to this file also.

-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. This parameter is for use by IBM service personnel. See Chapter 4, "Debugging levels," on page 127 for additional information on debug levels.

-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.

-I <*instancename*>
: Specifies the instance name for the directory server instance that is to be updated.

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

-q Specifies to run in quiet mode. All output except errors messages are suppressed. If the **-d** option is also specified, trace output is not suppressed.

-v Specifies to display version information about the command.

-? Displays the syntax format.

Examples

To unconfigure the directory server instance's change log and not prompt the user for confirmation, issue the command:

```
idsucfgchglg –n
```

To unconfigure the change log for the directory server instance, myinstance, on a machine with multiple instances, issue the command:

```
idsucfgchglg –I <myinstance>
```

idsucfgdb

Command to unconfigure a database for a directory server instance.

Synopsis

```
idsucfgdb [-I instancename [-r] [-f configfile] [-d debuglevel] [-b outputfile]
          [-q] [-n]] | -v | -?
```

Description

The **idsucfgdb** command unconfigures the database for a directory server instance. By default the database is only unconfigured from the ibmslapd.conf file and does not delete the database. To specify to delete the database during the unconfiguration process, the **–r** option can be specified. You are prompted to confirm that you want to continue with the requested actions.

Options

-b <*outputfile*>
: Specifies the full path of a file to redirect console output into. Only errors are sent to the file if used in conjunction with the **-q** option. If debugging is turned on, that output is sent to this file also.

-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. This parameter is for use by IBM service personnel. See Chapter 4, "Debugging levels," on page 127 for additional information on debug levels.

-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.

-I <*instancename*>
: Specifies the instance name for the directory server instance that is to be updated.

-n Specifies to run no prompt mode. All output is generated, except for messages that require user interaction. This option requires the **-w** option.

-q Specifies to run in quiet mode. All output except errors messages are suppressed. If the **-d** option is also specified, trace output is not suppressed.

-r Specifies to destroy any database currently configured with the directory server instance.

-v Specifies to display version information about the command.

-? Displays the syntax format.

Examples

To unconfigure the directory server instance's database and not prompt the user before unconfiguring it, issue the command:

```
idsucfgdb -n
```

To unconfigure and delete the directory server instance's database and not prompt the user for the confirmation before removing the directory server instance, issue the command:

```
idsucfgdb –r –n
```

idsucfgsch

Command to unconfigure a schema file for a directory server instance.

Synopsis

```
idsucfgsch [-I instancename -s schemafile [-f configfile] [-d debuglevel]
          [-b outputfile] [-q] [-n]] | -v | -?
```

Description

The **idsucfgsch** unconfigures a schema file for a directory server instance. The schema file must be currently configured in the directory server instance's ibmslapd.conf. The directory server instance owner must specify the schema file to remove the file from directory server instance's ibmslapd.conf file.

Options

-b <*outputfile*>
: Specifies the full path of a file to redirect console output into. Only errors are sent to the file if used in conjunction with the **-q** option. If debugging is turned on, that output is sent to this file also.

-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. This parameter is for use by IBM service personnel. See Chapter 4, "Debugging levels," on page 127 for additional information on debug levels.

-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.

-I <*instancename*>
: Specifies the instance name for the directory server instance that is to be updated.

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

-q Specifies to run in quiet mode. All output except errors messages are suppressed. If the **-d** option is also specified, trace output is not suppressed.

-s <*schemafile*>
: Specifies the schema file to remove from the directory server instance.

-v Specifies to display version information about the command.

-? Displays the syntax format.

Examples

To unconfigure the schema file /home/mydir/myschema.oc from the directory server instance's ibmslapd.conf file, issue the command:

```
idsucfgsch –s /home/mydir/myschema.oc
```

Note: The following system-defined schema files cannot be removed:

1. V3.system.at

2. V3.system.oc
3. V3.config.at
4. V3.config.oc
5. V3.ibm.at
6. V3.ibm.oc
7. V3.user.at
8. V3.user.oc
9. V3.ldapsyntaxes
10. V3.matchingrules

idsucfgsuf

Command to remove a suffix from a directory server instance.

