

IBM Tivoli Directory Server



Administration Guide

Version 6.0

IBM Tivoli Directory Server



Administration Guide

Version 6.0

Note

Before using this information and the product it supports, read the general information under Appendix P, "Notices," on page 603.

First Edition (April 2005)

This edition applies to version 6, release 0, of the IBM Tivoli Directory Server and to all subsequent releases and modifications until otherwise indicated in new editions.

© Copyright International Business Machines Corporation 2005. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Preface	ix
Who should read this book	ix
Publications	ix
IBM Tivoli Directory Server library.	ix
Related publications	ix
Accessing publications online.	x
Ordering publications	x
Accessibility	x
Tivoli technical training.	x
Support information	xi
Conventions used in this book	xi
Typeface conventions	xi
Operating system differences.	xi

Part 1. Directory overview 1

Chapter 1. Defining a directory 3

Directory clients and servers	3
Directory security.	4

Chapter 2. The IBM Tivoli Directory Server 5

Chapter 3. Distinguished names (DNs) 11

Distinguished name syntax	11
DN escaping rules	12
Enhanced DN processing.	13

Part 2. Server Administration 15

Chapter 4. Directory administration daemon 17

Starting an instance of the directory administration daemon	17
Stopping an instance of the directory administration daemon	17

Chapter 5. Configuration only mode . . 19

Minimum requirements for configuration only mode	19
How to start in configuration only mode	19
Using Web Administration:	19
Using the command line:	19
How to verify that the server is running in configuration only mode	20
Using Web Administration:	20
Using the command line:	20

Chapter 6. Web Administration Tool graphical user interface (GUI) 21

Starting the Web Administration Tool.	21
Logging in to the console.	22

Logging on to the console as the console administrator.	22
Logging on to the console as the server administrator, a member of an administrative group or an LDAP user	22
Console layout	22
Logging off the console	23
Using tables in the Web Administration Tool	23
Table icons	24
Select Action drop-down menu.	24
Paging	25
Sorting	25
Finding.	25
Filtering	26

Chapter 7. Setting up the Web Administration Tool 27

Managing the console	27
Changing the console administrator login	27
Changing the console administration password	27
Adding, modifying, and removing servers in the console	27
Managing console properties	28

Chapter 8. Managing the IBM Directory schema 31

Common schema support	33
Object identifier (OID).	33
Working with object classes	33
Defining object classes.	33
Viewing object classes	34
Adding an object class.	35
Editing an object class.	36
Copying an object class	38
Deleting an object class	39
Working with attributes	39
The IBMAttributeTypes attribute type	40
Matching rules	41
Indexing rules	42
Viewing attributes	43
Adding an attribute	44
Editing an attribute.	46
Copying an attribute	47
Deleting an attribute	49
Attribute syntax	50
The subschema entries.	50
The IBMsubschema object class.	50
Schema queries	51
Dynamic schema	51
Access controls	52
Replication	52
Disallowed schema changes	52
Object classes.	52
Attributes	53
Syntaxes	61

Matching rules	61
Schema checking	61
Checking an entry against the schema	61
DEN schema support	62
iPlanet compatibility	63
Generalized and UTC time	64

Chapter 9. Basic server administration tasks 67

Changing an administrator distinguished name and password	67
Using Web Administration:	67
Using the command line:	67
Starting and stopping the server	68
Using Web Administration:	68
Using the command line or Windows Services icon:	69
Checking server status.	69
Using Web Administration:	69
Using the command line:	76
Managing server connections	82
Using Web Administration:	82
Using the command line:	83
Managing connection properties	84
Using Web Administration:	84
Using the command line:	85
Creating the administrative group	86
Enabling and disabling the administrative group	87
Adding members to the administrative group	88
Modifying an administrative group member	89
Removing a member from the administrative group	90
Managing unique attributes	91
Creating unique attributes	91
Removing an attribute from the list of unique attributes	92

Chapter 10. Setting server properties 95

Changing server ports and enabling language tags	95
Using Web Administration:	96
Using the command line:	96
Setting Performance	97
Using Web Administration:	97
Using the command line:	98
Setting Searches	99
Using Web Administration:	99
Using the command line:	100
Searching the directory with paging and sorting	101
Sorted search control	102
Simple paged results	103
Enabling and disabling event notification	104
Enabling event notification	105
Disabling event notification.	106
Enabling and disabling transaction support	106
Enabling transaction support	106
Disabling transaction support	108
Adding and removing suffixes	108
Creating or adding suffixes	108
Removing a suffix	109

Adding attributes to and removing attributes from the attribute cache	110
Setting up and adding attributes to the attribute cache	110
Removing attributes from the attribute cache	111

Chapter 11. Securing the directory 113

Configuring security settings	113
Using Web Administration:	113
Using the command line:	114
Transaction Layer Security	115
Secure Sockets Layer	115
Using gsk7ikm	121
Setting the key database.	129
Using Web Administration:	129
Using the command line:	130
Setting the level of encryption for SSL and TLS communications	130
Using Web Administration:	130
Using the command line:	131
Password encryption	132
Password policy attributes	134
Setting password policy	134
Summary of settings for an EAL4 secure configuration	134
Setting the administration password and lockout policy	136
Setting the user password policy	137
Unlocking administrative accounts	139
Password Guidelines	139
Setting user password lockout.	141
Using Web Administration:	141
Using the command line:	142
Setting user password validation	142
Using Web Administration:	142
Using the command line:	143
Setting Kerberos	144
Using Web Administration:	145
Using the command line:	146
Using Kerberos	146
Identity mapping for Kerberos	146
Certificate revocation verification	148
Using Web Administration:	149
Using the command line:	149
Configuring the DIGEST-MD5 mechanism.	150
Using Web Administration:	150
Using the command line:	150

Chapter 12. Referrals 153

Setting up referrals to other LDAP directories	153
Using the referral object class and the ref attribute	153
Binding with a distributed namespace	155
An example of distributing the namespace through referrals	155
Creating default referrals	157
Using Web Administration:	157
Using the command line:	158
Modifying referrals	159
Using Web Administration:	159

Using the command line:	160
Removing referrals	160
Using Web Administration:	160
Using the command line:	160
Chapter 13. Replication	161
Replication terminology	161
Replication topology	163
Overview of replication	164
Simple replication	164
Cascading replication.	165
Peer-to-peer replication	165
Gateway replication	166
Replication conflict resolution	167
Replication error handling	168
Replication agreements	169
Things to consider before configuring replication	170
Replicating schema and password policy updates	171
Creating a master-replica topology	172
Using Web Administration:	173
Using the command line:	178
Setting up a simple topology with peer replication	181
Using Web Administration:	181
Using the command line:	183
Creating a master-forwarder-replica topology.	186
Changing the replica to a forwarding server	187
Setting up a complex topology with peer replication	192
Using Web Administration:	192
Using the command line:	196
Summary of steps for creating a complex replication topology	204
Unconfiguring a master/replica configuration	205
Setting up a gateway topology	207
Using Web Administration:	208
Using the command line:	211
Recovery procedures	217
Required recovery information	217
Recovering from a single-server failure	219
Recovering from a catastrophic failure	220
Multi-threaded replication	221
Replication error table	221
Web Administration tasks for managing replication	222
Replicating subtrees	222
Working with credentials	224
Managing topologies	228
Modifying replication properties	237
Creating replication schedules	238
Managing queues	240
Command line tasks for managing replication	242
Specifying a supplier DN and password for a subtree	242
Viewing replication configuration information	242
Monitoring replication status	243
Creating gateway servers	245
Chapter 14. Distributed directories	247
The proxy server	247
Splitting data within a subtree based on a hash of the RDN using a proxy server.	247

The distributed directory setup tool	249
Synchronizing information	251
Partition entries	252
Setting up a distributed directory with a proxy server	252
Setting up the back-end servers	252
Setting up the proxy server.	254
Failover and load balancing	259
Failover between proxy servers	259
Setting up backup replication for a distributed directory with proxy servers	260
Server groups	261
Creating an LDIF file for your data entries	262
Setting up the replication topology	263
Setting up a topology for global policies	264
Setting up the proxy servers	264
Partitioning the data	265
Loading the partitioned data	265
Starting replication	266
Chapter 15. Logging Utilities	267
Default log paths	267
Log management tool	268
Default log management	268
Modifying default log settings.	269
Using the Web Administration Tool	269
Using the command line	269
Modifying administration daemon error log settings	270
Using Web Administration Tool	270
Using the command line:	271
Enabling the administration daemon audit log and modifying administration audit log settings	271
Using Web Administration Tool	272
Using the command line:	273
Disabling the administration daemon audit log	273
Using Web Administration:	273
Using the command line:	274
Enabling the audit log and modifying audit log settings	274
Using Web Administration	276
Using the command line:	277
Disabling the audit log	278
Using Web Administration:	279
Using the command line:	279
Modifying bulkload error log settings	279
Using Web Administration	279
Using the command line:	280
Modifying configuration tools log settings.	280
Using Web Administration	280
Using the command line	281
Modifying DB2 error log settings.	281
Using Web Administration	281
Using the command line:	282
Modifying lost and found log settings	283
Using Web Administration	283
Using the command line	283
Modifying the server error log.	284
Using Web Administration	284
Using the command line	285
Viewing logs	285

View logs using Web Administration	285
View logs using the command line	286

Part 3. Directory Management 291

Chapter 16. Working with directory entries 293

Browsing the tree	293
Adding an entry	294
Using Web Administration	294
Using the command line	295
Multiple values for attributes	295
Binary data for attributes	295
Using Web Administration	295
Using the command line	296
Language tags	297
Attributes that cannot have associated language tags	298
Language tag values for attributes	298
Searching for entries containing attributes with language tags	299
Removing a language tag descriptor from an entry	299
Deleting an entry	300
Using Web Administration	300
Using the command line	300
Modifying an entry	301
Using Web Administration	301
Using the command line	301
Copying an entry	302
Using Web Administration	302
Using the command line	302
Editing access control lists for an entry	303
Adding an auxiliary object class	303
Using Web Administration	303
Using the command line	304
Deleting an auxiliary object class	304
Using Web Administration	304
Using the command line	305
Searching the directory entries.	305
Search filters	305
Options	307

Chapter 17. Access control lists 309

Overview.	309
EntryOwner information	309
Access control information	309
The access control attribute syntax	310
Subject	311
Pseudo DNs.	312
Object filter	313
Rights	313
Propagation	315
Access evaluation	316
Working with ACLs	318
Using the Web Administration Tool utility to manage ACLs	318
Using the command line utilities to manage ACLs	324
Subtree replication considerations	328

Chapter 18. Groups and roles 329

Groups	329
Static groups	329
Dynamic groups	329
Nested groups	331
Hybrid groups	331
Determining group membership	331
Group object classes	334
Group attribute types.	334
Creating a static group entry	335
Creating a dynamic group entry	336
Creating a nested group entry.	337
Verifying the group task.	338
Managing members of group entries	339
Adding a member to a group entry	339
Removing a member from a group entry	339
Managing memberships for an entry	340
Adding a group membership	340
Removing a group membership from an entry	340
Editing a memberURL in a dynamic group	341
Roles	342

Chapter 19. Managing search limit groups 343

Creating a search limit group	343
Using Web Administration:.	343
Using the command line:.	345
Modifying a search limit group	345
Using Web Administration:.	345
Using the command line:.	345
Copying a search limit group	346
Using Server Administration:.	346
Using the command line:.	346
Removing a search limit group	346
Using Web Administration:.	346
Using the command line:.	346

Chapter 20. Managing a proxy authorization group. 347

Creating a proxy authorization group	347
Using Web Administration:.	348
Using the command line:.	348
Modifying a proxy authorization group	349
Using Server Administration:.	349
Using the command line:.	349
Copying a proxy authorization group	349
Using Server Administration:.	349
Using the command line:.	350
Removing the proxy authorization group	350
Using Web Administration:.	350
Using the command line:.	350

Part 4. User-related tasks 351

Chapter 21. Realms, templates, users, and groups 353

Creating a realm	353
Creating a realm administrator	353
Creating the realm administration group	353

Creating the administrator entry	354
Adding the administrator to the administration group	354
Creating a template	355
Adding the template to a realm	357
Creating groups	357
Adding a user to the realm	357
Managing realms	358
Adding a realm	358
Editing a realm	358
Removing a realm	359
Editing ACLs on the realm	359
Managing templates	359
Adding a user template	359
Editing a template	361
Removing a template	361
Editing ACLs on the template	361
Managing users	362
Adding users	362
Finding users within the realm	362
Editing a user's information	362
Copying a user	363
Removing a user	363
Managing groups	363
Adding groups	363
Finding groups within the realm	363
Editing a group's information	364
Copying a group	364
Removing a group	364

Part 5. Command line utilities . . . 365

Chapter 22. Command line utilities 367

Client utilities	368
idsdirctl, ibmdirctl	369
idsldapchangepwd, ldapchangepwd	370
idsldapdelete, ldapdelete	373
idsldapdiff, ldapdiff	377
idsldapexop, ldapexop	386
idsldapmodify, ldapmodify, idsldapadd, ldapadd	396
idsldapmodrdn, ldapmodrdn	402
idsldapsearch, ldapsearch	406
idsldaptrace, ldaptrace	415
tbindmsg	418
SSL, TLS notes	419
Server utilities	420
ddsetup	420
idsbulkload, bulkload	423
idscfgchglg	429
idscfgdb	431
idscfgsch	432
idscfgsuf	433
idsdbback, dbback	434
idsdbrestore, dbrestore	435
idsdb2ldif, db2ldif	436
idsdiradm, ibmdiradm	438
idsdnpw	440
idsgendirksf	441
idsicrt	442

idsidrop	445
idsilist	446
idsimigr	448
idsldif2db, ldif2db	448
idslogmgmt	449
idslink	450
IDSProgRunner	450
idsrunstats, runstats	451
idssethost	451
idssetport	452
idsldapd, ibmslapd	454
idssnmp	455
idssupport	455
idsucfgchglg	455
idsucfgdb	456
idsucfgsch	457
idsucfgsuf	459
ldtrc	460
runscript	462
Debugging levels	462

Part 6. Appendixes 465

Appendix A. Error codes 467

Appendix B. Object Identifiers (OIDs) and attributes in the root DSE 473

Attributes in the root DSE	473
OIDs for supported and enabled capabilities	475
OIDs for ACI mechanisms	477
OIDs for extended operations	478
OIDs for controls	479

Appendix C. LDAP data interchange format (LDIF) 481

LDIF example	481
Version 1 LDIF support	482
Version 1 LDIF examples	482
IANA character sets supported by platform	483

Appendix D. ASCII characters from 33 to 126. 487

Appendix E. IPv6 support 489

Appendix F. Simple Networking Management Protocol. 491

Logging	493
Using the command line – idssnmp	494

Appendix G. Password policy operational attributes 495

Password policy queries	495
Overriding password policy and unlocking accounts	496
Replicating password policy operational attributes	497
Forcing an add or update for an entry	498

Appendix H. IBM Tivoli Directory Server 6.0 required attribute definitions.	501
Appendix I. Synchronizing two-way cryptography between server instances	537
Appendix J. Filtered ACLs and non-filtered ACLs – sample LDIF file	539
Appendix K. IBM Tivoli Directory Server 6.0 configuration schema object classes and attributes	547
Configuration object classes	547
Configuration attributes	551
Dynamically-changed attributes	578
Appendix L. Audit format	583
Audit format for a server audit	583
Auditing server events	585
Notes	585
Audit format for an Admin Daemon audit	585
Appendix M. Distributed directory setup tool options	587
Appendix N. Setting up SSL security – SSL scenarios	589

Using HTTPS for the embedded version of WebSphere Application Server Version 5.1.1	589
Creating secure connections between IBM WebSphere Application Server, and the IBM Tivoli Directory server and the administration daemon	590
Setting up an SSL connection between a client and server	596

Appendix O. Support information . . . 599

Searching knowledge bases.	599
Search the information center on your local system or network.	599
Search the Internet	599
Obtaining fixes.	599
Contacting IBM Software Support	600
Determine the business impact of your problem	601
Describe your problem and gather background information	601
Submit your problem to IBM Software Support	601

Appendix P. Notices . . . 603

Trademarks	604
------------	-----

Glossary . . . 607

Glossary	607
----------	-----

Index . . . 613

Preface

This document contains the information that you need to administer the IBM® Tivoli® Directory Server.

Who should read this book

This book is intended for system administrators.

Publications

Read the descriptions of the IBM Tivoli Directory Server library, the prerequisite publications, and the related publications to determine which publications you might find helpful. After you determine the publications you need, see “Accessing publications online” on page x for information about accessing publications online.

IBM Tivoli Directory Server library

The publications in the IBM Tivoli Directory Server library are:

IBM Tivoli Directory Server Version 6.0 Release Notes

Contains information about the new features in the IBM Tivoli Directory Server Version 6.0 release.

IBM Tivoli Directory Server Version 6.0 Installation and Configuration Guide

Contains complete information for installing the IBM Tivoli Directory Server client, server, and Web Administration Tool. Includes information about migrating from a previous version of IBM Tivoli Directory Server or SecureWay® Directory.

IBM Tivoli Directory Server Version 6.0 Performance Tuning Guide

Contains information about tuning your server for better performance.

IBM Tivoli Directory Server Version 6.0 Administration Guide

Contains instructions for performing administrator tasks through the Web Administration Tool and the command line.

IBM Tivoli Directory Server Version 6.0 Plug-ins Reference

Contains information about writing server plug-ins.

IBM Tivoli Directory Server Version 6.0 C-Client SDK Programming Reference

Contains information about writing Lightweight Directory Access Protocol (LDAP) client applications.

IBM Tivoli Directory Server Version 6.0 Problem Determination Guide

Contains information about possible problems and corrective actions that can be tried before contacting Software Support.

IBM Tivoli Directory Server Version 6.0 Messages

Contains information about error messages that you might see.

Related publications

Information related to IBM Tivoli Directory Server is available in the following publications:

- IBM Tivoli Directory Server Version 6.0 uses the JNDI client from Sun Microsystems. For information about the JNDI client, refer to the *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>.

- The Tivoli Software Library provides a variety of Tivoli publications such as white papers, datasheets, demonstrations, redbooks, and announcement letters. The Tivoli Software Library is available on the Web at: <http://www.ibm.com/software/tivoli/library/>
- The *Tivoli Software Glossary* includes definitions for many of the technical terms related to Tivoli software. The *Tivoli Software Glossary* is available from the **Glossary** link on the left side of the Tivoli Software Library Web page <http://www.ibm.com/software/tivoli/library/>

Accessing publications online

The publications for this product are available online in Portable Document Format (PDF) or Hypertext Markup Language (HTML) format, or both in the Tivoli software library: <http://www.ibm.com/software/tivoli/library>

To locate product publications in the library, click the **Product manuals** link on the left side of the library page. Then, locate and click the name of the product on the Tivoli software information center page.

Product publications include release notes, installation guides, user's guides, administrator's guides, and developer's references.

Note: To ensure proper printing of PDF publications, select the **Fit to page** check box in the Adobe Acrobat Print window (which is available when you click **File** → **Print**).

Ordering publications

You can order many Tivoli publications online at the following Web site:

<http://www.elink.ibm.com/public/applications/publications/cgi-bin/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, see the following Web site for a list of telephone numbers:

<http://www.ibm.com/software/tivoli/order-lit/>

Accessibility

Accessibility features help a user who has 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 also can use the keyboard instead of the mouse to operate all features of the graphical user interface.

Tivoli technical training

For Tivoli technical training information, refer to the IBM Tivoli Education Web site: <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:

- Searching knowledge bases: You can search across a large collection of known problems and workarounds, Technotes, and other information.
- 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 these three ways of resolving problems, see Appendix O, “Support information,” on page 599.

Conventions used in this book

This reference uses several conventions for special terms and actions and for operating system-dependent commands and paths.

Typeface conventions

The following typeface conventions are used in this reference:

Bold Lowercase commands or mixed case commands that are difficult to distinguish from surrounding text, keywords, parameters, options, names of Java classes, and objects are in **bold**.

Italic Variables, titles of publications, and special words or phrases that are emphasized are in *italic*.

<*Italic*>

Variables are set off with < > and are in <*italic*>.

Monospace

Code examples, command lines, screen output, file and directory names that are difficult to distinguish from surrounding text, system messages, text that the user must type, and values for arguments or command options are in monospace.

Operating system differences

This book uses the UNIX[®] convention for specifying environment variables and for directory notation. When you are using the Windows[®] command line, replace *\$variable* with *%variable%* for environment variables and replace each forward slash (/) with a backslash (\) in directory paths. If you are using the bash shell on a Windows system, you can use the UNIX conventions.

Part 1. Directory overview

Chapter 1. Defining a directory

A directory is a collection of information about objects arranged in a hierarchical structure. It is a data repository that enables users or applications to find resources that have the characteristics needed for a particular task.

If the name of an object is known, its characteristics can be retrieved. If the name of a particular individual object is not known, the directory can be searched for a list of objects that meet a certain requirement. Directories can usually be searched by specific criteria, not just by a predefined set of characteristics.

A directory is a data repository that has characteristics that set it apart from general purpose relational databases. A characteristic of a directory is that it is accessed (read or searched) much more often than it is updated (written). Because directories must be able to support high volumes of read requests, they are typically optimized for read access. Because directories are not intended to provide as many functions as general-purpose databases, they can be optimized to economically provide more applications with rapid access to directory data in large distributed environments.

A directory can be centralized or distributed. If a directory is centralized, there is one directory server at one location that provides access to the directory. If the directory is distributed, more than one server, sometimes geographically dispersed, provides access to the directory.

When a directory is distributed, the information stored in the directory can be partitioned or replicated. When information is partitioned, each directory server stores a unique and non-overlapping subset of the information. That is, each directory entry is stored by one and only one server. One technique to partition the directory is to use LDAP referrals returned from a server directing clients to refer Lightweight Directory Access Protocol (LDAP) requests to either the same or different name spaces stored in a different (or same) server. Partitioning can also be accomplished with a proxy server without using referrals. When information is replicated, the same directory entry is stored by more than one server. In a distributed directory, some information may be partitioned, and some information may be replicated.

Directory clients and servers

Directories are usually accessed using the client-server model of communication. The directory clients and servers might not be on the same machine. A server is capable of serving many clients. An application that wants to read or write information in a directory does not access the directory directly. Instead, it calls a function or application programming interface (API) that causes a message to be sent to another process. This second process accesses the information in the directory on behalf of the requesting application. The results of the read or write actions are then returned to the requesting application.

An API defines the programming interface that a particular programming language uses to access a service. The format and contents of the messages exchanged between client and server must adhere to an agreed upon protocol. LDAP defines a message protocol used by directory clients and directory servers. There is also an

associated LDAP API for the C language and ways to access the directory from a Java application using the Java Naming and Directory Interface (JNDI).

Directory security

A directory should support the basic capabilities needed to implement a security policy. The directory might not directly provide the underlying security capabilities, but it might be integrated with a trusted network security service that provides the basic security services. First, a method is needed to authenticate users. Authentication verifies that users are who they say they are. A user name and password is a basic authentication scheme. After users are authenticated, it must be determined if they have the authorization or permission to perform the requested operation on the specific object.

Authorization is often based on access control lists (ACLs). An ACL is a list of authorizations that may be attached to objects and attributes in the directory. An ACL identifies what type of access each user or a group of users is allowed or denied on a directory entry or object. In order to make ACLs shorter and more manageable, users with the same access rights are often put into groups or the ACLs can be filtered. See Chapter 17, "Access control lists," on page 309 for more information.

Chapter 2. The IBM Tivoli Directory Server

The IBM Tivoli Directory implements the Internet Engineering Task Force (IETF) LDAP V3 specifications. It also includes enhancements added by IBM in functional and performance areas. This version uses IBM DB2® as the backing store to provide per LDAP operation transaction integrity, high performance operations, and on-line backup and restore capability. The IBM Tivoli Directory Server interoperates with the IETF LDAP V3 based clients. Major features include:

- A dynamically extensible directory schema - This means that administrators can define new attributes and object classes to enhance the directory schema. Changes can be made to the directory schema, too, which are subject to consistency checks. Users may dynamically modify the schema content without restarting the directory server. Because the schema itself is part of the directory, schema update operations are done through standard LDAP APIs. The major functions provided by the LDAPv3 dynamic extensible schema are:
 - Queriable schema information through LDAP APIs
 - Dynamic schema changes through LDAP APIs
 - Server Root DSE
- NLS support – An IBM Tivoli Directory Server supports the UTF-8 (Universal Character Set Transformation Format) character set. This Unicode (or UCS) Transformation Format is an 8-bit encoding form that is designed for ease of use with existing ASCII-based systems. The IBM Tivoli Directory Server also supports data in multiple languages, and allows users to store, retrieve and manage information in a native language code page.
- Replication – Replication is supported, which makes additional copies of the directory available, improving performance and reliability of the directory service. Replication topologies also support forwarding and gateway servers.
- Referrals – Support for LDAP referrals, allowing directories to be distributed across multiple LDAP servers where each single server may contain only a subset of the whole directory data.
- Security features – IBM Tivoli Directory Server provides a rich set of security features.

Identification and authentication

Identification and authentication are used to determine the identity of the LDAP clients; that is, verifying that users are who they say they are. A user name and password is a basic authentication scheme. This user identity is used for determining access rights and for user accountability.

Simple Authentication and Security Layer (SASL)

This support provides for additional authentication mechanisms. For more information, see “Using Web Administration:” on page 113 and “Configuring the DIGEST-MD5 mechanism” on page 150.

The Secure Sockets Layer (SSL) and Transaction Layer Security (TLS)

This support provides encryption of data and authentication using X.509v3 public-key certificates. A server may be configured to run with or without SSL or TLS support or both. For more information, see “Secure Sockets Layer” on page 115 and “Transaction Layer Security” on page 115.

Access control

After users are authenticated, it must be determined whether they have

authorization or permission to perform the requested operation on the specific object. Authorization is often based on access control lists (ACLs). An ACL is a list of authorizations that can be attached to objects and attributes in the directory. An ACL lists what type of access each user or a group of users is allowed or denied. To make ACLs shorter and more manageable, users with the same access rights are often put into groups or the ACLs are filtered. The directory administrator can manage access control by specifying the access rights to objects for individual users or groups. Users can perform operations under alternate access rights by using proxied authorization. For proxied authorization, the user assumes the proxied identity and the ACL restrictions for the proxied identity. For more information, see Chapter 17, "Access control lists," on page 309.

Auditing

The IBM Tivoli Directory Server can perform auditing of security-relevant events, such as user authentication and modification to the directory tree. The audit function provides a means for accountability by generating audit records containing the time, user identity, and additional information about the operation. The directory administrator manages the behavior of the audit function, such as selection of auditable events, as well as audit review and clearing of audit files. For more information, see "Enabling the audit log and modifying audit log settings" on page 274.

Security roles

IBM Tivoli Directory Server supports five different security roles.

Directory administrator

The directory administrator is associated with a specific user account. There is only one directory administrator account for the LDAP server. The directory administrator has full rights to manage the LDAP server. The directory administrator is created during product installation and configuration. The directory administrator consists of a user ID and a password and predefined authorization to manipulate the entire directory. The directory administrator creates the end user security role. This is an LDAP entry with a specific distinguished name (DN), user password, and other attributes that represent the particular end user. The directory administrator also defines the level of authorization the end user will have over entries.

Administrative group members

Administrative group members are users that have been assigned a subset of administrative privileges. All administrative group members have the same set of privileges. The administrative group is a way for the directory administrator to delegate a limited set of administrative tasks to one or more individual user accounts. These users can perform most administrative tasks. Exceptions are operations that might increase the privileges of those users, such as change the password of the directory administrator or clearing audit log files. For more information, see "Creating the administrative group" on page 86.

Global administrative group members

The global administrative group is a way for the directory administrator to delegate administrative rights in a distributed

environment to the database backend. Global administrative group members are users that have been assigned the same set of privileges as the administrative group with regard to accessing entries in the database backend. Global administrative group members have complete access to the directory server backend. Global administrative group members do not have access to the audit log and thus the audit log can be used by local administrators to monitor global administrative group member activity.

The global administrative group members have no privileges or access rights to any data or operations that are related to the configuration settings of the directory server. This is commonly called the configuration backend. All global administrative group members have the same set of privileges.

LDAP user

LDAP users are users whose privileges are determined by ACLs. Each LDAP user is identified with an LDAP entry containing the authentication and authorization information for that end user. The authentication and authorization information might also allow the end user to query and update other entries. Depending on the type of authentication mechanism used, after the end user ID and password are validated, the end user can access any of the attributes of any entry to which that end user has permissions.

Master server DN

The master server DN is a role used by replication that can update the entries under a replica's or a forwarding replica's replication context to which the DN is defined as a master server DN. The master server DN can create a replication context entry on a replica or forwarding replica if the DN is defined as the master server DN to that specific replication context or as a general master server DN.

By sending a AES bind control, a master server DN can send AES encrypted data to a replica.

The following are some important points about the master server DN:

- There can be several master server DNs defined in a server's configuration file. There is an `ibm-slapdReplication` object that can contain a default or general `ibm-slapdMasterDN`, and there can be multiple `ibm-slapdSupplier` objects, each defining an `ibm-slapdMasterDN` for a specific replication context (that is, limited to a specific subtree). The administration password policy applies to them all.
- Any of those master server DNs can bind to the directory.
- Any of those master server DNs have access to update the `ibm-slapdSuffix` attribute of the entry
`cn=Directory, cn=RDBM Backends, cn=IBM Directory, cn=schemas, cn=Configuration`

in a server's configuration file. A master server DN does not have read or write access to any other entries in the configuration file.

- No master server DN has access to any other part of the configuration file.
- Only the general master server DN or the master server DN for the cn=IBMpolicies context can make updates to the schema.
- The master server DN for a specific context has full read and write access to all entries within that context.
- The general master server DN has full read and write access to all entries within all contexts.

Password policy

The password policy feature provided by the IBM Tivoli Directory Server allows the administrator to define the policy used for administrator and user passwords. The administrator places restrictions on passwords by specifying rules for syntax, validation, and lockout in the password policy. The administrator password policy configuration is stored in the configuration backend and can be modified only by the root administrator. The user password policy configuration is stored within the LDAP tree and can be modified by the root administrator or a member of the administrative group. The attribute values can be changed only when binding as administrator to the IBM Tivoli Directory Server. For more information, see “Setting password policy” on page 134.

Password encryption

IBM Directory enables you to prevent unauthorized access to user passwords.

The administrator can configure the server to encrypt userPassword attribute values in either a one-way encrypting format or a two-way encrypting format.

One-way encrypting formats:

- SHA-1
- crypt

After the server is configured, any new passwords (for new users) or modified passwords (for existing users) are encrypted before they are stored in the directory database.

For applications that require retrieval of clear passwords, such as middle-tier authentication agents, the directory administrator needs to configure the server to perform either a two-way encrypting or no encryption on user passwords.

Two-way encrypting format:

- AES

When you configure the server using Web Administration, you can select one of the following encryption options:

- None** No encryption. Passwords are stored in the clear text format.
- crypt** Passwords are encrypted by the UNIX crypt encrypting algorithm before they are stored in the directory.

SHA-1

Passwords are encrypted by the SHA-1 encrypting algorithm before they are stored in the directory.

AES128

Passwords are encrypted by the AES128 algorithm before they are stored in the directory and are retrieved as part of an entry in the original clear format.

AES192

Passwords are encrypted by the AES192 algorithm before they are stored in the directory and are retrieved as part of an entry in the original clear format.

AES256

Passwords are encrypted by the AES256 algorithm before they are stored in the directory and are retrieved as part of an entry in the original clear format.

The default option is AES256. A change is registered in a password encryption directive of the server configuration file:

```
ibm-SlapdPwEncryption: AES256
```

The server configuration file is located in:

```
<instance_directory>\etc\ibmslapd.conf
```

Notes:

1. If the UNIX crypt method is used, only the first 8 characters are effective.
 2. A one-way encrypted password can be used for password matching but it cannot be decrypted. During user login, the login password is encrypted and compared with the stored version for matching verification.
- Change log – Records changes made to the LDAP data and are logged in a separate database in the LDAP server to support meta-directories or client queries to monitor directory updates.
 - Dynamic configuration – Changes using LDAP APIs 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. It supports the standard host, port, SSL, 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.
 - Web Administration Tool – A Graphical User Interface (GUI) that can be used to administer and configure the IBM Directory. The administration and configuration functions enable the administrator to:
 - Perform the initial setup of the directory
 - Change configuration parameters and options
 - Manage the daily operations of the directory, such as adding or editing objects, for example, object classes, attributes, and entries.
 - Proxy server – Special type of IBM Tivoli Directory Server that provides proxy routing, distributed authentication, load balancing, fail over and support for enhanced groups and partitioning of containers.

- Administration daemon (idsdiradm) – Enables remote management of an instance of the IBM Tivoli Directory Server. It must be installed on the machine where the IBM Tivoli Directory Server is installed and must be running continuously.
- Configuration only mode – Gives an administrator remote access to the server even when errors are encountered during startup. The server does not depend on the successful initialization of the database back end. An administrator can use an LDAP protocol to query and update the configuration for the server.
- Attribute uniqueness controls – Can be configured to ensure that specified attributes always have unique values within a directory on a single directory server.
- Language tags – Enables the directory to associate natural language codes with values held in a directory and enables clients to query the directory for values that meet certain natural language requirements.
- Sorting on searches – Sorts the entries found by the search using the first 240 bytes of the specified attribute values.
- Paged results – Provides paging capabilities for LDAP clients that want to receive just a subset of search results (a page) instead of the entire list.
- Transactions – Enable an application to group a set of entry updates together in one transaction.
- Event notification – Enables a server to notify a registered client that an entry in the directory tree has been changed, added, or deleted.
- Multiple instances – Enables a user to have more than one directory instance on a server.

Chapter 3. Distinguished names (DNs)

Every entry in the directory has a distinguished name (DN). The DN is the name that uniquely identifies an entry in the directory. A DN is made up of attribute=value pairs, separated by commas, for example:

```
cn=Ben Gray,ou=editing,o=New York Times,c=US
cn=Lucille White,ou=editing,o=New York Times,c=US
cn=Tom Brown,ou=reporting,o=New York Times,c=US
```

Any of the attributes defined in the directory schema, other than system or restricted attributes, may be used to make up a DN. The order of the component attribute value pairs is important. The DN contains one component for each level of the directory hierarchy from the root down to the level where the entry resides. LDAP DNs begin with the most specific attribute (usually some sort of name), and continue with progressively broader attributes, often ending with a country attribute. The first component of the DN is referred to as the Relative Distinguished Name (RDN). It identifies an entry distinctly from any other entries that have the same parent. In the examples above, the RDN "cn=Ben Gray" separates the first entry from the second entry, (with RDN "cn=Lucille White"). These two example DNs are otherwise equivalent. The attribute:value pair making up the RDN for an entry must also be present in the entry. (This is not true of the other components of the DN.)

Distinguished name syntax

The Distinguished Name (DN) syntax supported by this server is based on RFC 2253. The Backus-Naur Form (BNF) syntax is defined as follows:

```
<name> ::= <name-component> ( <spaced-separator> )
          | <name-component> <spaced-separator> <name>

<spaced-separator> ::= <optional-space>
                    <separator>
                    <optional-space>

<separator> ::= ", " | ";"

<optional-space> ::= ( <CR> ) *( " " )

<name-component> ::= <attribute>
                    | <attribute> <optional-space> "+"
                    <optional-space> <name-component>

<attribute> ::= <string>
              | <key> <optional-space> "=" <optional-space> <string>

<key> ::= 1*( <keychar> ) | "OID." <oid> | "oid." <oid>
<keychar> ::= letters, numbers, and space

<oid> ::= <digitstring> | <digitstring> "." <oid>
<digitstring> ::= 1*<digit>
<digit> ::= digits 0-9

<string> ::= *( <stringchar> | <pair> )
           | "'" *( <stringchar> | <special> | <pair> ) "'"
           | "#" <hex>

<special> ::= ", " | "=" | <CR> | "+" | "<" | ">"
```

```

| "#" | ";"

<pair> ::= "\" ( <special> | "\" | "'" )
<stringchar> ::= any character except <special> or "\" or "'"

<hex> ::= 2*<hexchar>
<hexchar> ::= 0-9, a-f, A-F

```

A semicolon (;) character can be used to separate RDNs in a distinguished name, although the comma (,) character is the typical notation.

White-space characters (spaces) might be present on either side of the comma or semicolon. The white-space characters are ignored, and the semicolon is replaced with a comma.

In addition, space (' ' ASCII 32) characters may be present either before or after a '+' or '='. These space characters are ignored when parsing.

A value may be surrounded by double quotation ("" ACSII 34) characters, which are not part of the value. Inside the quoted value, the following characters can occur without being interpreted as escape characters:

- A space or "#" character occurring at the beginning of the string
- A space character occurring at the end of the string
- One of the characters "", "=", "+", "\", "<", ">", or ";"

Alternatively, a single character to be escaped may be prefixed by a backslash ('\ ASCII 92). This method can be used to escape any of the characters listed previously and the double quotation marks ("" ASCII 34) character.

This notation is designed to be convenient for common forms of names. The following example is a distinguished name written using this notation. First is a name containing three components. The first of the components is a multivalued RDN. A multivalued RDN contains more than one attribute:value pair and can be used to distinctly identify a specific entry in cases where a simple CN value might be ambiguous:

```
OU=Sales+CN=J. Smith,O=Widget Inc.,C=US
```

DN escaping rules

A DN can contain special characters. These characters are , (comma), = (equals), + (plus), < (less than), > (greater than), # (number sign), ; (semicolon), \ (backslash), and "" (quotation marks).

To escape these special characters or other characters in an attribute value in a DN string, use any the following methods:

- If a character to be escaped is one of special characters, precede it by a backslash ('\ ASCII 92). This example shows a method of escaping a comma in an organization name:

```
CN=L. Eagle,O=Sue\, Grabbit and Runn,C=GB
```

This is the preferred method.

- Otherwise replace the character to be escaped by a backslash and two hex digits, which form a single byte in the code of the character. The code of the character **must** be in UTF-8 code set.

CN=L. Eagle,o=Sue\2C Grabbit and Runn,C=GB

- Surround the entire attribute value by "" (quotation marks) (ASCII 34) that are not part of the value. Between the quotation character pair, all characters are taken as is, except for the \ (backslash). The \ (backslash) can be used to escape a backslash (ASCII 92) or quotation marks (ASCII 34), any of the special characters previously mentioned, or hex pairs as in method 2. For example, to escape the quotation marks in cn=xyz"qrs"abc, it becomes cn=xyz\"qrs\"abc or to escape a \:

"you need to escape a single backslash this way \\"

Another example, "\Zoo" is illegal, because 'Z' cannot be escaped in this context.

On the server end, when a DN is received in this form, the server reformats the DN using escape mechanisms number 1 and 2 for internal processing.

Enhanced DN processing

A composite RDN of a DN may consist of multiple components connected by the '+' operators. The server enhances the support for searches on entries that have such a DN. A composite RDN can be specified in any order as the base for a search operation.

idsldapsearch cn=mike+ou=austin,o=ibm,c=us

The server accepts DN normalization extended operations. DN normalization extended operations normalize DNs using the server schema. This extended operation might be useful for applications that use DNs. See the *IBM Tivoli Directory Server Version 6.0 C-client Programming Reference* for more information.

Part 2. Server Administration

Chapter 4. Directory administration daemon

The directory administration daemon (idsdiradm) enables remote management of an instance of the IBM Tivoli Directory Server. It must be installed on the machine where the IBM Tivoli Directory Server is installed and must be running continuously. The directory administration daemon accepts requests by way of LDAP extended operations and supports starting, stopping, restarting, and status monitoring of the IBM Tivoli Directory Server. By default, the first instance of the IBM Directory administration daemon listens on two ports, port 3538 for non-SSL connections and port 3539 for SSL connections, if SSL communication is enabled.

The directory administration daemon can also be used to perform root DSE searches.

To start the directory administration daemon, run the program idsdiradm from any command prompt. See “Starting an instance of the directory administration daemon.”

Notes:

1. The administration daemon supports auditing version 3 only.
2. If you enable SSL communication, the directory administration daemon must be stopped and restarted for SSL to take effect. See “Using Web Administration:” on page 113.
3. If you change the time zone on your Windows machine, you need to restart the server and the administration daemon in order for the server and administration daemon to recognize the time change. This ensures that the time stamps in the administration daemon’s logs match the time stamps in the server’s logs.

Starting an instance of the directory administration daemon

Note: By default, the administration daemon is started when you create a directory server instance.

To start an instance of the administration daemon do either of the following:

- For UNIX or Linux-based and Windows-based systems issue the command:
`idsdiradm -I <instancename>`
- For Windows-based systems, you can also use **Control Panel ->Administrative Tools->Services**, select **IBM Tivoli Directory Server Instance V6.0 - <instancename> Admin Daemon**, click **Start**.

Note: On Linux SLES systems, the Admin Daemon must not be started from inittab. Instead, start the Admin Daemon manually from the command line. See “idsdiradm, ibmdiradm” on page 438 for more information.

Stopping an instance of the directory administration daemon

To stop an instance of the administration daemon use one of the following methods:

- If you have already configured a directory administration DN and password, you can use the **ibmdirctl** command to stop the administration daemon. This command is not platform specific. See “idsdirctl, ibmdirctl” on page 369 for additional information.

Issue one of the commands:

```
ibmdirctl -D <adminDN> -w <adminPW> -h <hostname>  
          -p <port> admstop
```

The **ibmdirctl** command can be issued locally or remotely.

```
idsdiradm -I <instancename> -k
```

The **idsdiradm** command must be issued locally.

- For Windows-based systems, you can also use **IBM Tivoli Directory Server Instance V6.0 - <instancename> Admin Daemon**, click **Stop**.

Chapter 5. Configuration only mode

The IBM Tivoli Directory Server supports LDAP access to the server's configuration settings. An administrator can use LDAP protocol to query and update the configuration for the server. This feature enables remote administration. In order for this access to be more robust and reliable, the server does not depend on successful initialization of the database back ends. It is possible to start the server in configuration only mode with only the `cn=configuration` suffix active. In other words, as long as the configuration backend is available, the server starts and accepts LDAP requests. Configuration only mode gives an administrator remote access to the server even when errors are encountered during startup.

The following features are supported in configuration only mode:

- Access to the configuration file and log files.
- Auditing
- Event notification
- Kerberos
- SASL
- SSL

The following features are not supported in configuration only mode:

- Access to the database
- Changelog
- Password policy
- Replication
- Schema changes
- Transactions

Minimum requirements for configuration only mode

- The configuration file must be in the correct LDIF format and the server must be able to locate and read the file.
- The server must be able to read and load the schema according to the configuration file.
- The server must be able to load the configuration plug-in.

How to start in configuration only mode

Any failure during server startup causes the server to start in configuration only mode.

Using Web Administration:

Check the **Configuration only mode** when starting the server through the Web Administration Tool.

Using the command line:

Specify `-a` or `-A` on server startup.
`idsslapd -a -I <instancename>`

or

```
ibmdirctl -h <hostname> -D <adminDN> -w <adminpw> -p <portnumber>  
start -- -a
```

Note: The **-n** and **-N** options prevent the server from starting, if the server is unable to start with the database backends (not in configuration only mode). See “*idsdirctl, ibmdirctl*” on page 369 for more information about these *idsslapd* options.

How to verify that the server is running in configuration only mode

To determine if the server is running in configuration only mode, use one of the following methods.

Using Web Administration:

If the server has started in configuration only mode the  icon, located between the stop and start icons, is highlighted.

Using the command line:

Issue a search of the root DSE for the attribute **ibm-slapdisconfigurationmode**. If set to true, the server is running in configuration only mode.

```
idsldapsearch -s base -b " " objectclass=* ibm-slapdisconfigurationmode
```

Chapter 6. Web Administration Tool graphical user interface (GUI)

The IBM Tivoli Directory Server Version 6.0 Web Administration Tool is installed on an application server, such as the embedded version of IBM WebSphere® Application Server - Express (WAS) included with the IBM Tivoli Directory Server, and administered through a console. Servers that have been added to the console can be managed through the Web Administration Tool without having to have the tool installed on each server.

The preferred method of administering the server is by using the Web Administration Tool.

Before you can start using the Web Administration Tool for the server, you want to ensure that you have completed the following tasks during the configuration of that server:

- You must have set the administration DN and password to be able to start a given server.
- If the server is not configured as a proxy server, you must have configured a database to be able to start a given server in a state other than configuration only mode.
- To log on to a given server, either the server or the administration daemon must be running.
- You must have the administration daemon running to be able to start, stop, or restart a given server remotely or through the Web Administration Tool.

See the *IBM Tivoli Directory Server Version 6.0 Installation and Configuration Guide* and Chapter 4, "Directory administration daemon," on page 17 for information on these tasks.

Note: If you have other application servers running, ensure that the application server where the Web Administration Tool is installed is not running on the same port as the other application servers.

Starting the Web Administration Tool

To start the Web Administration Tool, you must start the application server in which it was installed.

For the embedded version of IBM WebSphere Application Server - Express go to the directory where you installed the IBM Tivoli Directory Server and issue the command:

For UNIX or Linux-based platforms

```
<WASinstalldir>/appsrv/bin/startServer.sh server1
```

If you used ISMP to install, the default location is

```
<IDSinstalldir>/appsrv/bin/startServer.sh server1
```

For Windows-based platforms

```
<WASinstalldir>\appsrv\bin\startServer.bat server1
```

If you used ISMP to install, the default location is

```
<IDSinstalldir>\appsrv\bin\startServer.bat server1
```

Logging in to the console

Open a Web browser and type the following address:
http://localhost:12100/IDSWebApp/IDSjsp/Login.jsp or if you are using secure communications with your Web browser
https://localhost:12101/IDSWebApp/IDSjsp/Login.jsp. The IBM Tivoli Directory Server Web Administration login page panel is displayed.

Note: localhost is a host name or an IP address, if you are logged on to a browser that is not on the same machine where the Web Administration Tool is installed.

Logging on to the console as the console administrator

To log on as the console administrator:

1. At the IBM Tivoli Directory Server Web Administration login page log in as **Console Admin**, the default selection in the **LDAP Hostname** field.
2. In the **Username** field type: superadmin, if the default user name has not been changed. See “Managing the console” on page 27.
3. In the **Password** field type: secret, if the default password has not been changed. See “Managing the console” on page 27.
4. Click **Login**.

The IBM Tivoli Directory Server Web Administration Tool console is displayed.

Logging on to the console as the server administrator, a member of an administrative group or an LDAP user

To log on as the server administrator, a member of the administrative group (see “Creating the administrative group” on page 86) or an LDAP user:

- At the IBM Tivoli Directory Server Web Administration login page select the LDAP host name or IP address and port for your machine from the drop-down menu.
- Enter the administrator DN and the password for that server (you set these up during the server configuration process). For example, if the administrator DN which was created during the server configuration process was cn=root, then enter the full administrator DN. Do not just use root.
- Click **Login**.

The IBM Tivoli Directory Server Web Administration Tool console is displayed with various server management tasks. The server management tasks vary depending upon the capabilities of the server and the type of user that you have logged on as.

Note: The Web Administration Tool does not support logging on to a given server using replication supplier credentials.

Console layout

The **IBM Tivoli Directory Server Web Administration Tool** console consists of five areas:

Banner area

The banner area located at the top of the panel contains the application name, IBM Tivoli Directory Server Web Administration Tool, and the IBM Logo.

Navigation area

The navigation area, located on the left side of the panel, displays expandable categories for various console or server tasks. The tasks available vary depending on your authority or the capabilities of the server you are logging onto or both.

Work area

The work area displays the tasks associated with the selected task in the navigation area. For example, if **Managing server security** is selected in the navigation area, the work area displays the Server Security page and the tabs containing tasks related to setting up server security.

Server status area

Note: If you are logged on as the console administrator, this area displays "Console administrator" and provides an icon link to the table of contents for task helps.

The server status area, located at the top of the work area, indicates the status and the name of the server being administered. It also has two icon links, one to the Start/Stop/Restart procedure and the other to general help information. When you select a task from the navigation area, the name of the selected task, a link to the error log files, and a link to the task help are also displayed.

Task status area

The task status area, located beneath the work area, displays the status of the current task.

Logging off the console

To log off of the console, click **Logout** in the navigation area.