Synopsis

```
idsucfgsuf [-I instancename -s suffix [-f configfile] [-d debuglevel]
         [-b outputfile] [-q] [-n]] | -v | -?
```

Description

The **idsucfgsuf** removes a suffix from a directory server instance. The suffix is removed from the directory server instance's ibmslapd.conf file. This command fails if the suffix does not exist in the configuration file.

Options

-b <*outputfile*>
Specifies the full path of a file to redirect console output into. Only errors are sent to the file if used in conjunction with the **-q** option. If debugging is turned on, that output is sent to this file also.

-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. This parameter is for use by IBM service personnel. See Chapter 4, "Debugging levels," on page 127 for additional information on debug levels.

-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.

-I <*instancename*>
Specifies the name of the directory server instance. This option is required if there are additional directory server instances on the local machine.

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

-q Specifies to run in quiet mode. All output except errors messages are suppressed. If the **-d** option is also specified, trace output is not suppressed.

-s <*suffix*>
Specifies to remove the suffix from the directory server instance.

-v Specifies to display version information about the command.

-? Displays the syntax format.

Examples

To remove the suffix o=sample from the ibmslapd.conf file on a machine with a single directory server instance, issue the command:

```
idscfgsuf -s o=sample
```

To remove the suffix o=sample from the ibmslapd.conf file of a directory server instance on a machine with a multiple directory server instances, issue the command:

```
idscfgsuf -I <instancename> -s o=sample
```

Note: These system defined suffixes cannot be removed:

- cn=pwdpolicy
- cn=localhost
- cn=configuration
- cn=ibmpolicies

ldtrc

The tracing utility. This utility is to be used in conjunction with IBM support to solve specific problems.

Synopsis

```
ldtrc (chg|clr|dmp|flw|fmt|inf|off|on) options
```

Description

The tracing utility, **ldtrc**, is used to activate or deactivate tracing of the Directory Server. To display syntax help for **ldtrc**, type: `ldtrc -?`

Note: The format and flow options require that the environment variable TRCTFIDIR be set to the directory containing the Trace Facility Information (*.tfi) files.

Options

chg | change
The trace must be active before you can use the **chg** option 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 sever errors (maxSevereErrors) is reached.
- [-this <thisPointer>] trace 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 as well as server debug messages. You can specify the name of the destination file where you want to dump the trace. The default destination files is:

For AIX, Linux, Solaris, and HP-UX systems:
/var/ldap/ibmslapd.trace.dump.

For Windows-based systems:
<*installationpath*>\var\ibmslapd.trace.dump

Note: This file contains binary ldtrc data that must be formated with the **ldtrc format** command.

flw | flow

- [-m <mask>] Where <mask> = <products>.<events>.<components>.<classes>.<functions>.
- [-p <pid>[.<tid>]] Shows control flow only the specified process or thread.
- [-r] Specifies to output trace in reverse chronological order.
- [-x <onlyRecord> | <firstRecord> - <lastRecord>] Shows the control flow only the specified record or show the control flow between the specified first and last records.
- [-this <thisPointer>] trace 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 this 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 the specified record or show the control flow between the specified first and last records.
- [-this <thisPointer>] trace 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 information about the trace. You must specify the source file which can be either a binary trace file, or trace buffer (if file is ″-″) and a destination file. The following is an example of the information that the **info** parameter gives:

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

on Turns on 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 sever 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>] trace only the specified object.
- [-perf] trace only performance records.

Note: The tracing facility must be on for server data to be traced.

off Turns off the tracing facility.

Examples

To turn the ldtrc facility on, issue the command:

```
ldtrc on
```

To turn off the ldtrc facility, issue the command:

```
ldtrc off
```

idsrun

The **idsrun** command is used by AIX, Linux, Solaris, and HP-UX systems. It is similar to IDSProgRunner, but it does not track the process it spawns. Instead, it just invokes the executable and exits. This program is used by the **idsdiradm** command to start a directory server and is used by the **idsicrt** command to start **idsdiradm**.

Chapter 4. Debugging levels

For all server utility debug options, the **ldtrc** utility must be running. The **ldtrc** utility is not required for the client utilities. For example to enable debugging the **idscfgdb** command for a directory server instance, myinstance, issue the commands:

```
ldtrc on
idscfgdb -I myinstance -d <debuglevel>
```

where the specified debug level value determines which categories of debug output are generated.