The Logout successful panel displays the message:

You have successfully been logged off the server. This action has occurred because you hit the logout button. Please note that this browser window and any other browser windows opened while you were working on the server have now expired. No further interaction can occur with the server by clicking in these windows.

You can re-login by clicking here.










Click the word **here** in this message to return to the IBM Tivoli Directory Server Web Administration Login Page.

Using tables in the Web Administration Tool

The IBM Tivoli Directory Server Web Administration Tool displays certain information, such as lists of attributes and entries, in tables. Tables contain several utilities that enable you to search for, organize and perform actions on these table items.

Table icons

IBM Tivoli Directory Server Web Administration Tool tables provide icons to help you organize and find information in the table. Some icons are displayed on some tables and not on others, depending on the current task. The following is a comprehensive list of the icons you might encounter:

-  Click the **Show Filter Row** icon to display filter rows for every column in the table. See “Filtering” on page 26 for more information about filtering.
-  Click the **Hide Filter Row** icon to display filter rows for every column in the table. See “Filtering” on page 26 for more information about filtering.
-  Click the **Clear all filters** icon clear all filters set for the table. See “Filtering” on page 26 for more information about filtering .
-  Click the **Edit sort** icon to sort the information in the table. See “Sorting” on page 25 for more information about sorting.
-  Click the **Clear all sorts** icon clear all sorts set for the table. See “Sorting” on page 25 for more information about sorting.
-  Click the **Collapse table** icon to hide the table data.
-  Click the **Expand table** icon to display the table data.
-  Click the **Select all** icon to select all items in the table.
-  Click the **Deselect all** icon to deselect all selected items in the table.

Select Action drop-down menu

The **Select Action** drop-down menu contains a comprehensive list of all available actions for the selected table.

For example, instead of using the icons to display and hide sorts and filters, you can use the **Select Action** drop-down menu. You can also use the **Select Action** drop-down menu to perform operations on the table contents; for example, on the Manage attributes panel, actions such as **View**, **Add**, **Edit**, **Copy** and **Delete** are displayed not only as buttons on the toolbar, but also in the **Select Action** drop-down menu. If the table supports it, you can also display or hide the **Show find toolbar** using the **Select Action** drop-down menu. See “Finding” on page 25 for more information on finding table items.

To perform an action using the **Select Action** menu:

1. Click the **Select Action** drop-down menu.
2. Select the action you want to perform; for example **Edit sort**.
3. Click **Go**.

Paging

To view different table pages, use the navigation controls at the bottom of the table. You can enter a specific page number into the navigation field and click **Go** to display a certain page. You can also use the **Next** and **Previous** arrows to move from page to page.

Sorting

To change the way items in a table are sorted:

1. Do one of the following:
 - Click the **Edit sort** icon on the table.
 - Click the **Select Action** drop-down menu, select **Edit sort** and click **Go**. A sorting drop-down menu is displayed for every column in the table.
2. From the first sort drop-down menu, select the column that you want to sort. Do the same for any of the other sortable columns that you want to sort.
3. Select whether to sort in ascending or descending order by selecting **Ascending** or **Descending** from the drop-down menu. Ascending is the default sort order. You can also sort using column headers. On every column is a small arrow. An arrow pointing up means that column is sorted in ascending order. An arrow pointing down means that column is sorted in descending order. To change the sort order, simply click on the column header.
4. When you are ready to sort, click **Sort**.

To clear all the sorts, click the **Clear all sorts** icon.

Finding

To find a specific item or items in a table:

Note: The **Show find toolbar** option is available on some tables and not on others, depending on the current task.

1. Select **Show find toolbar** from the **Select Action** drop-down menu and click **Go**.
2. Enter your search criteria in the **Search for** field.
3. If desired, select a condition upon which to search from the **Conditions** drop-down menu. The options for this menu are:
 - **Contains**
 - **Starts with**
 - **Ends with**
 - **Exact match**
4. Select the column upon which you want to base the search from the **Column** drop-down menu.
5. Select whether to display results in descending or ascending order from the **Direction** drop-down menu. Select **Down** to display results in descending order. Select **Up** to display results in ascending order.
6. Select the **Match case** check box, if you want search results to match the upper and lower case criteria in the **Search for** field.
7. When you have entered the desired criteria, click **Find** to search for the attributes.

Filtering

To filter items in a table, do the following:

1. Do one of the following:
 - Click the **Show filter** icon.
 - Click the **Select Action** drop-down menu, select **Show filter row** and click **Go**.

Filter buttons are displayed above each column.
2. Click **Filter** above the column you want to filter.
3. Select one of the following conditions from the **Conditions** drop-down menu:
 - **Contains**
 - **Starts with**
 - **Ends with**
4. Enter the text you want to filter on in the field; for example, if you selected **Starts with**, you might enter C.
5. If you want to match case (upper case text or lower case text) select the Match case check box.
6. When you are ready to filter the attributes, click **OK**.
7. Repeat step 2 through step 6 for every column you want to filter.

To clear all the filters, click the **Clear all filters** icon.

To hide the filter rows, click the **Show filter** icon again.

Chapter 7. Setting up the Web Administration Tool

After you have started the application server, you need to set up the console that is going to manage your directory servers. From the IBM Tivoli Directory Server Web Administration login page, log in as the console administrator and perform the following tasks:

Managing the console

At the IBM Tivoli Directory Server Web Administration Tool console:

Changing the console administrator login

To change the console administrator ID:

1. Expand **Console administration** in the navigation area.
2. Click **Change console administrator login**.
3. Enter the new administrator ID.

Note: Only one console administrator ID is allowed. The administrator ID is replaced by the new ID that you specified. When the Web Administration Tool is initially deployed the default console administrator value is **superadmin**.

4. Enter the current administrator password. The password, secret, is the same for the new administrator ID, until you change it.

Changing the console administration password

For security reasons, change the default console administrator password, secret, to another password.

Note: Because the password policy cannot be enforced for the password of the console administrator, the administrator must implement organizational means to ensure that the configuration shown for the password policy is also enforced for the password of the console administrator.

To change the console administrator password:

1. Expand **Console administration** in the navigation area.
2. Click **Change console administrator password**.
3. Enter the current password.
4. Enter the new password.
5. Enter the new password again to confirm that there are no typographical errors.
6. Click **OK**.

Adding, modifying, and removing servers in the console

Use the following procedures to add, edit, or delete servers in the console:

Adding a server to the console

To add a server to the console:

1. Expand **Console administration** in the navigation area.

2. Click **Manage console servers**. A table for listing of server host names and port numbers is displayed.
3. Click **Add**.
4. Enter the host name address or the IP address of the server. For example *servername.austin.ibm.com*
5. Specify the port numbers or accept the defaults.

Note: For multiple server instances on the same machine, although the host name remains the same, you must specify the correct port that was assigned to the directory server instance.

6. Specify if the server is SSL enabled. Ensure that you complete step 5 on page 29 under **Managing console properties**.
7. Click **OK** to apply the changes or click **Cancel** to exit the panel without making any changes.

Modifying a server in the console

To change the port number or SSL enablement of a server:

1. Expand **Console administration** in the navigation area.
2. Click **Manage console servers**. A listing of server host names and port numbers is displayed.
3. Select the radio button next to the server you want to modify.
4. Click **Edit**.
5. You can change the port numbers.
6. You can change whether the server is SSL enabled. Ensure that you complete step 5 on page 29 under **Managing console properties**, if you are enabling SSL.
7. Click **OK** to apply the changes or click **Cancel** to exit the panel without making any changes.

Removing a server from the console

To remove a server from the console:

1. Expand **Console administration** in the navigation area.
2. Click **Manage console servers**. A listing of server host names and port numbers is displayed.
3. Select the radio button next to the server you want to remove.
4. Click **Delete**.
5. A message to confirm that you want to remove the server is displayed. Click **OK** to remove the server or click **Cancel** to exit the panel without removing the server.

Managing console properties

To change the settings for the console properties:

1. Expand **Console administration** in the navigation area.
2. Click **Manage console properties**.
3. Click **Component management** - to specify the components that are enabled for all servers in the console. By default all the components are enabled.

Note: You might not see a management component or some of its tasks, even if it is enabled, if you do not have the correct authority on the server or the server does not have the needed capabilities, or both.

4. Click **Session properties** - to set the time out limit for the console session. The default setting is 60 minutes.

Note: A session might be valid for three to five minutes more than what you have set. This is because the invalidations are performed by a background thread in the application server that acts on a timer interval. This timer interval extends the session time out duration.

5. Click **SSL key database** - to set up the console so that it can communicate with other LDAP servers using the Secure Sockets Layer (SSL), if necessary. Set the key database path and file name, the key password, the trusted database path and file name, the trusted password in the appropriate fields. The supported file type is jks. See "Using gsk7ikm" on page 121 and "Secure Sockets Layer" on page 115 for information about key databases and SSL.

When you have finished setting up the console, click **Logout** to exit. See "Logging off the console" on page 23 for more information.

Chapter 8. Managing the IBM Directory schema

A schema is a set of rules that governs the way that data can be stored in the directory. The schema defines the type of entries allowed, their attribute structure, and the syntax of the attributes.

Note: The schema information shipped with the server, such as object class descriptions and syntax, is in English. It is not translated.

Data is stored in the directory using directory entries. A entry consists of an object class, which is required, and its attributes. Attributes can be either required or optional. The object class specifies the kind of information that the entry describes and defines the set of attributes it contains. Each attribute has one or more associated values. See Chapter 16, "Working with directory entries," on page 293 for additional information about entries.

The schema for the IBM Directory Version 6.0 is predefined, however, you can modify the schema, if you have additional requirements.

The IBM Tivoli Directory Server Version 6.0 includes dynamic schema support. The schema is published as part of the directory information, and is available in the Subschema entry (DN="cn=schema"). You can query the schema using the `ldap_search()` API and modify it using `ldap_modify()`. See the *IBM Directory Client SDK Programming Reference* for more information about these APIs.

The schema has more configuration information than that included in the LDAP Version 3 Request For Comments (RFCs) or standard specifications. For example, for a given attribute, you can state which indexes must be maintained. This additional configuration information is maintained in the subschema entry as appropriate. An additional object class is defined for the subschema entry `IBMsubschema`, which has "MAY" attributes that hold the extended schema information.

IBM Tivoli Directory Server requires that the schema defined for a naming context be stored in a special directory entry, "cn=schema". The entry contains all of the schema defined for the server. To retrieve schema information, you can perform an `ldap_search` by using the following:

```
DN: "cn=schema", search scope: base, filter: objectclass=subschema
or objectclass=*
```

The schema provides values for the following attribute types:

- `objectClasses` (See "Working with object classes" on page 33.)
- `attributeTypes` (See "Working with attributes" on page 39.)
- `IBMAttributeTypes` (See "The `IBMAttributeTypes` attribute type" on page 40.)
- matching rules (See "Matching rules" on page 41).
- `ldap syntaxes` (See "Attribute syntax" on page 50).

The syntax of these schema definitions is based on the LDAP Version 3 RFCs.

A sample schema entry might contain:

```

objectclasses=( 1.3.6.1.4.1.1466.101.120.111
                NAME 'extensibleObject'
                SUP top AUXILIARY )

objectclasses=( 2.5.20.1
                NAME 'subschema'
                AUXILIARY MAY
                ( dITStructureRules
                  $ nameForms
                  $ ditContentRules
                  $ objectClasses
                  $ attributeTypes
                  $ matchingRules
                  $ matchingRuleUse ) )

objectclasses=( 2.5.6.1
                NAME 'alias'
                SUP top STRUCTURAL
                MUST aliasedObjectName )

attributeTypes {
  ( 2.5.18.10 NAME 'subschemaSubentry' EQUALITY distinguishedNameMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 NO-USER-MODIFICATION
    SINGLE-VALUE USAGE directoryOperation )
  ( 2.5.21.5 NAME 'attributeTypes'
    EQUALITY objectIdentifierFirstComponentMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.3 USAGE directoryOperation )
  ( 2.5.21.6 NAME 'objectClasses'
    EQUALITY objectIdentifierFirstComponentMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.37 USAGE directoryOperation )
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 USAGE directoryOperation )
}

ldapSyntaxes {
  ( 1.3.6.1.4.1.1466.115.121.1.5 DESC 'Binary' )
  ( 1.3.6.1.4.1.1466.115.121.1.7 DESC 'Boolean' )
  ( 1.3.6.1.4.1.1466.115.121.1.12 DESC 'DN' )
  ( 1.3.6.1.4.1.1466.115.121.1.15 DESC 'Directory String' )
  ( 1.3.6.1.4.1.1466.115.121.1.24 DESC 'Generalized Time' )
  ( 1.3.6.1.4.1.1466.115.121.1.26 DESC 'IA5 String' )
  ( 1.3.6.1.4.1.1466.115.121.1.27 DESC 'INTEGER' )
  ( 1.3.6.1.4.1.1466.115.121.1.50 DESC 'Telephone Number' )
  ( 1.3.6.1.4.1.1466.115.121.1.53 DESC 'UTC Time' )
}

matchingRules {
  ( 2.5.13.2 NAME 'caseIgnoreMatch'
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
  ( 2.5.13.0 NAME 'objectIdentifierMatch'
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.38 )
  ( 2.5.13.30 NAME 'objectIdentifierFirstComponentMatch'
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.38 )
  ( 2.5.13.4 NAME 'caseIgnoreSubstringsMatch'
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.58 )
}

```

As shown in the preceding example, it is not required that all of the attribute values of a given attribute type be provided in a single production.

The schema information can be modified through the `ldap_modify` API. Consult the *Client SDK Programming Reference* for additional information. With the DN "cn=schema" you can add, delete, or replace an attribute type or an object class. To delete a schema entity, provide the oid in parenthesis (oid). You also can provide a full description. You can add or replace a schema entry with the LDAP Version 3 definition or with the IBM attribute extension definition or with both definitions.

Common schema support

The IBM Directory supports standard directory schema as defined in the following:

- The Internet Engineering Task Force (IETF) LDAP Version 3 RFCs, such as RFC 2252 and 2256.
- The Directory Enabled Network (DEN)
- The Common Information Model (CIM) from the Desktop Management Task Force (DMTF)
- The Lightweight Internet Person Schema (LIPS) from the Network Application Consortium

This version of LDAP includes the LDAP Version 3 defined schema in the default schema configuration. It also includes the DEN schema definitions.

IBM also provides a set of extended common schema definitions that other IBM products share when they exploit the LDAP directory. They include:

- Objects for white page applications such as `eperson`, `group`, `country`, `organization`, `organization unit` and `role`, `locality`, `state`, and so forth
- Objects for other subsystems such as `accounts`, `services` and `access points`, `authorization`, `authentication`, `security policy`, and so forth.

Object identifier (OID)

An object identifier (OID) is a string, of decimal numbers, that uniquely identifies an object. These objects are typically an object class or an attribute. These numbers can be obtained from the IANA (Internet Assigned Number Authority). The IANA Website is located at: <http://www.iana.org/iana/>.

If you do not have an OID, you can specify the object class or attribute name appended with `-oid`. For example, if you create the attribute `tempID`, you can specify the OID as `tempID-oid`.

Working with object classes

An object class specifies a set of attributes used to describe an object. For example, if you created the object class `tempEmployee`, it could contain attributes associated with a temporary employee such as, `idNumber`, `dateOfHire`, or `assignmentLength`. You can add custom object classes to suit the needs of your organization. The IBM Tivoli Directory Server schema provides some basic types of object classes, including:

- Groups
- Locations
- Organizations
- People

Note: Object classes that are specific to the IBM Tivoli Directory Server have the prefix `'ibm-'`.

Defining object classes

Object classes are defined by the characteristics of type, inheritance, and attributes.

Object class type

An object class can be one of three types:

Structural:

Every entry must belong to one and only one structural object class, which defines the base contents of the entry. This object class represents a real world object. Because all entries must belong to a structural object class, this is the most common type of object class.

Abstract:

This type is used as a superclass or template for other (structural) object classes. It defines a set of attributes that are common to a set of structural object classes. These object classes, if defined as subclasses of the abstract class, inherit the defined attributes. The attributes do not need to be defined for each of the subordinate object classes.

Auxiliary:

This type indicates additional attributes that can be associated with an entry belonging to a particular structural object class. Although an entry, can belong to only a single structural object class, it may belong to multiple auxiliary object classes.

Object Class Inheritance

This version of the IBM Tivoli Directory Server supports object inheritance for object class and attribute definitions. A new object class can be defined with parent classes (multiple inheritance) and the additional or changed attributes.

Each entry is assigned to a single structural object class. All object classes inherit from the abstract object class **top**. They can also inherit from other object classes. The object class structure determines the list of required and allowed attributes for a particular entry. Object class inheritance depends on the sequence of object class definitions. An object class can only inherit from object classes that precede it. For example, the object class structure for a person entry might be defined in the LDIF file as:

```
objectClass: top
objectClass: person
objectClass: organizationalPerson
```

In this structure, the `organizationalPerson` inherits from the `person` and the `top` object classes, while `person` object class only inherits from the `top` object class. Therefore, when you assign the `organizationalPerson` object class to an entry, it automatically inherits the required and allowed attributes from the superior object class. In this case, the `person` object class.

Schema update operations are checked against the schema class hierarchy for consistency before being processed and committed.

Attributes

Every object class includes a number of required attributes and optional attributes. Required attributes are the attributes that must be present in entries using the object class. Optional attributes are the attributes that may be present in entries using the object class.

Viewing object classes

You can view the object classes in the schema using either the Web Administration Tool, the preferred method or using the command line.

Using Web Administration:

Expand **Schema management** in the navigation area and click **Manage object classes**.

A read-only panel is displayed that enables you to view the object classes in the schema and their characteristics. The object classes are displayed in alphabetical order. Use the table options to locate the object class that you want to view. See “Using tables in the Web Administration Tool” on page 23 for information on how to use these options.

After you have located the object class that you want, you can view its type, required attributes, and optional attributes. Expand the drop-down menus for required attributes and optional attributes to see the full listings for each characteristic.

To view additional information about the object class:

1. Select the object class.
2. Click **View**.

The **View object class** panel is displayed.

This panel has two tabs. The **Formatted view** tab supplies the object class name, description, OID, object class type, superior object classes, required attributes, required inherited attributes, optional attributes and optional inherited attributes. The information is displayed in a printable format. The **Server view** tab provides the information in the format used in the attribute file on the server.

When you are finished click **Close** to return to the **Managing object classes** panel.

Using the command line:

To view the object classes contained in the schema issue the command:

```
idsldapsearch -b cn=schema -s base objectclass=* objectclasses
```

Adding an object class

Using Web Administration:

If you have not done so already, expand **Schema management** in the navigation area, then click **Manage object classes**. To create a new object class:

1. Click **Add**.

Note: You can also access this panel by expanding **Schema management** in the navigation area, and then clicking **Add an object class**.

2. At the **General properties** tab:

- Enter the **Object class name**. This is a required field, and is descriptive of the function of the object class. For example, **tempEmployee** for an object class used to track temporary employees.
- Enter a **Description** of the object class, for example, **The object class used for temporary employees**.
- Enter the **OID** for the object class. This is a required field. See “Object identifier (OID)” on page 33. If you do not have an OID, you can use the **Object class name** appended with **oid**. For example, if the object class name is **tempEmployee**, then the OID is **tempEmployeeoid**. You can change the value of this field.
- Select one or more **Superior object classes** from the menu . This selection determines the object class or classes from which other attributes are inherited. Typically the **Superior object classes** is **top**, however, it can be

another object class, or used in conjunction with other object classes. For example, a superior object classes for **tempEmployee** might be **top** and **ePerson**.

- Select an **Object class type**. See “Object class type” on page 33 for additional information about object class types.
 - Click the **Attributes** tab to specify the required and the optional attributes for the object class and view the inherited attributes, or click **OK** to add the new object class or click **Cancel** to return to **Manage object classes** without making any changes.
3. At the **Attributes** tab:
 - Select an attribute from the alphabetical list of **Available attributes** and click **Add to required** to make the attribute required or click **Add to optional** to make the attribute optional for the object class. The attribute is displayed in the appropriate list of selected attributes.
 - Repeat this process for all the attributes you want to select.
 - You can move an attribute from one list to another or delete the attribute from the selected lists by selecting it and clicking the appropriate **Move to** or **Remove** button.
 - You can view the lists of required and optional inherited attributes. Inherited attributes are based on the **Superior object classes** selected on the **General** tab. You cannot change the inherited attributes. However, if you change the **Superior object classes** on the **General** tab, a different set of inherited attributes is displayed.
 4. Click **OK** to add the new object class or click **Cancel** to return to **Manage object classes** without making any changes.

Note: If you clicked **OK** on the **General** tab without adding any attributes, you can add attributes by editing the new object class.

Using the command line:

To add an object class using the command line, issue the following command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where <filename> contains:

```
dn: cn=Schema
changetype: modify
add: objectclasses
objectclasses: ( <myobjectClass-oid> NAME '<myObjectClass>' DESC '<An object class
I defined for my LDAP application>' SUP '<objectclassinheritance>'
<objectclasstype> MUST (<attribute1> $ <attribute2>)
MAY (<attribute3> $ <attribute4> )
```

Editing an object class

Not all schema changes are allowed. See “Disallowed schema changes” on page 52 for change restrictions.

Using Web Administration:

If you have not done so already, expand **Schema management** in the navigation area, then click **Manage object classes**. To edit an object class:

1. Click the radio button next to the object class that you want to edit.
2. Click **Edit**.
3. Select a tab:
 - Use the **General** tab to:

- Modify the **Description**.
 - Change the **Superior object classes**. Select one or more superior object classes from the menu . This determines the object class or classes from which other attributes are inherited. Typically the superior object class is **top**, however, it can be another object class, or used in conjunction with other object classes. For example, a superior object classes for **tempEmployee** might be **top** and **ePerson**.
 - Change the **Object class type**. Select an object class type. See “Object class type” on page 33 for additional information about object class types.
 - Click the **Attributes** tab to change the required and the optional attributes for the object class and view the inherited attributes, or click **OK** to apply your changes or click **Cancel** to return to **Manage object classes** without making any changes.
- Use the **Attributes** tab to:

Select an attribute from the alphabetical list of **Available attributes** and click **Add to required** to make the attribute required or click **Add to optional** to make the attribute optional for the object class. The attribute is displayed in the appropriate list of selected attributes.

Repeat this process for all the attributes you want to select.

You can move an attribute from one list to another or delete the attribute from the selected lists by selecting it and clicking the appropriate **Move to** or **Remove** button.

You can view the lists of required and optional inherited attributes. Inherited attributes are based on the superior object classes selected on the **General** tab. You cannot change the inherited attributes. However, if you change the **Superior object classes** on the **General** tab, a different set of inherited attributes is displayed.
4. Click **OK** to apply the changes or click **Cancel** to return to **Manage object classes** without making any changes.

Using the command line:

View the object classes contained in the schema issue the command:

```
idsldapsearch -b cn=schema -s base objectclass=* objectclasses
```

To change an object class using the command line, issue the following command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where <filename>contains:

```
dn: cn=schema
changetype: modify
replace: objectclasses
objectclasses: ( <myobjectClass-oid> NAME '<myObjectClass>' DESC '<An object class
I defined for my LDAP application>' SUP '<newsuperiorclassobject>'
<newobjectclasstype> MUST (<attribute1> $ <attribute2>)
MAY (<attribute3> $ <attribute4> )
```

Note: Modify-replace requests directed at the "cn=schema" entry have a special behavior that is not true for other entries. Normally a modify-replace replaces all values of the specified attribute, with the set of new values specified in the modify operation. However, when applied to the schema, only the referenced value is replaced. If this was not the case, this example would replace the definition of "myObjectClass", but also delete the definitions of all other objectclasses. The same behavior is true for modify-replace operations to replace attributetypes values.

Copying an object class

Using Web Administration:

If you have not done so already, expand **Schema management** in the navigation area, then click **Manage object classes**. To copy an object class:

1. Click the radio button next to the object class that you want to copy.
2. Click **Copy**.
3. Select a tab:
 - Use the **General** tab to:
 - Type the new **object class name**. For example, you might copy **tempEmployee** as **tempEmployee2**.
 - Modify the **Description**.
 - Type the new **OID**. See “Object identifier (OID)” on page 33. If you do not have a registered OID for the object class you have copied, you can create one for your local use. For example, if your new object class is called **tempEmployee2** you might use **tempEmployee2oid** as the OID.
 - Change the **Superior object classes**. Select one or more superior object classes from the menu . This determines the object class or classes from which other attributes are inherited. Typically the superior object class is **top**, however, it can be another object class, or used in conjunction with other object classes. For example, a superior object classes for **tempPerson2** might be **top** and **ePerson**.
 - Change the **Object class type**. Select an object class type. See “Object class type” on page 33 for additional information about object class types.
 - Click the **Attributes** tab to change the required and the optional attributes for the object class and view the inherited attributes, or click **OK** to apply your changes or click **Cancel** to return to **Manage object classes** without making any changes.
 - Use the **Attributes** tab to:

Select an attribute from the alphabetical list of **Available attributes** and click **Add to required** to make the attribute required or click **Add to optional** to make the attribute optional for the object class. The attribute is displayed in the appropriate list of selected attributes.

Repeat this process for all the attributes you want to select.

You can move an attribute from one list to another or delete the attribute from the selected lists by selecting it and clicking the appropriate **Move to** or **Remove** button.

You can view the lists of required and optional inherited attributes. Inherited attributes are based on the superior object classes selected on the **General** tab. You cannot change the inherited attributes. However, if you change the **Superior object classes** on the **General** tab, a different set of inherited attributes is displayed.
4. Click **OK** to apply the changes or click **Cancel** to return to **Manage object classes** without making any changes.

Using the command line:

View the object classes contained in the schema issue the command:

```
idsldapsearch -b cn=schema -s base objectclass=* objectclasses
```

Select the object class that you want to copy. Use an editor to change the appropriate information and save the changes to *<filename>*. The issue the following command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where <filename>contains:

```
dn: cn=schema
changetype: modify
replace: objectclasses
objectclasses: ( <mynewobjectClass-oid> NAME '<mynewObjectClass>'
DESC '<A new object class I copied for my LDAP application>'
SUP '<superiorclassobject>'<objectclasstype>
MUST (<attribute1> $ <attribute2>)
MAY (>attribute3> $ <attribute4> $ <attribute3>) )
```

Deleting an object class

Not all schema changes are allowed. See “Disallowed schema changes” on page 52 for change restrictions.

Using Web Administration:

If you have not done so already, expand **Schema management** in the navigation area, then click **Manage object classes**. To delete an object class:

1. Click the radio button next to the object class that you want to delete.
2. Click **Delete**.
3. You are prompted to confirm deletion of the object class. Click **OK** to delete the object class or click **Cancel** to return to **Manage object classes** without making any changes.

Using the command line:

View the object classes contained in the schema issue the command:

```
idsldapsearch -b cn=schema -s base objectclass=* objectclasses
```

Select the object class you want to delete and issue the following command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where <filename>contains:

```
dn: cn=schema
changetype: modify
delete: objectclasses
objectclasses: ( <myobjectClass-oid> NAME '<myObjectClass>'
DESC '<An object class I defined for my LDAP application>'
SUP '<objectclassinheritance>' <objectclasstype >
MUST (<attribute1> $ <attribute2>) >
MAY (<attribute3> $ <attribute4>) )
```

Working with attributes

Each directory entry has a set of attributes associated with it through its object class. While the object class describes the type of information that an entry contains, the actual data is contained in attributes. An attribute is represented by one or more name-value-pairs that hold specific data element such as a name, an address, or a telephone number. The IBM Tivoli Directory Server represents data as name-value-pairs, a descriptive attribute, such as commonName (cn), and a specific piece of information, such as John Doe.

For example, the entry for John Doe might contain several attribute name-value-pairs.

```
dn: uid=jdoe, ou=people, ou=mycompany, c=us,
objectClass: top
objectClass: person
```

```
objectClass: organizationalPerson
cn: John Doe
sn: Doe
givenName: Jack
givenName: John
```

While the standard attributes are already defined in the schema file, you can create, edit, copy, or delete attributes to suit the needs of your organization.

The IBMAttributeTypes attribute type

The IBMAttributeTypes attribute can be used to define schema information not covered by the LDAP Version 3 standard for attributes. Values of IBMAttributeTypes must comply with the following grammar:

```
IBMAttributeTypesDescription = "(" whsp
    numericoid whsp
    [ "DBNAME" qdescrs ] ; at most 2 names (table, column)
    [ "ACCESS-CLASS" whsp IBMAccessClass whsp ]
    [ "LENGTH" wlen whsp ] ; maximum length of attribute
    [ "EQUALITY" whsp ] ; create index for matching rule
    [ "ORDERING" whsp ] ; create index for matching rule
    [ "APPROX" whsp ] ; create index for matching rule
    [ "SUBSTR" whsp ] ; create index for matching rule
    [ "REVERSE" whsp ] ; reverse index for substring
    whsp ")"

IBMAccessClass =
    "NORMAL" / ; this is the default
    "SENSITIVE" /
    "CRITICAL" /
    "RESTRICTED" /
    "SYSTEM" /
```

Numericoid

Used to correlate the value in attributetypes with the value in IBMAttributeTypes.

DBNAME

You can provide two names at the most. The first is the table name used for this attribute. The second is the column name used for the fully normalized value of the attribute in the table. If you provide only one name, it is used as the table name as well as the column name. If you do not provide any DBNAMEs, then the short attribute name is used (from the attributetypes).

ACCESS-CLASS

Attributes requiring similar permissions for access are grouped together in classes. Attributes are mapped to their attribute classes in the directory schema file. These classes are discreet; access to one class does not imply access to another class. Permissions are set with regard to the attribute access class as a whole. The permissions set on a particular attribute class apply to all attributes within that access class unless individual attribute access permissions are specified.

IBM defines five attribute classes that are used in evaluation of access to user attributes: **normal**, **sensitive**, **critical**, **system**, and **restricted**. As examples, the attribute **commonName** belongs to the normal class, and the attribute **userPassword** belongs to the critical class. User defined attributes belong to the normal access class unless otherwise specified. See "Rights" on page 313 for more information.

If ACCESS-CLASS is omitted, it defaults to normal.

LENGTH

The maximum length of this attribute. The length is expressed as the number of bytes. (IBM Directory Version 6.0 has a provision for increasing the length of an attribute.) In the attributetypes value, the string:

```
( attr-oid ... SYNTAX syntax-oid{len} ... )
```

can be used to indicate that the attributetype with oid attr-oid has a maximum length.

If the length of an attribute needs to be reduced, see “Manual procedure for changing existing attributes” on page 47.

EQUALITY, ORDERING, APPROX, SUBSTR, REVERSE

If any of these attributes are used, an index is created for the corresponding matching rule. For good search performance, an EQUALITY index should be specified for any attribute that will be used in search filters.

Matching rules

A matching rule provides guidelines for string comparison during a search operation. These rules are divided into three categories:

- Equality
- Ordering
- Substring

Table 1.

Equality matching rules		
Matching Rule	OID	Syntax
caseExactIA5Match	1.3.6.1.4.1.1466.109.114.1	Directory String syntax
caseExactMatch	2.5.13.5	Directory String syntax
caseIgnoreIA5Match	1.3.6.1.4.1.1466.109.114.2	IA5 String syntax
caseIgnoreMatch	2.5.13.2	Directory String syntax
distinguishedNameMatch	2.5.13.1	DN - distinguished name
generalizedTimeMatch	2.5.13.27	Generalized Time syntax
ibm-entryUuidMatch	1.3.18.0.2.22.2	Directory String syntax
integerFirstComponentMatch	2.5.13.29	Integer syntax - integral number
integerMatch	2.5.13.14	Integer syntax - integral number
objectIdentifierFirstComponentMatch	2.5.13.30	String for containing OIDs. The OID is a string containing digits (0-9) and decimal points (.).
objectIdentifierMatch	2.5.13.0	String for containing OIDs. The OID is a string containing digits (0-9) and decimal points (.).

Table 1. (continued)

Equality matching rules		
Matching Rule	OID	Syntax
octetStringMatch	2.5.13.17	Directory String syntax
telephoneNumberMatch	2.5.13.20	Telephone Number syntax
uTCTimeMatch	2.5.13.25	UTC Time syntax

Table 2.

Ordering matching rules		
Matching rule	OID	Syntax
caseExactOrderingMatch	2.5.13.6	Directory String syntax
caseIgnoreOrderingMatch	2.5.13.3	Directory String syntax
distinguishedNameOrderingMatch	1.3.18.0.2.4.405	DN - distinguished name
generalizedTimeOrderingMatch	2.5.13.28	Generalized Time syntax

Table 3.

Substring matching rules		
Matching rule	OID	Syntax
caseExactSubstringsMatch	2.5.13.7	Directory String syntax
caseIgnoreSubstringsMatch	2.5.13.4	Directory String syntax
telephoneNumberSubstringsMatch	2.5.13.21	Telephone Number syntax

Note: UTC-Time is time string format defined by ASN.1 standards. See ISO 8601 and X680. Use this syntax for storing time values in UTC-Time format. uTCTimeMatch is a deprecated matching rule. Use generalizedTimeMach instead. See “Generalized and UTC time” on page 64.

Indexing rules

Index rules attached to attributes make it possible to retrieve information faster. If only the attribute is given, no indexes are maintained. IBM Directory provides the following indexing rules:

- Equality
- Ordering
- Approximate
- Substring
- Reverse

Indexing rules specifications for attributes:

Specifying an indexing rule for an attribute controls the creation and maintenance of special indexes on the attribute values. This greatly improves the response time to searches with filters which include those attributes. The five possible types of indexing rules are related to the operations applied in the search filter.

Equality

Applies to the following search operations:

- equalityMatch '='

For example:

```
"cn = John Doe"
```

Ordering

Applies to the following search operation:

- greaterOrEqual '>='
- lessOrEqual '<='

For example:

```
"sn >= Doe"
```

Approximate

Applies to the following search operation:

- approxMatch '~='

For example:

```
"sn ~= doe"
```

Substring

Applies to the search operation using the substring syntax:

- substring '*'

For example:

```
"sn = McC*"
```

```
"cn = J*Doe"
```

Reverse

Applies to the following search operation:

- '*' substring

For example:

```
"sn = *baugh"
```

At a minimum, it is recommended that you specify equality indexing on any attributes that are to be used in search filters.

Viewing attributes

You can view the attributes in the schema using either the Web Administration Tool, the preferred method or using the command line.

Using Web Administration:

Expand **Schema management** in the navigation area and click **Manage attributes**. A read-only panel is displayed that enables you to view the attributes in the schema and their characteristics. The attributes are displayed in alphabetical order. Use the table options to locate the attribute that you want to view. See "Using tables in the Web Administration Tool" on page 23 for information on how to use these options.

After you have located the attribute that you want, you can view its syntax, whether it is multi-valued, and the object classes that contain it. Expand the drop-down menu for object classes to see the list of object classes for the attribute.

To view additional information about the attribute:

1. Select the attribute.

2. Click **View**.

The **View attributes** panel is displayed.

This panel has two tabs. The **Formatted view** tab supplies the attribute name, description, OID, superior attribute, syntax, attribute length, multiple values enabled status, matching rules, IBM extensions, and indexing rules. The information is displayed in a printable format. The Server view tab provides the information in the format used in the attribute file on the server.

When you are finished click **Close** to return to the IBM Tivoli Directory Server **Manage attributes** panel.

Using the command line:

To view the attributes contained in the schema issue the command:

```
idsldapsearch -b cn=schema -s base objectclass=* attributeTypes IBMAttributeTypes
```

Adding an attribute

Use either of the following methods to create a new attribute. The Web Administration Tool is the preferred method.

Using Web Administration:

If you have not done so already, expand **Schema management** in the navigation area, then click **Manage attributes**. To create a new attribute:

1. Click **Add**.

Note: You can also access this panel by expanding **Schema management** in the navigation area, then click **Add an attribute**.

2. Enter the **Attribute name**, for example, **tempId**. This is a required field and must begin with an alphabetical character.
3. Enter a **Description** of the attribute, for example, **The ID number assigned to a temporary employee**.
4. Enter the **OID** for the attribute. This is a required field. See “Object identifier (OID)” on page 33. If you do not have a registered OID, you can use the attribute name appended with oid. For example, if the attribute name is **tempID**, then the OID is **tempIDoid**. You can change the value of this field.
5. Select a **Superior attribute** from the drop-down list. The superior attribute determines the attribute from which properties are inherited.
6. Select a **Syntax** from the drop-down list. See “Attribute syntax” on page 50 for additional information about syntax.
7. Enter an **Attribute length** that specifies the maximum length of this attribute. The length is expressed as the number of bytes. The default value is 240.
8. Select the **Allow multiple values** check box to enable the attribute to have multiple values. See the glossary entry 609 for additional information about multiple values.
9. Select a matching rule from the each of the drop-down menus for **equality**, **ordering**, and **substring** matching rules. See the “Matching rules” on page 41 for a complete listing of matching rules.
10. Click the **IBM extensions** tab to specify additional extensions for the attribute, or click **OK** to add the new attribute or click **Cancel** to return to **Manage attributes** without making any changes.
11. At the **IBM extensions** tab:

- Enter the **DB2 table name** . This table name can be up to 128 bytes in length without truncating. The server generates the DB2 table name if this field is left blank. If you enter a DB2 table name, you must also enter a DB2 column name. For servers with version earlier than IBM Tivoli Directory Server version 6.0, the length is restricted to 16 bytes without truncating.
- Modify the **DB2 column name**. The server generates the DB2 column name if this field is left blank. If you enter a DB2 column name, you must also enter a DB2 table name. This column name can be up to 16 bytes in length without truncating.
- Set the **Security class** by selecting **normal**, **sensitive**, or **critical** from the drop-down list. See the Security class section under 319 for information about security classes.
- Set the **Indexing rules** by selecting one or more indexing rules. See “Indexing rules” on page 42 for additional information about indexing rules.

Note: At a minimum, it is recommended that you specify **Equality** indexing on any attributes that are to be used in search filters.

12. Click **OK** to add the new attribute or click **Cancel** to return to **Manage attributes** without making any changes.

Note: If you clicked OK on the General tab without adding any extensions, you can add extensions by editing the new attribute.

Using the command line:

The following example adds an attribute type definition for an attribute called "myAttribute", with Directory String syntax (see "Attribute syntax" on page 50) and Case Ignore Equality matching (see "Matching rules" on page 41). The IBM-specific part of the definition says that the attribute data is stored in a column named "myAttrColumn" in a table called "myAttrTable". If these names were not specified, both the column and table name would have defaulted to "myAttribute". The attribute is assigned to the "normal" access class, and values have a maximum length of 200 bytes.

```
idsldapmodify -D <admin> -w <adminpw> -i myschema.ldif
```

where the **myschema.ldif** file contains:

```
dn: cn=schema
changetype: modify
add: attributetypes
attributetypes: ( myAttribute-oid NAME ( 'myAttribute' )
DESC 'An attribute I defined for my LDAP application'
EQUALITY 2.5.13.2 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
{200} USAGE userApplications )
-
add: ibmattributetypes
ibmattributetypes: ( myAttribute-oid DBNAME ( 'myAttrTable' 'myAttrColumn' )
ACCESS-CLASS normal LENGTH 200 )
```

Note: In this example, there are two locations where "length" can be specified. In this example, 200 is the specified length. For example:

- {200} USAGE userApplications)
- ACCESS-CLASS normal LENGTH 200)

Both of these pieces of code demonstrate how to specify length. If length is specified in either of these locations, then they both must match.. See “idsldapmodify, ldapmodify, idsldapadd, ldapadd” on page 396 for more information about this command.

Editing an attribute

Not all schema changes are allowed. See “Disallowed schema changes” on page 52 for change restrictions.

Any part of a definition can be changed before you have added entries that use the attribute. After you have added entries that use the attribute, you can use the edit procedure to change the indexing rules and to increase the size of the attribute length. You can also change to enable multiple values.

Note: You can disable multiple values only if the existing entries are single-valued. You cannot disable the multi-value option if any of the existing entries are multi-valued.

Use either of the following methods to edit an attribute. The Web Administration Tool is the preferred method.

Using Web Administration:

If you have not done so already, expand **Schema management** in the navigation area, then click **Manage attributes**. To edit an attribute:

1. Click the radio button next to the attribute that you want to edit.
2. Click **Edit**.
3. Select a tab:
 - Use the **General** tab to:
 - Select a tab, either:
 - **General** to:
 - Modify the **Description**.
 - Change the **Superior attribute**.
 - Change the **Syntax**.
 - Set the **Attribute length**.
 - Note:** You can only increase the size of the attribute length. If you need to reduce the size of the attribute length, you must perform additional steps before editing the attribute. See “Manual procedure for changing existing attributes” on page 47.
 - Change the **Multiple value** settings.
 - Select a **Matching rule**.
 - Click the **IBM extensions** tab to edit the extensions for the attribute, or click **OK** to apply your changes or click **Cancel** to return to **Manage attributes** without making any changes.
 - **IBM extensions** (if you are connected to an IBM Tivoli Directory Server) to:
 - Change the **Security class**.
 - Note:** You cannot change the security class of attributes that have a security classification of system or restricted.
 - Change the **Indexing rules**.

- Click **OK** to apply your changes or click **Cancel** to return to **Manage attributes** without making any changes.
4. When you are finished editing the attributes, click **Close** to return to **Introduction** panel.

Using the command line:

This example adds indexing to the attribute, so that searching on it is faster. Use the `idsldapmodify` command and the LDIF file to change the definition.

Note: You can only increase the size of the attribute length. If you need to reduce the size of the attribute length, you must perform additional steps before editing the attribute. See “Manual procedure for changing existing attributes.”

```
idsldapmodify -D <admin dn> -w <admin pw> -i myschemachange.ldif
```

Where the `myschemachange.ldif` file contains:

```
dn: cn=schema
changetype: modify
replace: attributetypes
attributetypes: ( myAttribute-oid NAME ( 'myAttribute' ) DESC 'An attribute
                  I defined for my LDAP application' EQUALITY 2.5.13.2
                  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 {200} USAGE userApplications )
-
replace: ibmattributetypes
ibmattributetypes: ( myAttribute-oid DBNAME ( 'myAttrTable' 'myAttrColumn' )
                    ACCESS-CLASS normal LENGTH 200 EQUALITY SUBSTR )
```

Note: Both portions of the definition (**attributetypes** and **ibmattributetypes**) must be included in the replace operation, even though only the **ibmattributetypes** section is changing. The only change is adding “EQUALITY SUBSTR” to the end of the definition to request indexes for equality and substring matching.

See “idsldapmodify, ldapmodify, idsldapadd, ldapadd” on page 396 for more information about this command.

Manual procedure for changing existing attributes

If an attribute definition needs to be changed and the table has already been populated for this attribute, perform the following operations:

1. Use the `idsdb2ldif` utility to export the directory data into an LDIF file.
2. Unconfigure the database.

```
idsucfgdb -I <instance_name> -r
```
3. Change the attribute definition in the schema file. See “Editing an attribute” on page 46.
4. Configure the database.
5. Use either the `idsldif2db` or the `idsbulkload` utility to import the data into the database.

Copying an attribute

Use either of the following methods to copy an attribute. The Web Administration Tool is the preferred method.

Using Web Administration:

If you have not done so already, expand **Schema management** in the navigation area, then click **Manage attributes**. To copy an attribute:

1. Click the radio button next to the attribute that you want to copy.

2. Click **Copy**.
3. Type the name of the new attribute in the **Attribute name** field. For example, you might copy **tempID** as **tempID2**.
4. Modify a **Description** of the attribute, for example, **The ID number assigned to a temporary employee**.
5. Type the new **OID**. See “Object identifier (OID)” on page 33. If you do not have a registered OID for the attribute you have copied, you can create one for your local use. For example, if your new attribute is called **tempID2** you might use **tempID2oid** as the OID.
6. Select a **Superior attribute** from the drop-down list. The superior attribute determines the attribute from which properties are inherited.
7. Select a **Syntax** from the drop-down list. See “Attribute syntax” on page 50 for additional information about syntax.
8. Enter a **Attribute length** that specifies the maximum length of this attribute. The length is expressed as the number of bytes.
9. Select the **Allow multiple values** check box to enable the attribute to have multiple values. See the glossary entry 609 for additional information about multiple values.
10. Select a matching rule from the each of the drop-down menus for equality, ordering, and substring matching rules. See the “Matching rules” on page 41 for a complete listing of matching rules.
11. Click the **IBM extensions** tab to modify additional extensions for the attribute, or click **OK** to apply your changes or click **Cancel** to return to **Manage attributes** without making any changes.
12. At the **IBM extensions** tab:
 - Enter the **DB2 table name** . This table name can be up to 128 bytes in length without truncating. The server generates the DB2 table name if this field is left blank. If you enter a DB2 table name, you must also enter a DB2 column name. For servers with version earlier than IBM Tivoli Directory Server version 6.0, the length is restricted to 16 bytes without truncating.
 - Enter the **DB2 column name**. This column name can be up to 16 bytes in length without truncating. The server generates the DB2 column name if this field is left blank. If you enter a DB2 column name, you must also enter a DB2 table name.
 - Modify the **Security class** by selecting **normal**, **sensitive**, or **critical** from the drop-down list.

Note: You cannot change the security class of attributes that have a security classification of system or restricted.
 - Modify the **Indexing rules** by selecting one or more indexing rules. See “Indexing rules” on page 42 for additional information about indexing rules.

Note: At a minimum, it is recommended that you specify Equal indexing on any attributes that are to be used in search filters.
13. Click **OK** to apply your changes or click **Cancel** to return to **Manage attributes** without making any changes.

Note: If you clicked **OK** on the **General** tab without adding any extensions, you can add or modify extensions by editing the new attribute.

Using the command line:

View the attributes contained in the schema issue the command:

```
idsldapsearch -b cn=schema -s base objectclass=* attributeTypes IBMAttributeTypes
```

Select the attribute that you want to copy. Use an editor to change the appropriate information and save the changes to *<filename>*. Then issue the following command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where *<filename>* contains:

```
dn: cn=schema
changetype: modify
add: attributetypes
attributetypes: ( <mynewAttribute-oid> NAME 'mynewAttribute' DESC '<A new
                  attribute I copied for my LDAP application>' EQUALITY 2.5.13.2
                  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 {200} USAGE userApplications )
-
add: ibmattributetypes
ibmattributetypes: ( myAttribute-oid DBNAME ( 'myAttrTable' 'myAttrColumn' )
                    ACCESS-CLASS normal LENGTH 200 )
```

Deleting an attribute

Not all schema changes are allowed. See “Disallowed schema changes” on page 52 for change restrictions.

Use either of the following methods to delete an attribute. The Web Administration Tool is the preferred method.

Using Web Administration:

If you have not done so already, expand **Schema management** in the navigation area, then click **Manage attributes**. To delete an attribute:

1. Click the radio button next to the attribute that you want to delete.
2. Click **Delete**.
3. You are prompted to confirm deletion of the attribute. Click **OK** to delete the attribute or click **Cancel** to return to **Manage attributes** without making any changes.

Using the command line:

```
idsldapmodify -D <adminDN> -w <adminPW> -i myschemadelete.ldif
```

Where the **myschemadelete.ldif** file includes:

```
dn: cn=schema
changetype: modify
delete: attributetypes
attributetypes: ( myAttribute-oid )
-
delete: ibmattributetypes
ibmattributetypes: ( myAttribute-oid )
```

See “idsldapmodify, ldapmodify, idsldapadd, ldapadd” on page 396 for more information about this command.

Attribute syntax

Attribute syntax identifies the required format of the data.

Table 4.