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
0x1000	8192	LDAP_DEBUG_RDBM	Relational backend activities (RDBM)
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, specifying a bitmask value of ″65535″ turns on full debug output and generates the most complete information.

When you are finished, issue the following command at a command prompt:

```
ldtrc off
```

Contact IBM Service for assistance with interpreting of the debug output and resolving of the problem.

Appendix A. Synchronizing two-way cryptography between server instances

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

If you already have a server instance, and you have another server instance that you want to cryptographically synchronize with the first server instance, use the following procedure ***before*** you do any of the following:

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

To cryptographically synchronize two server instances, assuming that you have already created the first server instance:

1. Create the second server instance, but do not start the server instance, run the **idsbulkload** command, or run the **idsldif2db** command on the second server instance.
2. Copy the ibmslapddir.ksf file (the key stash file) from the first server instance to the second server instance, overwriting the second server's original ibmslapddir.ksf file. The file is in the idsslapd-*instance_name*\etc directory on Windows systems, or in the idsslapd-*instance_name*/etc directory on AIX, Linux, Solaris, and HP-UX systems. (*instance_name* is the name of the server instance.)
3. Use the **idsgendirksf** utility to recreate the ibmslapddir.ksf file (the key stash file) from the first server instance. This file is used to replace the second server instance's original ibmslapddir.ksf file. For information about the **idsgendirksf** utility, see "idsgendirksf" on page 94. The file is in the idsslapd-*instance_name*\etc directory on Windows systems, or in the idsslapd-*instance_name*/etc directory on AIX, Linux, Solaris, and HP-UX systems. (*instance_name* is the name of the server instance).
4. Start the second server instance, run the **idsbulkload** command, or run the **idsldif2db** command on the second server instance.

The server instances are now cryptographically synchronized, and AES-encrypted data will load correctly.

Although the procedure discusses two server instances, you might need a group of server instances that are cryptographically synchronized.

Note: When importing LDIF data, if the LDIF import file is not cryptographically synchronized with the server instance that is importing the LDIF data, any AES-encrypted entries in the LDIF import file will not be imported.

Appendix B. IANA character sets supported by platform

The following table defines the set of 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 platform, and that all string content in the LDIF file is assumed to be represented in the specified charset. ″n/a″ indicates that the conversion is not supported for the associated platform.

String content is defined to be all attribute values that follow an attribute name and a single colon.

See IANA Character Sets for more information about IANA-registered character sets. Go to:

`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		
TIS-620	n/a	n/a	X	X	n/a	874	874

Table 2. IANA-defined character sets (continued)

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

The following table shows ASCII characters from 33 to 126. These are the characters that can be used in the encryption seed string.

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				

Appendix D. Using the command line – idssnmp

idssnmp has the following command line options:

-q This will not display the log messages to the screen. This is an optional parameter.

-v Displays the version number of the **idssnmp** tool. This is an optional parameter.

-? Displays the usage. This is an optional parameter.

When IBM Tivoli Directory Integrator ends, it returns one of the following exit codes:

0 User started IBM Tivoli Directory Integrator with **-v** parameter (show info and exit).

1

- Cannot open logfile (**-l** parameter)
- Cannot open configuration file
- Stopped by admin request

2 Exit after auto-run. When you start IBM Tivoli Directory Integrator specifying the **-w** option, IBM Tivoli Directory Integrator runs the AssemblyLines specified by the **-r** parameter and then exits.

9 License expired or invalid.

Appendix E. Notices

This information was developed for products and services offered in the U.S.A. IBM might 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 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 (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, 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 may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the information. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this information 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
Department MU5A46
11301 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.

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.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, or other countries, or both:

AIX
DB2
i5/OS
IBM
iSeries
Lotus
OMEGAMON
OS/400
Passport Advantage
pSeries
SecureWay
Tivoli
WebSphere
World Registry
xSeries
z/OS
zSeries

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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.

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

Other company, product, and service names may be trademarks or service marks of others.

Printed in USA

SC23-7834-00