Syntax	OID
Attribute Type Description syntax	1.3.6.1.4.1.1466.115.121.1.3
Binary - octet string	1.3.6.1.4.1.1466.115.121.1.5
Boolean - TRUE/FALSE	1.3.6.1.4.1.1466.115.121.1.7
Directory String syntax	1.3.6.1.4.1.1466.115.121.1.15
DIT Content Rule Description syntax	1.3.6.1.4.1.1466.115.121.1.16
DITStructure Rule Description syntax	1.3.6.1.4.1.1466.115.121.1.17
DN - distinguished name	1.3.6.1.4.1.1466.115.121.1.12
Generalized Time syntax	1.3.6.1.4.1.1466.115.121.1.24
IA5 String syntax	1.3.6.1.4.1.1466.115.121.1.26
IBM Attribute Type Description	1.3.18.0.2.8.1
Integer syntax - integral number	1.3.6.1.4.1.1466.115.121.1.27
LDAP Syntax Description syntax	1.3.6.1.4.1.1466.115.121.1.54
Matching Rule Description	1.3.6.1.4.1.1466.115.121.1.30
Matching Rule Use Description	1.3.6.1.4.1.1466.115.121.1.31
Name Form Description	1.3.6.1.4.1.1466.115.121.1.35
Object Class Description syntax	1.3.6.1.4.1.1466.115.121.1.37
String for containing OIDs. The OID is a string containing digits (0-9) and decimal points (.). See "Object identifier (OID)" on page 33.	1.3.6.1.4.1.1466.115.121.1.38
Telephone Number syntax	1.3.6.1.4.1.1466.115.121.1.50
UTC Time syntax. UTC-Time is time string format defined by ASN.1 standards. See ISO 8601 and X680. Use this syntax for storing time values in UTC-Time format. See "Generalized and UTC time" on page 64.	1.3.6.1.4.1.1466.115.121.1.53

The subschema entries

There is one subschema entry per server. All entries in the directory have an implied `subschemaSubentry` attribute type. The value of the `subschemaSubentry` attribute type is the DN of the subschema entry that corresponds to the entry. All entries under the same server share the same subschema entry, and their `subschemaSubentry` attribute type has the same value. The subschema entry has the hardcoded DN `'cn=schema'`.

The subschema entry belongs to the object classes `'top'`, `'subschema'`, and `'IBMsubschema'`. The `'IBMsubschema'` object class has no `MUST` attributes and one `MAY` attribute type (`'IBMattributeTypes'`).

The IBMsubschema object class

The `IBMsubschema` object class is used only in the subschema entry as follows:


```
( <objectClass-oid-TBD> NAME 'IBMSubschema' AUXILIARY
  MAY IBMAttributeTypes )
```

Schema queries

The `ldap_search()` API can be used to query the subschema entry, as shown in the following example:

```
DN          : "cn=schema"
search scope : base
filter      : objectclass=subschema or objectclass=*
```

This example retrieves the full schema. To retrieve all of the values of selected attribute types, use the `attrs` parameter in `ldap_search`. You cannot retrieve only a specific value of a specific attribute type.

See the *IBM Directory Version 6.0: Client SDK Programming Reference* for more information about the `ldap_search` API.

Dynamic schema

To perform a dynamic schema change, use the `ldap_modify` API with a DN of "cn=schema". It is permissible to add, delete, or replace only one schema entity (for example, an attribute type or an object class) at a time.

To delete a schema entity, provide the oid in parentheses:

```
( oid )
```

You can also provide a full description. In either case, the matching rule used to find the schema entity to delete is `objectIdentifierFirstComponentMatch`.

To add or replace a schema entity, you **MUST** provide a LDAP Version 3 definition and you **MAY** provide the IBM definition. In all cases, you must provide only the definition or definitions of the schema entity that you want to affect.

For example, to delete the attribute type 'cn' (its OID is 2.5.4.3), use `ldap_modify()` with:

```
LDAPMod attr;
LDAPMod *attrs[] = { &attr, NULL };
char *vals [] = { "( 2.5.4.3 )", NULL };
attr.mod_op = LDAP_MOD_DELETE;
attr.mod_type = "attributeTypes";
attr.mod_values = vals;
ldap_modify_s(ldap_session_handle, "cn=schema", attrs);
```

To add a new attribute type bar with OID 20.20.20 that has a NAME of length 20 chars:

```
char *vals1[] = { "( 20.20.20 NAME 'bar' SUP NAME )", NULL };
char *vals2[] = { "( 20.20.20 LENGTH 20 )", NULL };
LDAPMod attr1;
LDAPMod attr2;
LDAPMod *attrs[] = { &attr1, &attr2, NULL };
attr1.mod_op = LDAP_MOD_ADD;
attr1.mod_type = "attributeTypes";
attr1.mod_values = vals1;
attr2.mod_op = LDAP_MOD_ADD;
attr2.mod_type = "IBMAttributeTypes";
attr2.mod_values = vals2;
ldap_modify_s(ldap_session_handle, "cn=schema", attrs);
```

Note: You cannot change the ACCESS-CLASS type to or from "system" or "restricted".

See "Working with attributes" on page 39 for examples using the Web Administration Tool and the `idsldapmodify` command.

See the *IBM Directory Version 6.0: Client SDK Programming Reference* for more information about the `ldap_modify` API.

Access controls

Dynamic schema changes can be performed only by a replication supplier, the server administrator or a member of an administrator group.

Replication

Schema replication needs to be explicitly setup on `cn=ibmpolicies` to have the changes under `cn=schema` replicated to the specified replicationAgreements. In previous releases, schema changes were propagated to all the agreements mentioned in the directory server. However, IBM Tivoli Directory Server version 6.0 schema changes are propagated only to those agreements, that occur below `cn=ibmpolicies` and to no other agreements occurring in the Directory Information Tree (DIT).

When a dynamic schema change is performed, it is replicated just like any other `ldap_modify` operation. See "Replicating schema and password policy updates" on page 171.

See Chapter 13, "Replication," on page 161 for more additional information.

Disallowed schema changes

Not all schema changes are allowed. Change restrictions include the following:

- Any change to the schema must leave the schema in a consistent state.
- An attribute type that is a supertype of another attribute type may not be deleted. An attribute type that is a "MAY" or a "MUST" attribute type of an object class may not be deleted.
- An object class that is a superclass of another may not be deleted.
- Attribute types or object classes that refer to nonexisting entities (for example, syntaxes or object classes) cannot be added.
- Attribute types or object classes cannot be modified in such a way that they end up referring to nonexisting entities (for example, syntaxes or object classes).

Changes to the schema that affect the operation of the server are not allowed. The following schema definitions are required by the directory server. They must not be changed.

Object classes

The following object class definitions must not be modified:

- `accessGroup`
- `accessRole`
- `alias`
- `referral`
- `replicaObject`
- `top`

- ibm-slapdPwdPolicyAdmin
- ibm-pwdPolicyExt
- pwdPolicy

Attributes

The following attribute definitions must not be modified:

Operational attributes

There are attributes that have special meaning to the directory server, known as operational attributes. These are attributes that are maintained by the server, and either reflect information the server manages about an entry, or affect server operation. These attributes have special characteristics:

- The attributes are not returned by a search operation unless they are specifically requested (by name) in the search request.
- These attributes cannot be deleted.
- The attributes are not part of any object class. The server controls what entries have the attributes.

The following lists of operational attributes are supported by the IBM Tivoli Directory Server:

- aclEntry
- aclPropagate
- aclSource
- aliasedObjectName, aliasedentryName
- createTimeStamp
- creatorsName
- entryOwner
- hasSubordinates
- ibm-allGroups
- ibm-allMembers
- ibm-capabilitiesubentry
- ibm-effectiveAcl
- ibm-entryChecksum
- ibm-entryChecksumOp
- ibm-entryUuid
- ibm-filterAclEntry
- ibm-filterAclInherit
- ibm-pwdAccountLocked
- ibm-replicationChangeLDIF
- ibm-replicationFailedChangeCount
- ibm-replicationFailedChanges
- ibm-replicationIsQuiesced
- ibm-replicationLastActivationTime
- ibm-replicationLastChangeId
- ibm-replicationLastFinishTime
- ibm-replicationLastGlobalChangeId
- ibm-replicationLastResult
- ibm-replicationLastResultAdditional

- ibm-replicationNextTime
- ibm-replicationPendingChangeCount
- ibm-replicationPendingChanges
- ibm-replicationperformance
- ibm-replicationState
- ibm-replicationThisServerIsMaster
- ibm-searchSizeLimit
- ibm-searchTimeLimit
- ibm-slapdCryptoSalt
- modifiersName
- modifyTimestamp
- numSubordinates
- ownerPropagate
- ownerSource
- pwdAccountLockedTime
- pwdChangedTime
- pwdExpirationWarned
- pwdFailureTime
- pwdGraceUseTime
- pwdHistory
- pwdReset
- subschemaSubentry
- subtreeSpecification

See Appendix H, “IBM Tivoli Directory Server 6.0 required attribute definitions,” on page 501 for more information about these attributes.

Restricted attributes

The following lists of restricted attributes are supported by the IBM Tivoli Directory Server:

- aclEntry
- aclPropagate
- entryOwner
- ibm-filterAclEntry
- ibm-filterAclInherit
- ownerPropagate

Root DSE attributes

The following attributes relate to the root DSE and must not be modified:

- altServer
- changelog
- firstchangenumber
- IBMDirectoryVersion
- ibm-effectiveReplicationModel
- ibm-enabledCapabilities
- ibm-ldapservicename
- ibm-sasldigestrealmname

- ibm-serverId
- ibm-supportedCapabilities
- ibm-supportedReplicationModels
- lastchangenumber
- namingContexts
- supportedControl
- vendorName
- vendorVersion

See Appendix H, “IBM Tivoli Directory Server 6.0 required attribute definitions,” on page 501 for more information about these attributes.

Schema definition attributes

The following attributes are related to Schema definitions and must not be modified:

- attributeTypes
- ditContentRules
- ditStructureRules
- IBMAttributeTypes
- ldapSyntaxes
- matchingRules
- matchingRuleUse
- nameForms
- objectClasses
- supportedExtension
- supportedLDAPVersion
- supportedSASLMechanisms

See Appendix H, “IBM Tivoli Directory Server 6.0 required attribute definitions,” on page 501 for more information about these attributes.

Configuration attributes

The following are attributes that affect the configuration of the server. While the values can be modified, the definitions of these attributes must not be changed for the server to operate correctly

- ibm-audit
- ibm-auditAdd
- ibm-auditAttributesOnGroupEvalOp
- ibm-auditBind
- ibm-auditCompare
- ibm-auditDelete
- ibm-auditExtOp
- ibm-auditExtOpEvent
- ibm-auditFailedOpOnly
- ibm-auditGroupsOnGroupControl
- ibm-auditLog
- ibm-auditModify
- ibm-auditModifyDN
- ibm-auditSearch

- ibm-auditUnbind
- ibm-auditVersion
- ibm-pwdPolicy
- ibm-replicaConsumerConnections
- ibm-replicaConsumerId
- ibm-replicaCredentialsDN
- ibm-replicaGroup
- ibm-replicaKeyfile
- ibm-replicaKeylabel
- ibm-replicaKeypwd
- ibm-replicaMethod
- ibm-replicaReferralURL
- ibm-replicaScheduleDN
- ibm-replicaServerId
- ibm-replicaURL
- ibm-replicationBatchStart
- ibm-replicationExcludedCapability
- ibm-replicationImmediateStart
- ibm-replicationOnHold
- ibm-replicationServerIsMaster
- ibm-replicationTimesUTC
- ibm-scheduleFriday
- ibm-scheduleMonday
- ibm-scheduleSaturday
- ibm-scheduleSunday
- ibm-scheduleThursday
- ibm-scheduleTuesday
- ibm-scheduleWednesday
- ibm-slapedAclCache
- ibm-slapedAclCacheSize
- ibm-slapedAdminDN
- ibm-slapedAdminGroupEnabled
- ibm-slapedAdminPW
- ibm-slapedAllowAnon
- ibm-slapedAllReapingThreshold
- ibm-slapedAnonReapingThreshold
- ibm-slapedAuthIntegration
- ibm-slapedBoundReapingThreshold
- ibm-slapedBulkloadErrors
- ibm-slapedCachedAttribute
- ibm-slapedCachedAttributeSize
- ibm-slapedChangeLogMaxAge
- ibm-slapedChangeLogMaxEntries
- ibm-slapedCLIErrors
- ibm-slapedConfigPwdPolicyOn

- ibm-slapdCryptoSync
- ibm-slapdDB2CP
- ibm-slapdDBAlias
- ibm-slapdDbConnections
- ibm-slapdDbInstance
- ibm-slapdDbLocation
- ibm-slapdDbName
- ibm-slapdDbUserID
- ibm-slapdDbUserPW
- ibm-slapdDerefAliases
- ibm-slapdDigestAdminUser
- ibm-slapdDigestAttr
- ibm-slapdDigestRealm
- ibm-slapdDistributedDynamicGroups
- ibm-slapdDN
- ibm-slapdEnableEventNotification
- ibm-slapdEntryCacheSize
- ibm-slapdErrorLog
- ibm-slapdESizeThreshold
- ibm-slapdEThreadActivate
- ibm-slapdEThreadEnable
- ibm-slapdETimeThreshold
- ibm-slapdFilterCacheBypassLimit
- ibm-slapdFilterCacheSize
- ibm-slapdIdleTimeOut
- ibm-slapdIncludeSchema
- ibm-slapdInvalidLine
- ibm-slapdIpAddress
- ibm-slapdKrbAdminDN
- ibm-slapdKrbEnable
- ibm-slapdKrbIdentityMap
- ibm-slapdKrbKeyTab
- ibm-slapdKrbRealm
- ibm-slapdLanguageTagsEnabled
- ibm-slapdLdapCrlHost
- ibm-slapdLdapCrlPassword
- ibm-slapdLdapCrlPort
- ibm-slapdLdapCrlUser
- ibm-slapdLog
- ibm-slapdLogArchivePath
- ibm-slapdLogMaxArchives
- ibm-slapdLogOptions
- ibm-slapdLogSizeThreshold
- ibm-slapdMasterDN
- ibm-slapdMasterPW

- ibm-slapdMasterReferral
- ibm-slapdMaxEventsPerConnection
- ibm-slapdMaxEventsTotal
- ibm-slapdMaxNumOfTransactions
- ibm-slapdMaxOpPerTransaction
- ibm-slapdMaxPendingChangesDisplayed
- ibm-slapdMaxTimeLimitOfTransactions
- ibm-slapdMigrationInfo
- ibm-slapdPagedResAllowNonAdmin
- ibm-slapdPagedResLmt
- ibm-slapdPlugin
- ibm-slapdPort
- ibm-slapdProxyBackendServerDn
- ibm-slapdProxyBindMethod
- ibm-slapdProxyConnectionPoolSize
- ibm-slapdProxyDigestRealm
- ibm-slapdProxyDigestUserName
- ibm-slapdProxyDn
- ibm-slapdProxyNumPartitions
- ibm-slapdProxyPartitionBase
- ibm-slapdProxyPartitionIndex
- ibm-slapdProxyPw
- ibm-slapdProxyTargetURL
- ibm-slapdPwEncryption
- ibm-slapdReadOnly
- ibm-slapdReferral
- ibm-slapdReplConflictMaxEntrySize
- ibm-slapdReplContextCacheSize
- ibm-slapdReplDbConns
- ibm-slapdReplMaxErrors
- ibm-slapdReplicaSubtree
- ibm-slapdSchemaAdditions
- ibm-slapdSchemaCheck
- ibm-slapdSecurePort
- ibm-slapdSecurity
- ibm-slapdServerBackend
- ibm-slapdServerId
- ibm-slapdSetenv
- ibm-slapdSizeLimit
- ibm-slapdSortKeyLimit
- ibm-slapdSortSrchAllowNonAdmin
- ibm-slapdSslAuth
- ibm-slapdSslCertificate
- ibm-slapdSslCipherSpec
- ibm-slapdSslCipherSpecs

- ibm-slapdSslFIPsModeEnabled
- ibm-slapdSslFIPsProcessingMode
- ibm-slapdSSLKeyDatabase
- ibm-slapdSSLKeyDatabasePW
- ibm-slapdSslKeyRingFile
- ibm-slapdSslKeyRingFilePW
- ibm-slapdStartupTraceEnabled
- ibm-slapdSuffix
- ibm-slapdsupportedCapabilities
- ibm-slapdSupportedWebAdmVersion
- ibm-slapdSysLogLevel
- ibm-slapdTimeLimit
- ibm-slapdTraceEnabled
- ibm-slapdTraceMessageLevel
- ibm-slapdTraceMessageLog
- ibm-slapdTransactionEnable
- ibm-slapdUseProcessIdPW
- ibm-slapdVersion
- ibm-slapdWriteTimeout
- ibm-UniqueAttributeTypes
- ids-instanceDesc
- ids-instanceLocation
- ids-instanceVersion
- passwordMaxRepeatedChars
- passwordMinAlpaChars
- passwordMinDiffChars
- passwordMinOtherChars
- pwdAllowUserChange
- pwdAttribute
- pwdCheckSyntax
- pwdExpireWarning
- pwdFailureCountInterval
- pwdGraceLoginLimit
- pwdInHistory
- pwdLockout
- pwdLockoutDuration
- pwdMaxAge
- pwdMaxFailure
- pwdMinAge
- pwdMinLength
- pwdMustChange
- pwdSafeModify
- replicaBindDN
- replicaBindMethod
- replicaCredentials, replicaBindCredentials

- replicaHost
- replicaPort
- replicaUpdateTimeInterval
- replicaUseSSL

See Appendix H, “IBM Tivoli Directory Server 6.0 required attribute definitions,” on page 501 or Configuration attributes for more information about these attributes.

User application attributes

Additionally, there are several user application attributes that must not have their definitions modified:

- businessCategory
- cn, commonName
- changeNumber
- changes
- changeTime
- changeType
- deleteOldRdn
- description
- dn, distinguishedName
- globalGroupName
- ibm-changeInitiatorsName
- ibm-kn, 'ibm-kerberosName
- ibm-replCredName
- ibm-replDailySchedName
- ibm-replWeeklySchedName
- krbAliasedObjectName
- krbHintAliases
- krbPrincSubtree
- krbPrincipalName
- krbRealmName
- krbRealmName-V2
- member
- name
- newRdn
- newSuperior
- o, organizationName, organization
- objectClass
- ou, organizationalUnit, organizationalUnitName
- owner
- ref
- secretKey
- seeAlso
- targetDN

See Appendix H, “IBM Tivoli Directory Server 6.0 required attribute definitions,” on page 501 for more information about these attributes.

Syntaxes

No syntaxes are allowed to be modified.

Matching rules

No matching rules are allowed to be modified.

Schema checking

When the server is initialized, the schema files are read and checked for consistency and correctness. If the checks fail, the server fails to initialize and issues an error message. During any dynamic schema change, the resulting schema is also checked for consistency and correctness. If the checks fail, an error is returned and the change fails. Some checks are part of the grammar (for example, an attribute type can have at most one supertype, or an object class can have any number of superclasses).

The following items are checked for attribute types:

- Two different attribute types cannot have the same name or OID.
- The inheritance hierarchy of attribute types does not have cycles.
- The supertype of an attribute type must also be defined, although its definition might be displayed later, or in a separate file.
- If an attribute type is a subtype of another, they both have the same USAGE.
- All attribute types have a syntax either directly defined or inherited.
- Only operational attributes can be marked as NO-USER-MODIFICATION.

The following items are checked for object classes:

- Two different object classes cannot have the same name or OID.
- The inheritance hierarchy of object classes does not have cycles.
- The superclasses of an object class must also be defined, although its definition might appear later or in a separate file.
- The "MUST" and "MAY" attribute types of an object class must also be defined, although its definition might appear later or in a separate file.
- Every structural object class is a direct or indirect subclass of top.
- If an abstract object class has superclasses, the superclasses must also be abstract.

Checking an entry against the schema

When an entry is added or modified through an LDAP operation, the entry is checked against the schema. By default, all checks listed in this section are performed. However, you can selectively disable some of them by providing an `ibm-slapdSchemaCheck` value to the `ibmslapd.conf` configuration directive. See the *IBM Tivoli Directory Server Version 6.0 Installation and Configuration Guide* for information about schema configuration attributes.

To comply with the schema an entry is checked for the following conditions:

With respect to object classes:

- Must have at least one value of attribute type "objectClass".
- Can have any number of auxiliary object classes including zero. This is not a check, but a clarification. There are no options to disable this.
- Can have any number of abstract object classes, but only as a result of class inheritance. This means that for every abstract object class that the

entry has, it also has a structural or auxiliary object class that inherits directly or indirectly from that abstract object class.

- Must have at least one structural object class.
- Must have exactly one immediate or base structural object class. This means that of all the structural object classes provided with the entry, they all must be superclasses of exactly one of them. The most derived object class is called the "immediate" or "base structural" object class of the entry, or simply the "structural" object class of the entry.
- Cannot change its immediate structural object class (on `ldap_modify`).
- For each object class provided with the entry, the set of all of its direct and indirect superclasses is calculated; if any of those superclasses is not provided with the entry, then it is automatically added.

The validity of the attribute types for an entry is determined as follows:

- The set of MUST attribute types for the entry is calculated as the union of the sets of MUST attribute types of all of its object classes, including the implied inherited object classes. If the set of MUST attribute types for the entry is not a subset of the set of attribute types contained by the entry, the entry is rejected.
- The set of MAY attribute types for the entry is calculated as the union of the sets of MAY attribute types of all of its object classes, including the implied inherited object classes. If the set of attribute types contained by the entry is not a subset of the union of the sets of MUST and MAY attribute types for the entry, the entry is rejected.
- If any of the attribute types defined for the entry are marked as NO-USER-MODIFICATION, the entry is rejected.

The validity of the attribute type values for an entry is determined as follows:

- For every attribute type contained by the entry, if the attribute type is single-valued and the entry has more than one value, the entry is rejected.
- For every attribute value of every attribute type contained by the entry, if its syntax does not comply with the syntax checking routine for the syntax of that attribute, the entry is rejected.
- For every attribute value of every attribute type contained by the entry, if its length is greater than the maximum length assigned to that attribute type, the entry is rejected.

The validity of the DN is checked as follows:

- The syntax is checked for compliance with the BNF for DistinguishedNames. If it does not comply, the entry is rejected.
- It is verified that the RDN is made up with only attribute types that are valid for that entry.
- It is verified that the values of attribute types used in the RDN appear in the entry.

DEN schema support

The Directory-Enabled Network (DEN) specification defines a standard schema form that stores and describes the relationships among objects that represent users, applications, network elements, and networking services.

To support DEN, the IBM Tivoli Directory Server provides the following features:

- Subclassing (class inheritance). Class definitions can be created from existing definitions through subclassing. The new class definition inherits the properties from the parent class definition. The SUP option in the object class definition is used to specify the parent (or superior) object classes.
- LDAP syntaxes required by DEN, which include the following:
 - Boolean
 - DN
 - Directory String
 - Generalized Time
 - UTC Time
 - IA5 String
 - Integer

iPlanet compatibility

The parser used by the IBM Tivoli Directory Server allows the attribute values of schema attribute types (objectClasses and attributeTypes) to be specified using the grammar of iPlanet. For example, descrs and numeric-oids can be specified with surrounding single quotation marks (as if they were qdescrs). However, the schema information is always made available through ldap_search. As soon as a single dynamic change (using ldap_modify) is performed on an attribute value in a file, the whole file is replaced by one where all attribute values follow the IBM Directory Version 6.0 specifications. Because the parser used on the files and on ldap_modify requests is the same, an ldap_modify that uses the iPlanet grammar for attribute values is also handled correctly.

When a query is made on the subschema entry of a iPlanet server, the resulting entry can have more than one value for a given OID. For example, if a certain attribute type has two names (such as 'cn' and 'commonName'), then the description of that attribute type is provided twice, once for each name. The IBM Tivoli Directory Server can parse a schema where the description of a single attribute type or object class appears multiple times with the same description (except for NAME and DESCR). However, when the IBM Tivoli Directory Server publishes the schema it provides a single description of such an attribute type with all of the names listed (the short name comes first). For example, here is how iPlanet describes the common name attribute:

```
( 2.5.4.3 NAME 'cn'
  DESC 'Standard Attribute'
  SYNTAX '1.3.6.1.4.1.1466.115.121.1.15' )

( 2.5.4.3 NAME 'commonName'
  DESC 'Standard Attribute, alias for cn'
  SYNTAX '1.3.6.1.4.1.1466.115.121.1.15' )
```

This is how the IBM Tivoli Directory Server describes it:

```
( 2.5.4.3 NAME ( 'cn' 'commonName' ) SUP name )
```

The IBM Tivoli Directory Server supports subtypes. If you do not want 'cn' to be a subtype of name (which deviates from the standard), you can declare the following:

```
( 2.5.4.3 NAME ( 'cn' 'commonName' )
  DESC 'Standard Attribute'
  SYNTAX '1.3.6.1.4.1.1466.115.121.1.15' )
```

The first name ('cn') is taken as the preferred or short name and all other names after 'cn' as alternate names. From this point on, the strings '2.5.4.3', 'cn' and 'commonName' (as well as their case-insensitive equivalents) can be used interchangeably within the schema or for entries added to the directory.

Generalized and UTC time

There are different notations used to designate date and time-related information. For example, the fourth day of February in the year 1999 can be written as:

```
2/4/99
4/2/99
99/2/4
4.2.1999
04-FEB-1999
```

as well as many other notations.

IBM Tivoli Directory Server standardizes the timestamp representation by requiring the LDAP servers to support two syntaxes:

- The Generalized Time syntax, which takes the form:

```
YYYYMMDDHHMMSS[. | ,fraction][(+|-HHMM)|Z]
```

There are 4 digits for the year, 2 digits each for the month, day, hour, minute, and second, and an optional fraction of a second. Without any further additions, a date and time is assumed to be in a local time zone. To indicate that a time is measured in Coordinated Universal Time, append a capital letter Z to a time or a local time differential. For example:

```
"19991106210627.3"
```

which in local time is 6 minutes, 27.3 seconds after 9 p.m. on 6 November 1999.

```
"19991106210627.3Z"
```

which is the coordinated universal time.

```
"19991106210627.3-0500"
```

which is local time as in the first example, with a 5 hour difference in relation to the coordinated universal time.

If you designate an optional fraction of a second, a period or a comma is required. For local time differential, a '+' or a '-' must precede the hour-minute value

- The Universal time syntax, which takes the form:

```
YYMMDDHHMM[SS][(+ | -)HHMM)|Z]
```

There are 2 digits each for the year, month, day, hour, minute, and optional second fields. As in GeneralizedTime, an optional time differential can be specified. For example, if local time is a.m. on 2 January 1999 and the coordinated universal time is 12 noon on 2 January 1999, the value of UTCTime is either:

```
"9901021200Z"
```

or

```
"9901020700-0500"
```

If the local time is a.m. on 2 January 2001 and the coordinated universal time is 12 noon on 2 January 2001, the value of UTCTime is either:

```
"0101021200Z"  
  or  
"0101020700-0500"
```

UTCtime allows only 2 digits for the year value, therefore the usage is not recommended.

The supported matching rules are `generalizedTimeMatch` for equality and `generalizedTimeOrderingMatch` for inequality. Substring search is not allowed. For example, the following filters are valid:

```
generalized-timestamp-attribute=199910061030  
utc-timestamp-attribute>=991006  
generalized-timestamp-attribute=*
```

The following filters are not valid:

```
generalized-timestamp-attribute=1999*  
utc-timestamp-attribute>=*1010
```

Chapter 9. Basic server administration tasks

Note: Unless stated otherwise, the following tasks can be performed by the directory administrator, a member of the administrative group, or a member of the local administrative group.

- “Changing an administrator distinguished name and password”
- “Starting and stopping the server” on page 68
- “Checking server status” on page 69
- “Managing server connections” on page 82
- “Managing connection properties” on page 84
- “Creating the administrative group” on page 86
- “Managing unique attributes” on page 91

Changing an administrator distinguished name and password

This task can be performed by the directory administrator only.

The administrator name and password is usually set during the server installation and configuration process. However, you can change an administrator name and an administrator password by using either the Web Administration Tool or the command line. See “Setting the administration password and lockout policy” on page 136 for information about administration password security restrictions.

Using Web Administration:

Click **User properties** in the navigation area of the Web Administration Tool. Two selections are displayed:

Change administrator login

Specify a new Administrator DN in the field and enter the current password. Click **OK** or click **Cancel** to return to the Introduction panel without making any changes.

Note: This selection is available only if you are logged in as the directory administrator. It is not available if you are logged in as a user or an administrative group member.

Change password

To change the password for the currently logged-in DN, type your current password in the **Current password** field. Then type your new password in the **New password** field and type it again in the **Confirm new password** field and click **OK**. Click **Cancel** to return to the Introduction panel without making any changes.

Using the command line:

You can use either the **idsdnpw** command or the **idsxcfg** utility from the command line.

Using the **idsdnpw** command:

```
idsdnpw -u <adminDN> -p <adminPW>
```

To use the **idsxcfg** utility type **idsxcfg** on a command line. When the IBM Tivoli Directory Server Configuration Tool panel is displayed select **Manage administrator DN** to change the administrator’s DN or **Manage administrator password** to change the administrator’s password and follow the directions. See the *IBM Tivoli Directory Server Version 6.0 Installation and Configuration Guide* for additional information on using the **idsxcfg** utility.

See Chapter 3, “Distinguished names (DNs),” on page 11 for more information about distinguished names.

Starting and stopping the server

You can use either of the following methods to start or stop the server.

Using Web Administration:

Note: The administration daemon (**idsdiradm**) for the given directory instance must be running.

The current status of the server, either started, stopped, or started in configuration mode, is indicated by the icons in the upper left-hand corner of the server status area. The current status is also described in the first sentence of the work area, for example:

The Directory Server is currently running

1. If you have not done so already, click **Server Administration** in the Web Administration navigation area and then click **Start/Stop/Restart Server** in the expanded list.
2. The message area displays the current state of the server (stopped, running, or running in configuration only mode). Depending on the state of the server, running or stopped, buttons are enabled for you to change the state of the server.

Table 5. Actions available based on the status of the server

Server status	Buttons available
Stopped	Start, Close
Running	Stop, Restart, Close
Running in configuration only mode	Stop, Restart, Close

- If the server is running, click **Stop** to stop the server or **Restart** to stop and then start the server.
 - If the server is stopped, click **Start** to start the server.
 - Click **Close** to return to the Introduction panel.
3. A message is displayed when the server successfully starts or stops.

If you need to perform server configuration maintenance, select the **Start / Restart in configuration only mode** check box. In this mode only the system administrator can bind to the server. All other connections are refused until the server is restarted with DB2 backends enabled (the **Start / Restart in configuration only mode** check box deselected). See Chapter 5, “Configuration only mode,” on page 19 for additional information.

Note: Configuration maintenance can be done while the server is running.

Using the command line or Windows Services icon:

Use the following commands to start server:

Note: The administration daemon (**idsdiradm**) must be running for the **ibmdirctl**
`ibmdirctl -h mymachine -D myDN -w mypassword -p 3538 start`

or

```
idsslapd -I <instancename>
```

Use the following commands to stop the server:

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

or

```
idsslapd -I <instancename> -k
```

to start and stop the server respectively. See “**idsdirctl, ibmdirctl**” on page 369 and “**idsdiradm, ibmdiradm**” on page 438 for additional information.

For Windows systems use the previous commands or:

1. From the desktop, double-click the **My Computer** icon.
2. Double-click the **Control Panel** icon.
3. Double-click the **Administrative Tools** icon
4. Double-click the **Services** icon.
5. To start the server select **Control Panel ->Administrative Tools->Services**, select **IBM Tivoli Directory Server Instance V6.0 - <instancename>** and click **Start**.
6. To stop the server select **Control Panel ->Administrative Tools->Services**, select **IBM Tivoli Directory Server Instance V6.0 - <instancename>** and click **Stop**.

Note: If you change the time zone on your Windows machine, you need to restart the server and the administration daemon in order for the server and administration daemon to recognize the time change. This ensures that the time stamps in the administration daemon’s logs match the time stamps in the server’s logs.

Checking server status

You can check the status of the server by searching for the object classes under `cn=monitor`. To do this, use one of the following methods:

Using Web Administration:

Expand the Server administration category in the navigation area. Click **View server status**. This panel has nine tabs. At the bottom of this panel you can click **Refresh** to update the status displayed on the tab you are currently viewing or you can click **Close** to return to the IBM Tivoli Directory Server Introduction panel. If the directory server is running, the following information is displayed:

General

Click the **General** tab to display the following information:

Hostname

The host name of the LDAP server.

Server status

The server is either **Running** or **Running configuration only mode**. You can determine the server status at any time by the three icons displayed in the left side corner of the server status area.

Start time

The time the server was started. The start time is in the format:
year-month-day hour:minutes:seconds GMT

Current time

The current time on the server. The current time is in the format:
year-month-day hour:minutes:seconds GMT

Total threads

The number of worker threads being used by the server.

Total threads blocked on write

The number of threads sending data back to the client.

Total threads blocked on read

The number of threads reading data from the client.

Number of connections

The number of currently active connections.

Total connections

The total number of connections since the server was started.

Number of entries sent

The number of entries sent by the server since the server was started.

Percentage of entry cache used

The percentage of entry cache currently used. This value is not displayed in configuration only mode.

Percentage of search filter cache used

The percentage of search filter cache currently used. This value is not displayed in configuration only mode.

ACL cache

A Boolean value indicating that the ACL cache is active (TRUE) or inactive (FALSE). This value is not displayed in configuration only mode.

Maximum ACL cache size

The maximum number of entries allowed in the ACL cache. This value is not displayed in configuration only mode.

Bypass alias dereferencing

The server runtime value that indicates if alias processing can be bypassed. It displays true, if no alias object exists in the directory, and false, if at least one alias object exists in the directory.

Total number of SSL connections

The total number of SSL connections since the server was started. This information displays only if the server you are connected to supports the monitor connection type counts feature.

Total number of TLS connections

The total number of TLS connections since the server was started. This information displays only if the server you are connected to supports the monitor connection type counts feature.

Operation counts

Click **Operation counts** to display the following information:

Number of operations requested

The number of initiated requests since the server was started.

Number of operations completed

The number of completed requests since the server was started.

Number of search operations requested

The number of initiated searches since the server was started.

Number of search operations completed

The number of completed searches since the server was started.

Number of bind operations requested

The number of bind requests since the server was started.

Number of bind operations completed

The number of completed bind requests since the server was started.

Number of unbind operations requested

The number of unbind requests since the server was started.

Number of unbind operations completed

The number of completed unbind requests since the server was started.

Number of add operations requested

The number of add requests since the server was started.

Number of add operations completed

The number of completed add requests since the server was started.

Number of delete operations requested

The number of unbind requests since the server was started.

Number of delete operations completed

The number of completed unbind requests since the server was started.

Number of modify RDN operations requested

The number of modify RDN requests since the server was started.

Number of modify RDN operations completed

The number of completed modify RDN requests since the server was started.

Number of modify operations requested

The number of modify requests since the server was started.

Number of modify operations completed

The number of completed modify requests since the server was started.

Number of compare operations requested

The number of compare requests since the server was started.

Number of compare operations completed

The number of completed compare requests since the server was started.

Number of abandon operations requested

The number of abandon requests since the server was started.

Number of abandon operations completed

The number of completed abandon requests since the server was started.

Number of extended operations requested

The number of extended requests since the server was started.

Number of extended operations completed

The number of completed extended requests since the server was started.

Number of unknown operations requested

The number of unknown requests since the server was started.

Number of unknown operations completed

The number of completed unknown requests since the server was started.

Work queue

Click **Work queue** to display the following:

Number of worker threads available

The number of worker threads available for work.

Depth of the work queue

The current size of the work queue.

Largest size of the work queue

The largest size that the work queue has ever reached.

Number of connections closed by automatic connection cleaner

The number of idle connections closed by the automatic connection cleaner.

Number of times the automatic connection cleaner has run

The number of times the automatic connection cleaner has run.

Emergency thread currently active

The indicator of whether the emergency thread is running. An emergency thread is available when some number of items, configurable on the server, are on the work queue. This provides a method for the administrator to access the server during a denial of service attack.

Number of times the emergency thread has been activated

The number of times the emergency thread has been activated.

Last time the emergency thread was activated

The last time the emergency thread was activated.

View worker status

Click **View worker status** to display information about the worker threads that are currently active. This information is useful when the server is not performing as expected or performing poorly. Performing this search suspends all server activity until it is completed. A warning to that effect is displayed and explains that the time to complete this operation depends on the number of connections and active worker threads. Click **Yes** to display the information.

The following worker thread information is displayed in a table.

Thread ID

The ID of the worker thread, for example, 2640.

Operation

The type of work request received, for example, search.

Bind DN

The DN used to bind to the server.

Client IP

The IP address of the client.

To view a worker thread's details, select the worker thread you want more information about from the View worker status table and click **View**. The following information fields about the selected worker thread are displayed:

Thread ID

The ID of the worker thread, for example, 2640.

Operation

The type of work request receive, for example, search.

LDAP version

The LDAP version level, either V1, V2 or V3.

Bind DN

The DN used to bind to the server.

Client IP

The IP address of the client.

Client port

The port used by the client.

Connection ID

The number that identifies the connection.

Received at

The date and time that the work request was received.

Request parameters

Additional information about the operation. For example, if the request was a search, the following information is also provided:

```
base=cn=workers,cn=monitor
scope=baseObject
dereferaliases=neverDerefAliases
typesonly=false
filter=(objectclass=*)
attributes=all
```

Click **Close** to return to the **View worker status** panel.

Directory cached attributes

Click **Directory cached attributes** to display the following information. The status items are displayed in a table format. You can use the arrows next to each header to specify a sort in either ascending or descending order. You can also either use the **Select Action** drop-down list to select **Edit sort** and click **Go** or click the **Edit sort** icon to specify up to three sort criteria.

Table 6. Directory cached attributes table

Attribute ^	Number of cache hits ^	Cache size ^

Attribute

The name of the attribute.

Number of cache hits

The number of times the cache for this attribute was used to resolve a search filter.

Cache size

The amount of memory used by this attribute cache.

This tab also contains two non-editable fields:

Cached attribute total size (in kilobytes)

The amount of memory being used by the cache.

Note: This number includes additional memory used to manage the caches. Consequently, this total is larger than the sum of the memory used for the individual attribute caches.

Cached attribute configured size

The maximum amount of memory that can be used by attribute caching. See “Adding attributes to and removing attributes from the attribute cache” on page 110 for instructions.

Directory cache candidates

This table is a list of the 10 non-cached attributes that are most frequently used in search filters that can be resolved by the attribute cache manager. If the frequency of the usage of these attributes is excessive, you might want to add them to the attribute cache. You can use the arrows next to each header to specify a sort in either ascending or descending order. You can also either use the **Select Action** drop-down list to select **Edit sort** and click **Go** or click the **Edit sort** icon to specify up to two sort criteria.

Table 7. Directory cache candidates table

Attribute ^	Number of hits ^

Attribute

The name of the attribute.

Number of hits

The number of times the attribute has been used in filters that can be resolved by the attribute cache manager.

Changelog cached attributes

Click **Changelog cached** attributes to display the following information. The status items are displayed in a table format. You can use the arrows next to each header to specify a sort in either ascending or descending order. You can also either use the **Select Action** drop-down list to select **Edit sort** and click **Go** or click the **Edit sort** icon to specify up to three sort criteria.

Table 8. Changelog cached attributes table

Attribute ^	Number of cache hits ^	Cache size ^

Attribute

The name of the attribute.

Number of cache hits

The number of times the cache for this attribute was used to resolve a search filter.

Cache size

The amount of memory used by this attribute cache.

This tab also contains two non-editable fields:

Cached attribute total size (in kilobytes)

The amount of memory being used by the cache.

Note: This number includes additional memory used to manage the caches. Consequently, this total is larger than the sum of the memory used for the individual attribute caches.

Cached attribute configured size

The maximum amount of memory that can be used by attribute caching. See “Adding attributes to and removing attributes from the attribute cache” on page 110 for instructions.

Changelog cache candidates

This table is a list of the 10 non-cached attributes that are most frequently used in search filters that can be resolved by the attribute cache manager. If the frequency of the usage of these attributes is excessive, you might want to add them to the attribute cache. You can use the arrows next to each header to specify a sort in either ascending or descending order. You can also either use the **Select Action** drop-down list to select **Edit sort** and click **Go** or click the **Edit sort** icon to specify up to two sort criteria.

Table 9. Changelog cache candidates table

Attribute ^	Number of hits ^

Attribute

The name of the attribute.

Number of hits

The number of times the attribute has been used in filters that can be resolved by the attribute cache manager.

Trace and logs

Click Trace and logs to view the following information:

Trace enabled

The current trace value for the server. TRUE, if collecting trace data, FALSE, if not collecting trace data. See “idsldaptrace, ldaptrace” on page 415 for information about enabling and starting the trace function.

Trace message level

The current ldap_debug value for the server. The value is in hexadecimal form, for example,

0x0=0
0xffff=65535

For more information, see “Debugging levels” on page 462.

Trace message log

The name of the file that contains the trace output.

Note: If the value is stderr, the output is displayed in the command window where the LDAP server was started. If the server was not started from the command line, no data is displayed.

Number of messages added to server logs

The number of error messages recorded since the server started.

Number of messages added to DB2 error log

The number of DB2 error messages recorded since the server started.

Number of messages added to audit log

The number of messages recorded by the audit log since the server started.

Number of error messages added to audit log

The number of failed operation messages recorded by the audit log.

Using the command line:

To determine server status using the command line use the **idsldapsearch** command for the following bases

- cn=monitor
- cn=workers,cn=monitor
- cn=connections,cn=monitor
- cn=changelog,cn=monitor

cn=monitor

```
idsldapsearch -h <servername> -p <portnumber> -b cn=monitor -s base objectclass=*
```

This command returns the following information:

cn=monitor

version=IBM Tivoli Directory (SSL), Version 6.0

totalconnections

The total number of connections since the server was started.

total_ssl_connections

The total number of SSL connections since the server was started.

total_tls_connections

The total number of TLS connections since the server was started.

currentconnections

The number of active connections.

maxconnections

The maximum number of active connections allowed.

writewaiters

The number of threads sending data back to the client.

readwaiters

The number of threads reading data from the client.

opsinitiated

The number of requests since the server was started.

livethreads

The number of worker threads being used by the server.

opscompleted

The number of completed requests since the server was started.

entriessent

The number of entries sent by the server since the server was started.

searchesrequested

The number of requested searches since the server was started.

searchescompleted

The number of completed searches since the server was started.

bindsrequested

The number of bind operations requested since the server was started.

bindscompleted

The number of bind operations completed since the server was started.

unbindsrequested

The number of unbind operations requested since the server was started.

unbindscompleted

The number of unbind operations completed since the server was started.

addsrequested

The number of add operations requested since the server was started.

addscompleted

The number of add operations completed since the server was started.

deletesrequested

The number of delete operations requested since the server was started.

deletescompleted

The number of delete operations completed since the server was started.

modrdnsrequested

The number of modify RDN operations requested since the server was started.

modrdnscompleted

The number of modify RDN operations completed since the server was started.

modifiesrequested

The number of modify operations requested since the server was started.

modifiescompleted

The number of modify operations completed since the server was started.

comparesrequested

The number of compare operations requested since the server was started.

comparescompleted

The number of compare operations completed since the server was started.

abandonsrequested

The number of abandon operations requested since the server was started.

abandonscompleted

The number of abandon operations completed since the server was started.

extopsrequested

The number of extended operations requested since the server was started.

extopscompleted

The number of extended operations completed since the server was started.

unknownopsrequested

The number of unknown operations requested since the server was started.

unknownopscompleted

The number of unknown operations completed since the server was started.

slapderrorlog_messages

The number of server error messages recorded since the server was started or since a reset was performed.

slapdclierrors_messages

The number of DB2 error messages recorded since the server was started or since a reset was performed.

auditlog_messages

The number of audit messages recorded since the server was started or since a reset was performed.

auditlog_failedop_messages

The number of failed operation messages recorded since the server was started or since a reset was performed.

filter_cache_size

The maximum number of filters allowed in the cache.

filter_cache_current

The number of filters currently in the cache.

filter_cache_hit

The number of filters found in the cache.

filter_cache_miss

The number of search operations that attempted to use the filter cache, but didn't find a matching operation in the cache.

filter_cache_bypass_limit

Search filters that return more entries than this limit are not cached.

entry_cache_size

The maximum number of entries allowed in the cache.

entry_cache_current

The number of entries currently in the cache.

entry_cache_hit

The number of entries found in the cache.

entry_cache_miss

The number of entries not found in the cache.

acl_cache

A Boolean value indicating that the ACL cache is active (TRUE) or inactive (FALSE).

acl_cache_size

The maximum number of entries in the ACL cache.

cached_attribute_total_size

The amount of memory in kilobytes used by attribute caching.

cached_attribute_configured_size

The amount of memory in kilobytes that can be used by attribute caching.

cached_attribute_hit

The number of times the attribute has been used in a filter that could be processed by the changelog attribute cache. The value is reported as follows:

cached_attribute_hit=attrname:#####

cached_attribute_size

The amount of memory used for this attribute in the changelog attribute cache. This value is reported in kilobytes as follows:

cached_attribute_size=attrname:#####

cached_attribute_candidate_hit

A list of up to ten most frequently used noncached attributes that have been used in a filter that could have been processed by the changelog attribute cache if all of the attributes used in the filter had been cached. The value is reported as follows:

cached_attribute_candidate_hit=attrname:#####

You can use this list to help you decide which attributes you want to cache. Typically, you want to put a limited number of attributes into the attribute cache because of memory constraints.

currenttime

The current time on the server. The current time is in the format:

year-month-day hour:minutes:seconds GMT

starttime

The time the server was started. The start time is in the format:

year-month-day hour:minutes:seconds GMT

trace_enabled

The current trace value for the server. TRUE, if collecting trace data, FALSE, if not collecting trace data. See “idsldaptrace, ldaptrace” on page 415 for information about enabling and starting the trace function.

trace_message_level

The current ldap_debug value for the server. The value is in hexadecimal form, for example:

0x0=0
0xffff=65535

For more information, see “Debugging levels” on page 462.

trace_message_log

The current LDAP_DEBUG_FILE environment variable setting for the server.

en_currentregs

The current number of client registrations for event notification.

en_notificationssent

The total number of event notifications sent to clients since the server was started.

bypass_deref_aliases

The server runtime value that indicates if alias processing can be bypassed. It displays true, if no alias object exists in the directory, and false, if at least one alias object exists in the directory.

available_workers

The number of worker threads available for work.

current_workqueue_size

The current depth of the work queue.

largest_workqueue_size

The largest size that the work queue has ever reached.

idle_connections_closed

The number of idle connections closed by the Automatic Connection Cleaner.

auto_connection_cleaner_run

The number of times that the Automatic Connection Cleaner has run.

emergency_thread_running

The indicator of whether the emergency thread is running.

totaltimes_emergency_thread_run

The number of times the emergency thread has been activated.

lasttime_emergency_thread_run

The last time the emergency thread was activated.

cn=workers,cn=monitor

For worker thread information ensure that auditing is enabled and issue the following command:

```
idsldapsearch -D <adminDN> -w <adminpw> -b cn=workers,cn=monitor
-s base objectclass=*
```

This command gives the following type of information for each active worker:

cn=workers,cn=monitor**cn=workers****objectclass=container****cn=thread2640,cn=workers,cn=monitor**

thread The number of the worker thread. For example 2640.

ldapversion

The LDAP version level, either V1 or V2.

binddn

The DN used to bind to the server.

clientip

The IP address of the client.

clientport

The port used by the client.

connectionid

The number identifying the connection.

received

The date and time that the work request was received.

workrequest

The type of work request received and additional information about the request. For example, if the request was a search, the following information is also provided:

```
base=cn=workers,cn=monitor
scope=baseObject
dereferaliases=neverDerefAliases
typesonly=false
filter=(objectclass=*)
attributes=all
```

cn=connections,cn=monitor

```
idsldapsearch -D <adminDN> -w <adminpw> -h <servername> -p <portname> -b  
cn=connections,cn=monitor -s base objectclass=*
```

This search returns something similar to the following:

```
cn=connections,cn=monitor  
connection=3546 : 9.48.181.83 : 2005-02-28 21:53:54 GMT : 1 : 5 : CN=ROOT : :  
connection=3550 : 9.48.181.83 : 2005-02-28 21:53:54 GMT : 1 : 3 : CN=ROOT : :  
connection=3551 : 9.48.181.83 : 2005-02-28 21:53:55 GMT : 1 : 4 : CN=ROOT : :  
connection=3553 : 9.48.181.83 : 2005-02-28 21:53:55 GMT : 1 : 3 : CN=ROOT : :  
connection=3554 : 9.48.181.83 : 2005-02-28 21:53:55 GMT : 1 : 5 : CN=ROOT : :  
connection=3555 : 9.48.181.83 : 2005-02-28 21:53:55 GMT : 1 : 2 : CN=ROOT : :  
connection=3556 : 9.48.181.83 : 2005-02-28 21:53:55 GMT : 1 : 2 : CN=ROOT : :  
connection=3557 : 9.48.181.83 : 2005-02-28 21:53:55 GMT : 1 : 1 : CN=ROOT : :  
connection=3558 : 9.48.181.83 : 2005-02-28 21:53:55 GMT : 1 : 1 : CN=ROOT : :  
connection=3559 : 9.48.181.83 : 2005-02-28 21:53:55 GMT : 0 : 1 : CN=ROOT : :
```

connection=xxxx

The connection number.

9.48.181.83

The server IP address.

2005-02-28 21:53:54 GMT

The current time on the server. The current time is in the format:
year-month-day hour:minutes:seconds GMT

1 : 5 The opsInProgress and opsCompleted, respectively.

- opsInProgress – The number of requests in progress.
- opsCompleted – The number of completed requests since the server was started.

CN=ROOT

This is the Administrator DN.

cn=changelog,cn=monitor

```
idsldapsearch -D <adminDN> -w <adminpw> -h <servername> -p <portname> -b  
cn=changelog,cn=monitor -s base objectclass=*
```

This search returns something similar to the following:

```
CN=CHANGELOG,CN=MONITOR  
cached_attribute_total_size=0  
cached_attribute_configured_size=0
```

cached_attribute_total_size

The amount of memory used by the changelog attribute cache, in kilobytes. This number includes additional memory used to manage the cache that is not charged to the individual attribute caches. Consequently, this total is larger than the sum of the memory used by all the individual attribute caches.

cached_attribute_configured_size

The maximum amount of memory, in kilobytes, that is enabled to be used by the changelog attribute cache

cached_attribute_hit

The number of times the attribute has been used in a filter that could be processed by the changelog attribute cache. The value is reported as follows:

```
cached_attribute_hit=attrname:####
```

cached_attribute_size

The amount of memory used for this attribute in the changelog attribute cache. This value is reported in kilobytes as follows:

```
cached_attribute_size=attrname:#####
```

cached_attribute_candidate_hit

A list of up to ten most frequently used noncached attributes that have been used in a filter that could have been processed by the changelog attribute cache if all of the attributes used in the filter had been cached. The value is reported as follows:

```
cached_attribute_candidate_hit=attrname:#####
```

You can use this list to help you decide which attributes you want to cache. Typically, you want to put a limited number of attributes into the attribute cache because of memory constraints.

Managing server connections

You can use one of the following methods to check the connection status of the server.

Using Web Administration:

Expand the Server administration category in the navigation area. Click **Manage server connections**. A table containing the following information for each connection is displayed. You can use the arrows next to each header to specify a sort in either ascending or descending order. You can also either use the **Select Action** drop-down list to select **Edit sort** and click **Go** or click the **Edit sort** icon to specify up to three sort criteria.

DN Specifies the DNs of a client connection to the server.

IP address

Specifies the IP address of the client that has a to the server.

Start time

Specifies the date and time when the connection was made.

Status Specifies whether the connection is active or idle. A connection is considered active if it has any operations in progress.

Ops pending

Specifies the number of operations pending since the connection was established.

Ops completed

Specifies the number of operations that have been completed for each connection.

Type Specifies whether the connection is secured by SSL or TLS. Otherwise the field is blank.

Notes:

1. This table displays up to 20 connections at a time.

You can specify to have this table displayed by either DN or IP address by expanding the drop-down menu at the top of the panel and making a selection. The default selection is by DN. Similarly you can also specify whether to display the table in ascending or descending order.

Click **Refresh** or select **Refresh** from the **Select Action** drop-down list and click **Go** to update the current connection information.

If you are logged on as the administrator or as a member of the administration group, you have additional selections to disconnect server connections available on the panel. This ability to disconnect server connections enables you to stop denial of service attacks and to control server access. You can disconnect a connection by expanding the drop-down menus and selecting a DN, an IP address or both and clicking **Disconnect**. Depending on your selections the following actions occur:

Table 10. Disconnection rules

DN chosen	IP address chosen	Action
<DNvalue>	None	All connections bound with the specified DN are disconnected.
None	<IPvalue>	All connections over the specified IP address are disconnected.
<DNvalue>	<IPvalue>	All connections bound as the specified DN and over the specified IP address are disconnected.
None	None	This is not a valid condition. You must specify a DN or an IP address or both to use the disconnect function.

The default value for each of the drop-down menus is **None**.

To disconnect all server connections except for the one making this request click **Disconnect all**. A confirmation warning is displayed. Click **OK** to proceed with the disconnect action or click **Cancel** to end the action and return to the **Manage server connections** panel.

Using the command line:

To view server connections, issue the command:

```
idsldapsearch -D <adminDN> -w <adminPW> -h <servername> -p <portnumber>
-b cn=connections,cn=monitor -s base objectclass=*
```

This command returns information in the following format:

```
cn=connections,cn=monitor
connection=1632 : 9.41.21.31 : 2002-10-05 19:18:21 GMT : 1 : 1 : CN=ADMIN : :
connection=1487 : 127.0.0.1 : 2002-10-05 19:17:01 GMT : 1 : 1 : CN=ADMIN : :
```

Note: If appropriate, an SSL or a TLS indicator is added on each connection.

To end server connections issue, one of the following commands:

```
# To disconnect a specific DN:
idsldapexop -D <adminDN> -w <adminPW> -op unbind -dn cn=john
```

```
# To disconnect a specific IP address:
idsldapexop -D <adminDN> -w <adminPW> -op unbind -ip 9.182.173.43
```

```
#To disconnect a specific DN over a specific IP address:
idsldapexop -D <adminDN> -w <adminPW> -op unbind -dn cn=john -ip 9.182.173.43
```

```
#To disconnect all connections:
idsldapexop -D <adminDN> -w <adminPW> -op unbind -all
```

See “idsldapexop, ldapexop” on page 386 for more information on ending connections.

Managing connection properties

The ability to manage connection properties enables you to prevent clients from locking up the server by closing connections of clients that:

- Send data slowly, send partial data or send no data.
- Do not read data results or read results slowly.
- Do not unbind.
- Bind anonymously.

It also ensures that an administrator always has access to the server in the cases that the backend is kept busy with long running tasks.

Using Web Administration:

These selections are displayed only if you are logged in as the administrator or a member of the administration group on a server that supports this feature.

Expand the Server administration category in the navigation area. Click **Manage connection properties**.

Note: The actual maximum threshold numbers are limited by the number of files permitted per process. On UNIX or Linux[®] systems you can use the **ulimit -a** command to determine the limits. On Windows systems this is a fixed number.

1. Select the General tab.
2. The **Allow anonymous connections** check box is already selected for you so that anonymous binds are allowed. This is the default setting. You can click the check box to deselect the **Allow anonymous connections** feature. This action causes the server to unbind all anonymous connections.

Note: Disallowing anonymous binds might cause some applications to fail.

3. Set the threshold number to initiate the cleanup of anonymous connections. You can specify a number between 0 and 65535 in the **Cleanup threshold for anonymous connections** field. The default setting is 0. When this number of anonymous connections is exceeded, connections are cleaned up based on the idle timeout limit that you set in the **Idle time out** field.
4. Set the threshold number to initiate the cleanup of authenticated connections. You can specify a number between 0 and 65535 in the **Cleanup threshold for authenticated connections** field. The default setting is 1100. When this number of authenticated connections is exceeded, connections are cleaned up based on the idle timeout limit that you set in the **Idle time out** field.
5. Set the threshold number to initiate the cleanup of all connections. You can specify a number between 0 and 65535 in the **Cleanup threshold for all connections** field. The default setting is 1200. When this total number of connections is exceeded, connections are cleaned up based on the idle timeout limit that you set in the **Idle time out** field.
6. Set the number of seconds that a connection can be idle before it is closed by a cleanup process. You can specify a number between 0 and 65535 in the **Idle**

timeout limit field. The default setting is 300. When a cleanup process is initiated, any connections, subject to the process, that exceed the limit are closed.

7. Set the number of seconds between write attempts that will be allowed. You can specify a number between 0 and 65535 in the **Result timeout limit** field. The default setting is 120. Any connections that exceed this limit are ended.

Note: This applies to Windows systems only. A connection that exceeds 30 seconds is automatically dropped by the operating system. Therefore this **Result timeout limit** setting is overridden by the operating system after 30 seconds.

8. Select the **Emergency thread** tab.
9. The **Enable emergency thread** check box is already selected for you so that the emergency thread can be activated. This is the default setting. An emergency thread becomes available when either the work queue, that is the number of pending operations or the time limit threshold since the last item was removed from the work queue is exceeded. This thread provides a method for the administrator to access the server during a denial of service attack. You can click the check box to deselect the **Enable emergency thread** feature. This action prevents the emergency thread from being activated.
10. Set the number limit for work requests that activate the emergency thread. Specify a number between 0 and 65535 in the **Pending request threshold** field to set the limit of work requests that can be in the queue before activating the emergency thread. The default is 50. When the specified limit is exceeded, the emergency thread is activated.
11. Set the number of minutes that can elapse since the last work item was removed from the queue. If there are work items in the queue and this time limit is exceeded, the emergency thread is activated. You can specify a number between 0 and 240 in the **Time threshold** field. The default setting is 5.
12. Select from the drop-down menu, the criterion to be used to activate the emergency thread. You can select:
 - **Size only** - The emergency thread is activated only when the queue exceeds the specified amount of pending work items.
 - **Time only** - The emergency thread is activated only when the time limit between removed work items exceeds the specified amount.
 - **Size or time** - The emergency thread is activated when either the queue size or time threshold exceeds the specified amounts.
 - **Size and Time** - The emergency thread is activated when both the queue size and the time threshold exceed the specified amounts.Size and Time is the default setting.
13. When you are finished, click **Apply** to save your changes without exiting, or click **OK** to apply your changes and exit, or click **Cancel** to exit this panel without making any changes.

Using the command line:

To perform the same operations using the command line, issue the following command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where <filename> contains:

```

dn: cn=Connection Management,cn=Front End, cn=Configuration
changetype: modify
replace: ibm-slapdAllowAnon
ibm-slapdAllowAnon: TRUE
-
replace: ibm-slapdAnonReapingThreshold
ibm-slapdAnonReapingThreshold: 0
-
replace: ibm-slapdBoundReapingThreshold
ibm-slapdBoundReapingThreshold: 1100
-
replace: ibm-slapdAllReapingThreshold
ibm-slapdAllReapingThreshold: 1200
-
replace: ibm-slapdIdleTimeOut
ibm-slapdIdleTimeOut: 300
-
replace: ibm-slapdWriteTimeout
ibm-slapdWriteTimeout: 120
-
replace: ibm-slapdEThreadEnabl
ibm-slapdEThreadEnable: TRUE
-
replace: ibm-slapdESizeThreshold
ibm-slapdESizeThreshold: 50
-
replace: ibm-slapdETimeThreshold
ibm-slapdETimeThreshold: 5
-
#ibm-slapdEThreadActivate can be set to S for size only, T for
#time only, SOT for size or time, and SAT for size and time.
replace: ibm-slapdEThreadActivate
ibm-slapdEThreadActivate: { S | T | SOT | SAT}

```

To update the settings dynamically, issue the following **idsldapexop** command:

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

The **idsldapexop** command updates only those attributes that are dynamic. For other changes to take effect you must stop and restart the server. See “Dynamically-changed attributes” on page 578 for a list of the attributes that can be updated dynamically.

Creating the administrative group

An administrative group provides the ability to provide 24 hour administrative capabilities without having to share a single ID and password among the administrators. Members of the administrative group have their own unique IDs and passwords. The administrative group member DNs must not match each other and they must also not match the IBM Tivoli Directory Server administrator’s DN. Conversely, the IBM Tivoli Directory Server administrator DN must not match the DN of any administrative group member. This rule also applies to the Kerberos or Digest-MD5 IDs of the IBM Tivoli Directory Server administrator and the administrative group members. These DNs must not match any of the IBM Tivoli Directory Server’s replication supplier DNs. This also means that IBM Tivoli Directory Server’s replication supplier DNs must not match any of the administrative group member DNs or the IBM Tivoli Directory Server administrator DN.

Note: The IBM Tivoli Directory Server’s replication supplier DNs can match each other.

The members of the administrative group have all the capabilities of the directory administrator with the following exceptions:

- Only the IBM Tivoli Directory Server administrator has the ability to add or remove members from the administrative group. In addition only the IBM Tivoli Directory Server administrator can modify the DN, password, Kerberos ID, or Digest-MD5 ID of any administrative group member. However, a member of the administrative group can modify his own password, but cannot modify his own DN, Kerberos ID, or Digest-MD5 ID. An administrative group member cannot see the password of any other administrative group member or the IBM Tivoli Directory Server administrator.
- Only the IBM Tivoli Directory Server administrator has the ability to add or remove the `cn=Kerberos,cn=Configuration` and the `cn=Digest,cn=Configuration` entries in the configuration backend. Administrative group members can modify all the attributes in these entries except for the directory administrator's Kerberos ID and Digest-MD5 ID.
- Only the IBM Tivoli Directory Server administrator has the ability to modify or update any of the audit log settings. Members of the administrative group are able only to view the audit log and the audit log settings.
- Only the IBM Tivoli Directory Server administrator has the ability to clear the audit log.

Note: See “Global administration group” on page 252 for information on how administrative rights are delegated for the database backend in a distributed directory environment.

Enabling and disabling the administrative group

You must be the IBM Tivoli Directory Server administrator to perform this operation.

Note: In this task and the Manage administrative group tasks that follow, the operation buttons are disabled for members of the administrative group. Members of the administrative group can only view the **Administrative group members** table on the **Manage administrative group** panel.

Using Web Administration:

Expand the Server administration category in the navigation area. Click **Manage administrative group**.

1. To enable or disable the administrative group, click the check box next to **Enable administrative group**. If the box is checked, the administrative group is enabled.
2. Click **OK**.

Note: If you disable the administrative group, any member who is logged in can continue administrative operations until that member is required to rebind. To stop any additional operations by already bound administrative group members, perform an unbind operation. See “Managing server connections” on page 82 for more information.

Using the command line:

To perform the same operations using the command line, issue the following command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where <filename> contains:

```
dn: cn=Configuration
changetype: modify
replace: ibm-slapdAdminGroupEnabled
#specify TRUE to enable or FALSE to disable the administrative group
#TRUE has been preselected for you.
ibm-slapdAdminGroupEnabled: TRUE
objectclass: top
objectclass: ibm-slapdConfigEntry
objectclass: ibm-slapdTop
```

To update the settings dynamically, issue the following **idsldapexop** command:

```
idsldapexop -D <adminDN> -w <adminPW> -op readconfig -scope single
cn=Configuration ibm-slapdAdminGroupEnabled
```

Adding members to the administrative group

You must be the IBM Tivoli Directory Server administrator to perform this operation.

Using Web Administration:

To add a member to the administrative group, on the **Manage administrative group panel**, click **Add**.

On the **Add administrative group member** panel:

1. Enter the member's administrator DN (this must be a valid DN syntax).
2. Enter the member's password. See "Setting the administration password and lockout policy" on page 136 for information about administration password security restrictions.
3. Enter the member's password again to confirm it.
4. Optionally, enter the member's **Kerberos ID**. The Kerberos ID must be in either `ibm-kn` or `ibm-KerberosName` format. The values are case insensitive, for example, `ibm-kn=root@TEST.AUSTIN.IBM.COM` is equivalent to `ibm-kn=ROOT@TEST.AUSTIN.IBM.COM`.

Note: This field is only available for the AIX® and Windows platforms. It is displayed only, if the kerberos supported capabilities OID (1.3.18.0.2.32.30) is found on the server.

5. Optionally, enter the member's **Digest-MD5 user name**.
6. Click **OK**.

Note: The Digest-MD5 user name is case sensitive.

Repeat this procedure for each member you want to add to the administrative group.

The member administrator DN, Digest-MD5 username, if specified, and Kerberos ID, if specified, are displayed in the **Administrative group members** list box.

Note: Kerberos support is only available for the AIX and Windows platforms. The Kerberos ID column in the is displayed in the **Administrative group members** list box only, if the kerberos supported capabilities OID (1.3.18.0.2.32.30) is found on the server.

Using the command line:

To perform the same operations using the command line, issue the following command:

```
idsldapadd -D <adminDN> -w<adminPW> -i<filename>
```

where *<filename>* contains:

```
dn: cn=AdminGroup, cn=Configuration
cn: AdminGroup
objectclass: top
objectclass: container
```

```
dn: cn=admin1, cn=AdminGroup, cn=Configuration
cn: admin1
ibm-slapdAdminDN: <memberDN>
ibm-slapdAdminPW: <password>
#ibm-slapdKrbAdminDN and ibm-slapdDigestAdminUser are optional attributes.
ibm-slapdKrbAdminDN: <KerberosID>
ibm-slapdDigestAdminUser: <DigestID>
objectclass: top
objectclass: ibm-slapdConfigEntry
objectclass: ibm-slapdAdminGroupMember
```

Note: If you already have a member created in the administrative group, omit the first entry.

To update the settings dynamically, issue the following **idsldapexop** command:

```
idsldapexop -D <adminDN> -w <adminPW> -op readconfig -scope subtree
cn=AdminGroup,cn=Configuration
```

Modifying an administrative group member

You must be the IBM Tivoli Directory Server administrator to perform this operation.

Using Web Administration:

To modify an administrative group member's information, on the Manage administrative group panel:

1. Select the member whose information you want to modify.
2. Click **Edit**.
3. Change the member's administrator DN (this must be a valid DN syntax).
4. Change the member's password.
5. Enter the member's password again to confirm it.
6. Enter or change the member's **Kerberos ID**. The Kerberos ID must be in either **ibm-kn** or **ibm-KerberosName** format. The values are case insensitive, for example, **ibm-kn=root@TEST.AUSTIN.IBM.COM** is equivalent to **ibm-kn=ROOT@TEST.AUSTIN.IBM.COM**.

Note: This field is only available for the AIX and Windows platforms. It is displayed only, if the kerberos supported capabilities **OID(1.3.18.0.2.32.30)** is found on the server.

7. Enter or change the member's **Digest-MD5 user name** . The Digest-MD5 user name is case sensitive.
8. Click **OK**.

Repeat this procedure for each member you want to modify in the administrative group.

Note: If you are member of the administrative group, you can change your password using the **User properties->Change password** panel.

Using the command line:

To perform the same operations using the command line, issue the following command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where <filename> contains:

```
dn: cn=admin1, cn=AdminGroup, cn=Configuration
cn: admin1
changetype: modify
replace: ibm-slapdAdminDN
ibm-slapdAdminDN: cn=<memberDN>
-
replace: ibm-slapdAdminPW
ibm-slapdAdminPW: <password>
-
replace: ibm-slapdKrbAdminDN
ibm-slapdKrbAdminDN: <KerberosID>
-
replace: ibm-slapdDigestAdminUser
ibm-slapdDigestAdminUser: <DigestID>
```

To update the settings dynamically, issue the following **idsldapexop** command:

```
idsldapexop -D <adminDN> -w <adminPW> -op readconfig -scope subtree
cn=AdminGroup,cn=Configuration
```

Removing a member from the administrative group

You must be the IBM Tivoli Directory Server administrator to perform this operation.

Using Server Administration:

To remove a member of the administrative group, on the Manage administrative group panel:

1. Select the member you want to remove.
2. Click **Delete**.
3. You are prompted to confirm the removal.
4. Click **OK** to delete the member or **Cancel** to return to the Manage administrative group panel without making any changes.

Repeat this procedure for each member you want to remove from the administrative group.

Using the command line:

To perform the same operations using the command line, issue the following command:

```
idsldapdelete -D <adminDN> -w<adminPW> -i<filename>
```

where <filename> contains:

```
#list additional DNs here, one per line:
cn=admin1, cn=AdminGroup, cn=Configuration
```

To remove multiple members, list the DNs. Each DN must be on a separate line.

To update the settings dynamically, issue the following **idsldapexop** command:

```
idsldapexop -D <adminDN> -w <adminPW> -op readconfig -scope subtree
cn=AdminGroup,cn=Configuration
```

Managing unique attributes

The Unique Attributes feature ensures that specified attributes always have unique values within a directory. These attributes can be specified in two entries only, `cn=uniqueattribute,cn=localhost` and `cn=uniqueattribute,cn=IBMpolicies`. The values for a unique attribute are stored on the server where the attribute has been designated as unique. Search results for unique attributes are unique for that server's database only. Search results that include results from referrals might not be unique.

Note: Binary attributes, operational attributes, configuration attributes, and the `objectclass` attribute cannot be designated as unique.

Creating unique attributes

Note: On a per attribute basis, language tags are mutually exclusive with unique attributes. If you designate a particular attribute as being a unique attribute, it cannot have language tags associated with it.

Using Web Administration:

Expand the Server administration category in the navigation area. Click **Manage unique attributes**.

1. Select the attribute that you want to add as a unique attribute from the **Available attributes** menu. The available attributes listed are those that can be designated as unique. For example, `sn`.

Note: An attribute remains in the list of available attributes until it has been placed in both the `cn=localhost` and the `cn=IBMpolicies` containers.

2. Click either **Add to `cn=localhost`** or **Add to `cn=IBMpolicies`**. The difference between these two containers is that `cn=IBMpolicies` entries are replicated and `cn=localhost` entries are not. The attribute is displayed in the appropriate list box. You can list the same attribute in both containers.

Note: If an entry is created under both `cn=localhost` and `cn=IBMpolicies`, the resultant union of these two entries is the consolidation of their unique attributes list. For example, if the attributes `cn` and `employeeNumber` are designated as unique in `cn=localhost` and the attributes `cn` and `telephoneNumber` are designated as unique on `cn=IBMpolicies`, the server treats the attributes `cn`, `employeeNumber`, and `telephoneNumber` as unique attributes.

3. Repeat this process for each attribute you want to add to the attribute cache.
4. Click **OK** to save your changes or click **Cancel** to exit this panel without making any changes.

Using the command line:

To designate that an attribute must have unique values, issue the following command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where `<filename>` contains:

```
dn: cn=uniqueattributes,cn=localhost
changetype: add
ibm-UniqueAttributeTypes: sn
objectclass: top
objectclass: ibm-UniqueAttributeTypes
```

To add additional attributes, issue the command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where <filename> contains:

```
dn: cn=uniqueattributes,cn=localhost
cn: uniqueattributes
changetype: modify
add: ibm-UniqueAttributeTypes
ibm-UniqueAttributeTypes: AIXAdminUserId
-
add: ibm-UniqueAttributeTypes
ibm-UniqueAttributeTypes: adminGroupNames
```

When adding or modifying a unique attribute entry, if establishing a unique constraint for any of the listed unique attribute types results in errors, the entry is not added or created in the directory. The problem must be resolved and the command to add or modify must be reissued before the entry can be created or modified. For example, while adding a unique attribute entry to the directory, if establishing a unique constraint on a table for one of the listed unique attribute types failed (that is, because of having duplicate values in the database), a unique attribute entry is not added to the directory. An error DSA is unwilling to perform is issued.

Note: If an entry is created under both `cn=localhost` and `cn=IBMpolicies`, the resultant union of these two entries is the consolidation of their unique attributes list. For example, if the attributes `cn` and `employeeNumber` are designated as unique in `cn=localhost` and the attributes `cn` and `telephoneNumber` are designated as unique on `cn=IBMpolicies`, the server treats the attributes `cn`, `employeeNumber`, and `telephoneNumber` as unique attributes.

When an application tries to add an entry to the directory with a value for the attribute that duplicates an existing directory entry, an error with result code 20 (LDAP: error code 20 - Attribute or Value Exists) from the LDAP server is issued.

When the server starts, it checks the list of unique attributes and determines if the DB2 constraints exist for each of them. If the constraint does not exist for an attribute because it was removed by the `idsbulkload` utility or because it was removed manually by the user, it is removed from the unique attributes list and an error message is logged in the error log, `ibmslapd.log`. For example, if the attribute `cn` is designated as unique in `cn=uniqueattributes,cn=localhost` and there is no DB2 constraint for it the following message is logged:

```
Values for the attribute CN are not unique.
The attribute CN was removed from the unique attribute
entry: CN=UNIQUEATTRIBUTES,CN=LOCALHOST
```

Removing an attribute from the list of unique attributes

To remove an attribute from the list of unique attributes, use either of the following methods.

Note: If a unique attribute exists in both `cn=uniqueattribute,cn=localhost` and `cn=uniqueattribute,cn=IBMpolicies` and it is removed from only one entry, the server continues to treat that attribute as a unique attribute. The attribute become nonunique when it has been removed from both entries.

Using Web Administration:

Expand the Server administration category in the navigation area. Click **Manage unique attributes**.

1. Select the attribute that you want to remove from the unique attributes list by clicking the attribute in the appropriate list box. For example AIXAdminUserId from the previous task.
2. Click **Remove**.
3. Repeat this process for each attribute you want to remove from the list.
4. Click **OK** to save your changes or click **Cancel** to exit this panel without making any changes.

Note: If you remove the last unique attribute from the cn=localhost or the cn=IBMpolicies list boxes, the container entry for that list box, cn=uniqueattribute,cn=localhost or cn=uniqueattribute,cn=IBMpolicies is automatically deleted.

Using the command line:

To remove an attribute from the list of unique attributes using the command line, issue the following command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where <filename> contains:

```
dn: cn=uniqueattributes,cn=localhost
changetype: modify
cn: uniqueattributes
ibm-UniqueAttributeTypes: AIXAdminUserId
```

To remove all of the unique attributes stored in, for example, cn=localhost issue the command:

```
idsldapdelete -D <adminDN> -w <Adminpw> "cn=uniqueattributes,cn=localhost"
```

By deleting "cn=uniqueattributes" entry from the directory, the unique constraints enforced on the unique attributes are dropped to allow nonunique values for the attributes again.

Chapter 10. Setting server properties

You can set the following properties for your server:

- “Changing server ports and enabling language tags”
- “Setting Searches” on page 99
- “Enabling and disabling transaction support” on page 106
- “Enabling and disabling event notification” on page 104
- “Adding and removing suffixes” on page 108
- Chapter 12, “Referrals,” on page 153
- “Adding attributes to and removing attributes from the attribute cache” on page 110

While the Web Administration Tool is the preferred method, updates to the server configuration file can be made using LDAP utilities. The LDAP modify requests can be generated by:

- A C-application using the C-client provided with the IBM Tivoli Directory Server
- A Java application using JNDI
- Any other interface that generates a standard V3 LDAP.

Examples that are provided use the `idsldapmodify` command line utility.

The `idsldapmodify` command can be run either in interactive mode or with input specified in a file. For most examples in this guide, the file contents to be used with the `idsldapmodify` command are supplied. The general form of the command to use with these files is:

```
idsldapmodify -D <adminDN> -w <password> -i <filename>
```

To update the server configuration settings dynamically, you need to issue the following `idsldapexop` commands. This command updates all configuration settings that are dynamic:

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

This command updates a single setting.

```
idsldapexop -D <adminDN> -w<adminPW> -op readconfig -scope single <entry DN>  
                  <attribute>
```

The `idsldapexop` command updates only those attributes that are dynamic. For other changes to take effect you must stop and restart the server. See “Dynamically-changed attributes” on page 578 for a list of the attributes that can be updated dynamically. See Chapter 22, “Command line utilities,” on page 367 for more information about the `idsldapmodify` and `idsldapexop` commands.

Note: Only the administrator and members of the administrative group are allowed to update the server configuration settings.

Changing server ports and enabling language tags

Note: Remember, if you change the port setting for the server, you must also change the port settings for the server in the console. See “Modifying a server in the console” on page 28.

Using Web Administration:

Click the **Manage server properties** tab in the Web Administration navigation area to display the Manage server properties panel. This panel is displayed with the **General** tab preselected. The General panel has two read-only information fields, which display the host name of the server and the version level of the IBM Tivoli Directory Server that is installed on the machine.

This panel also has three modifiable required fields, **Unsecure port** (default value is 389), **Secure port** (default value 636) that display the respective current port numbers and a check box to enable and disable language tag support.

Note: The well-known ports are those from 0 through 1023. The registered ports are those from 1024 through 49151. The dynamic or private ports are those from 49152 through 65535.

If you want to change the port settings or enable language tags or both:

1. Click **Unsecure port** and enter a number ranging from 1 through 65535. For this example 399. Remember, if you change the port setting for the server, you must also change the port settings for the server in the console. See “Modifying a server in the console” on page 28.
2. Click **Secure port** and enter a number ranging from 1 through 65535. For this example 699. Remember, if you change the port setting for the server, you must also change the port settings for the server in the console. See “Modifying a server in the console” on page 28.
3. Click the **Enable language tag support** check box to enable support for language tags. The default setting is disabled. See “Language tags” on page 297 for more information.

Note: After enabling the language tag feature, if you associate language tags with the attributes of an entry, the server returns the entry with the language tags. This occurs even if you later disable the language tag feature. Because the behavior of the server might not be what the application is expecting, to avoid potential problems, do not disable the language tag feature after it has been enabled.

4. When you are finished, click **Apply** to save your changes without exiting, or click **OK** to apply your changes and exit, or click **Cancel** to exit this panel without making any changes.

If you have changed a port number, you must stop the server for changes to take effect. See “Starting and stopping the server” on page 68.

Note: You can enable or disable language tags dynamically, without restarting the server.

After stopping the server you must also stop and start the administration daemon locally to resynchronize the ports. See Chapter 4, “Directory administration daemon,” on page 17. Restart the server.

Using the command line:

To determine whether the language tag feature is enabled, issue a root DSE search specifying the attribute **ibm-enabledCapabilities**.

```
idsldapsearch -b "" -s base objectclass=* ibm-enabledCapabilities
```

If the OID **1.3.6.1.4.1.4203.1.5.4** is returned, the feature is enabled.

If the language tag support is not enabled, any LDAP operation that associates a language tag with an attribute is rejected with the error message:

```
LDAP_NO_SUCH_ATTRIBUTE
```

To assign the ports that are not the default ports and to enable language tags using the command line, issue the following command:

```
idsldapmodify -D <adminDN> -w <password> -i <filename>
```

where <filename> contains:

```
dn: cn=configuration
changetype: modify
replace: ibm-slapdPort
ibm-slapdPort: 399
```

```
-
replace: ibm-slapdSecurePort
ibm-slapdSecurePort: 699
```

```
dn: cn=Directory, cn=RDBM Backends, cn=IBM Directory, cn=Schemas, cn=Configuration
replace: ibm-slapdLanguageTagsEnabled
ibm-slapdLanguageTagsEnabled: TRUE
```

You must stop the server for changes to take effect. See “Starting and stopping the server” on page 68.

Note: You can enable or disable language tags dynamically, without restarting the server.

After stopping the server you must also stop and start the administration daemon locally to resynchronize the ports. See Chapter 4, “Directory administration daemon,” on page 17.

```
ibmdirctl -D <AdminDN> -w <Adminpw> -p 389 stop
```

```
ibmdirctl -D<AdminDN> -w <Adminpw> admstop
```

```
idsdiradm
```

```
ibmdirctl -D<AdminDN> -w <Adminpw> start
```

Setting Performance

Note: For the latest tuning information, see the *IBM Tivoli Directory Server Version 6.0 Performance Tuning Guide* located on the Tivoli Software Library Web site. See “Accessing publications online” on page x for information about accessing online publications.

You can change the search limits and connections settings to enhance performance.

Using Web Administration:

Expand the **Manage server properties** category in the navigation area of the Web Administration Tool, select the **Performance** tab.

1. Specify the **Number of database connections**. This sets the number of DB2 connections used by the server. The minimum number you must specify is 5. The default setting is 15. If your LDAP server receives a high volume of client requests or clients are receiving “connection refused” errors, you might see better results by increasing the setting of the number of connections made to DB2 by the server. The maximum number of connections is determined by the setting on your DB2 database. While there are no longer any server limitations upon the number of connections you specify, each connection does consume

resources. Consult the *IBM Tivoli Directory Server Version 6.0 Performance Tuning Guide* for the latest tuning recommendations for your system.

2. Specify the **Number of database connections for replication**. This sets the number of DB2 connections used by the server for replication. The minimum number you must specify is 1. The default setting is 4. Consult the *IBM Tivoli Directory Server Version 6.0 Performance Tuning Guide* for the latest tuning recommendations for your system.

Note: The total number of connections specified for database connections and database connections for replication cannot exceed the number of connections set in your DB2 database.

3. Select **Cache ACL information** to use the following ACL cache settings. This option must be selected in order for the other cache setting options on this panel to take effect.
4. Specify the **Maximum number of elements in ACL cache**. The default is 25,000.
5. Specify the **Maximum number of elements in entry cache**. The default is 25,000.
6. Specify the **Maximum number of elements in search filter cache**. The default is 25,000. The search filter cache consists of actual queries on the requested attribute filters and resulting entry identifiers that matched. On an update operation, all filter cache entries are invalidated.
7. Specify the **Maximum number of elements from a single search added to search filter cache**. If you select **Elements**, you must enter a number. The default is 100. Otherwise, select **Unlimited**. Search entries that match more entries than the number specified here are not added to the search filter cache.
8. When you are finished, click **Apply** to save your changes without exiting, or click **OK** to apply your changes and exit, or click **Cancel** to exit this panel without making any changes.
9. If you are setting the number of database connections, you must restart the server for the changes to take effect. If you were modifying only the settings, the server does not need to be restarted.

Using the command line:

To perform the same operations using the command line, issue the following command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where <filename> contains:

```
dn: cn=Directory,cn=RDBM Backends,cn=IBM Directory,cn=Schemas,cn=Configuration
changetype: modify
replace: ibm-slapdDbConnections
ibm-slapdDbConnections: 15
-
replace: ibm-slapdReplDbConns
ibm-slapdReplDbConns: 4

dn: cn=Front End, cn=Configuration
changetype: modify
replace: ibm-slapdACLCache
ibm-slapdACLCache: TRUE
-
replace: ibm-slapdACLCacheSize
ibm-slapdACLCacheSize: 25000
-
replace: ibm-slapdEntryCacheSize
```



```
ibm-slapdEntryCacheSize: 25000
-
replace: ibm-slapdFilterCacheSize
ibm-slapdFilterCacheSize: 25000
-
replace: ibm-slapdFilterCacheBypassLimit
ibm-slapdFilterCacheBypassLimit: 100
```

To update the settings dynamically, issue the following **idsldapexop** command:
`idsldapexop -D <adminDN> -w<adminPW> -op readconfig -scope entire`

The **idsldapexop** command updates only those attributes that are dynamic. For other changes to take effect you must stop and restart the server. See “Dynamically-changed attributes” on page 578 for a list of the attributes that can be updated dynamically.

Setting Searches

You can set search parameters to control users’ search capabilities, such as paged and sorted searching.

Paged results allow you to manage the amount of data returned from a search request. You can request a subset of entries (a page) instead of receiving all the results at once. Subsequent search requests display the next page of results until the operation is canceled or the last result is returned. Sorted search allows a client to receive search results sorted by a list of criterion, where each criteria represents a sort key. This selection moves the responsibility of sorting from the client application to the server, where it might be done more efficiently.

A directory entry with objectclass of ‘alias’ or ‘aliasObject’ contains an attribute ‘aliasedObjectName’ that is used to reference another entry in the directory. Only search requests can specify if aliases are dereferenced. Dereferencing means to trace the alias back to the original entry. The IBM Tivoli Directory Server response time for searches with the alias dereferencing option set to **always** or **search** might be significantly longer than that of searches with dereferencing option set to **never**, if alias entries exist in the directory.

The server side dereference option can be set to **never**, **find**, **search**, or **always**. This option value is combined with the dereference option value specified in a search request by a logical AND operation. The resulting value is used as the dereference option in the search operation.

Using Web Administration:

Expand the **Manage server properties** category in the navigation area of the Web Administration Tool, select the **Search settings** tab.

1. Set the **Search size limit**. Click either the **Entries** or the **Unlimited** radio button. If you select **Entries**, you need to specify in the field the maximum number of entries a search returns. The default setting is 500. If more entries fit the search criteria, they are not returned. This limit does not apply to the administrator or administrative group members.
2. Set the **Search time limit**. Click either the **Seconds** or the **Unlimited** radio button. If you select **Entries**, you need to specify in the field the maximum amount of time the server spends processing the request. The default setting is 900. This limit does not apply to the administrator administrative group members.

3. To restrict search sorting capabilities to administrators, select the **Only allow administrators to sort searches** check box.
4. To restrict search paging capabilities to administrators, select the **Only allow administrators to page searches** check box.
5. To set the level of alias dereferencing, expand the drop-down menu for **Alias dereferencing** and select one of the following. The default setting is **always**.
 - never** Aliases are never dereferenced
 - find** Aliases are dereferenced when finding the starting point for the search, but not when searching under that starting entry.
 - search** Aliases are dereferenced when searching the entries beneath the starting point of the search, but not when finding the starting entry.
 - always** Aliases are always dereferenced, both when finding the starting point for the search, and also when searching the entries beneath the starting entry. Always is the default setting.

Note: This option is available only if your server supports dereferencing aliases.

6. Specify in seconds the time to wait (idle time out) for paged searches. Paged searches require an open connection between the LDAP server and the DB2 database where the LDAP data is stored. The idle time out administrative limit is designed to age out DB2 database connections held open for paged results search requests.
7. Specify the maximum number of concurrent paged searches allowed by the server at any given time. The default setting is 3.

Note: Setting the value to 0 disables paged searches.
8. Specify the maximum number of attributes used for sorting in sorted searches. The default setting is 3.

Note: Setting the value to 0 disables sorted searches.
9. When you are finished, click **Apply** to save your changes without exiting, or click **OK** to apply your changes and exit, or click **Cancel** to exit this panel without making any changes.

See “Searching the directory with paging and sorting” on page 101 for additional information about searches.

Using the command line:

To perform the same operations using the command line, issue the following command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where <filename> contains:

```
dn: cn=Configuration
changetype: modify
replace: ibm-slapdTimeLimit
ibm-slapdTimeLimit: 900
-
replace : ibm-slapdDerefAliases
ibm-slapdDerefAliases: {never|find|search|always}
-
replace: ibm-slapdSizeLimit
ibm-slapdSizeLimit: 500
```

```

dn: cn=Directory,cn=RDBM Backends,cn=IBM Directory,cn=Schemas,cn=Configuration
changetype: modify
replace: ibm-slapdPagedResAllowNonAdmin
ibm-slapdPagedResAllowNonAdmin: false
-
replace: ibm-slapdPagedResLmt
ibm-slapdPagedResLmt: 3
-
replace: ibm-slapdSortKeyLimit
ibm-slapdSortKeyLimit: 3
-
replace: ibm-slapdSortSrchAllowNonAdmin
ibm-slapdSortSrchAllowNonAdmin: false

dn: cn=Front End, cn=Configuration
changetype: modify
replace: ibm-slapdIdleTimeOut
ibm-slapdIdleTimeOut: 300

```

To update the settings dynamically, issue the following **idsldapexop** command:
idsldapexop -D <adminDN> -w<adminPW> -op readconfig -scope entire

See “idsldapsearch, ldapsearch” on page 406 for information on how to perform searches using the command line.

Searching the directory with paging and sorting

The search function searches for a filter match on only the first 240 bytes of an attribute if indexing is enabled for that attribute. Additionally, if sort is specified on a search request, the server sorts the entries found by the search using only the first 240 bytes. Any end user or client application needs to take into consideration that a match for a search filter that exists in a value after the first 240 bytes might not be returned to the client depending on whether indexing is enabled for that attribute.

Note: This restriction is specific to the IBM Tivoli Directory Server. IBM LDAP servers on other platforms, including z/OS™ and i5/OS might have different restrictions. Consult the documentation for each platform to determine restrictions.

The administrator can tell if indexing has been enabled for an attribute by looking at the attribute definition in the Web Administration Tool, (**Schema management -> Manage attributes -> <attributename> -> Edit ->IBM extensions**) or by looking at the attribute definition returned by a search of cn=schema. When viewing an attribute definition in the Web Administration Tool, the IBM extensions tab displays the following:

Indexing rules

- Equality
- Ordering
- Approximate
- Substring
- Reverse

The appropriate indexing rules are checked for the attribute. If the **idsldapsearch** utility is used, the **ibmattributetypes** value contains the keywords: APPROX, EQUALITY, ORDERING, SUBSTR or REVERSE. For example, the ‘cn’ attribute has the following indexes defined:

```
attributetypes=( 2.5.4.3 NAME ( 'cn' 'commonName' ) DESC 'This is the X.500
commonName attribute, which contains a name of an object.
If the object corresponds to a person, it is typically the
persons full name.' SUP 2.5.4.41 EQUALITY 2.5.13.2
ORDERING 2.5.13.3 SUBSTR 2.5.13.4 )
ibmattributetypes=( 2.5.4.3 DBNAME ( 'cn' 'cn' ) ACCESS-CLASS NORMAL LENGTH
256 EQUALITY ORDERING SUBSTR APPROX )
```

See “Indexing rules” on page 42.

Sorted search control

Sorted Search Results provides sort capabilities for LDAP clients that have limited or no sort functionality. Sorted Search Results allows an LDAP client to receive search results sorted based on a list of criteria, where each criteria represents a sort key. The sort criteria includes attribute types, matching rules, and descending order. The server uses this criteria to sort search results before returning them. This moves the responsibility of sorting from the client application to the server, where it might be done much more efficiently. For example, a client application might want to sort the list of employees at the company’s Grand Cayman site by surname, common name, and telephone number. Instead of building the search list twice so it can be sorted (once at the server and then again at the client after all the results are returned), the search list is built only once, and then sorted, before returning the results to the client application.

The server sorts search entries based on attributes and by default allows a maximum of three sort keys (attribute names) per search operation. To change the value of this administrative limit, change the following line, `ibm-slapdSortKeyLimit: 3`, in the `ibmslapd.conf` file. See “Setting Searches” on page 99 for information on how to do this. If the line does not exist, add it to set the new maximum (if the line does not exist, the server is using the default value).

By default the server honors requests from nonadministrator binds, including those binding anonymously. Because sorting search results before returning them uses more server resources than simply returning them, you might want to configure the server to honor only requests from users binding with administrator authority. To honor sorted search requests submitted using only administrator bind, change the following line, `ibm-slapdSortSrchAllowNonAdmin: true` to `ibm-slapdSortSrchAllowNonAdmin: false`, in the `ibmslapd.conf` file. See “Setting Searches” on page 99. If the line does not exist, add it with a value of false to enable only administrator binds to perform sorted search operations.

The LDAP server returns all referrals to the client at the end of a search request. It is up to the application using the client services to decide whether to set the criticality of the sorted search request, and to handle a lack of support of those controls on referral servers as appropriate based on the application. Additionally, the LDAP server does not ensure that the referral server supports the sorted search control. Multiple lists could be returned to the client application, some not sorted. It is the client application’s decision as to how to best present this information to the end user. Possible solutions include: combine all referral results before presenting to the end user; show multiple lists and the corresponding referral server host name; take no extra steps and show all results to the end user as they are returned from the server. The client application must turn off referrals to get one truly sorted list, otherwise when chasing referrals with sorted search controls specified, unpredictable results might occur.

It is important to note when taking advantage of the server sorted search results that:

- The server takes advantage of the underlying DB2 database to perform sorting of search results. This means that there might be different sorted search results based on the data code page for the database (especially if your database code page is UTF-8).
- Ordering rules specified for a sort key attribute are ignored by the server. At this time, ordering rules are not supported by the server.
- There is no support for multi-server sorting (referrals). The server cannot guarantee that referred servers support sorted search results.

More information about the server side sorted search control can be found in RFC 2891. The control OID for sorted search results is 1.2.840.113556.1.4.473, and is included in the Root DSE information as a supported control.

Simple paged results

Simple Paged Results provides paging capabilities for LDAP clients that want to receive just a subset of search results (a page) instead of the entire list. The next page of entries is returned to the client application for each subsequent paged results search request submitted by the client until the operation is canceled or the last result is returned. The server ignores a simple paged results request if the page size is greater than or equal to the `sizeLimit` value for the server because the request can be satisfied in a single operation.

Because paging of search results holds server resources throughout the life of the simple paged results request, there are several new administrative limits employed to ensure that server resources cannot be abused, or misused, through the use of simple paged results search requests.

ibm-slapedPagedResAllowNonAdmin

By default, the server honors requests from non-administrator binds, including those binding anonymously. If you want the server to honor simple paged results search requests only from users binding with administrator authority, you need to change the following line, `ibm-slapedPagedResAllowNonAdmin: true` to `ibm-slapedPagedResAllowNonAdmin: false`, in the `ibmslapd.conf` file. See “Setting Searches” on page 99. If the line does not exist, add it with a value of `false` to allow only Administrator bind.

ibm-slapedPagedResLmt

By default, the server allows a maximum of three outstanding simple paged results operations at any given time. To ensure the fastest response for subsequent simple paged results request, the server holds a database connection open throughout the life of the search request until the user cancels the simple paged results request, or the last result is returned to the client application. This administrative limit is designed to ensure that other operations being handled by the server are not denied service because all database connections are in use by outstanding simple paged results search requests. For consistent results, set the **ibm-slapedPagedResLmt** value lower than the maximum number of database connections for your server. To change the value of this administrative limit, change the following line, `ibm-slapedPagedResLmt: 3`, in the `ibmslapd.conf` file. See “Setting Searches” on page 99. If the line does not exist add it to set the new maximum (if the line does not exist, the server is using the default value).

ibm-slapedIdleTimeOut

The idle time out administrative limit is designed to age out DB2 database connections held open for simple paged results search requests. The default

idle time for a simple paged results request is 500 seconds. For example, if a client application were to pause for 510 seconds between pages, the server would age out the request in order to free the database connection for use by other server operations. The server returns the appropriate error to the client application for the next simple paged results request submitted, at which point the client application needs to restart the simple paged results request. The idle timer for each simple paged results request is restarted after every page returned to the client application. The server checks for aged out simple paged results request every 5 seconds, so if you set the value of `ibm-slapdIdleTimeOut` value lower than 5 seconds, you still have to wait 5 seconds for the simple paged results requests to be aged out. To change the value of this administrative limit, change the following line, `ibm-slapdIdleTimeOut: 300`, in the `ibmslapd.conf` file. See “Setting Searches” on page 99. If the line does not exist, add it to set the new maximum (if the line does not exist, the server is using the default value).

The LDAP server returns all referrals to the client at the end of a search request, the same as a search without any controls. That means that if the server has 10 pages of results returned, all the referrals are returned on the 10th page, not at the end of each page. When chasing referrals, the client application needs to send in an initial paged results request, with the cookie set to null, to each of the referral servers. It is up to the application using the client services to decide whether or not to set the criticality as to the support of paged results, and to handle a lack of support of this control on referral servers as appropriate based on the application. Additionally, the LDAP server does not ensure that the referral server supports paged results controls. Multiple lists could be returned to the client application, some not paged. It is at the client application’s decision as to how to best present this information to the end user. Possible solutions include: combine all referral results before presenting to the end user; show multiple lists and the corresponding referral server host name; take no extra steps and show all results to the end user as they are returned from the server. The client application must turn off referrals to get one truly paged list, otherwise when chasing referrals with the paged results search control specified, unpredictable results might occur.

More information about the server side simple paged results control can be found in RFC 2686. The control OID for simple paged results is 1.2.840.113556.1.4.319, and is included in the Root DSE information as a supported control.

Enabling and disabling event notification

The event notification function allows a server to notify a registered client that an entry in the directory tree has been changed, added, or deleted. This notification is in the form of an unsolicited message.

When an event occurs, the server sends a message to the client as an LDAP v3 unsolicited notification. The messageID is 0 and the message is in the form of an extended operation response. The responseName field is set to the registration OID. The response field contains the unique registration ID and a timestamp for when the event occurred. The time field is in UTC time format.

When a transaction occurs, the event notifications for the transaction steps cannot be sent until the entire transaction is completed.

Note: ACLs are only checked on the entry that the event is registered on, when the event is registered. A user who does not have access to some of the

entries below his access entry might receive notification of changes for those entries. The user is not told the exact change, just that a change has occurred. Also, if the ACLs are changed on the original entry to not allow the user access, the registered events remain, even though the user no longer has access. For these reasons event notification must be disabled for an EAL4 secure configuration. See Chapter 17, "Access control lists," on page 309 for information about ACLs.

Enabling event notification

To enable event notification, use one of the following procedures.

Note: For an EAL4 secure configuration, event notification must be disabled.

Using Web Administration:

Expand the **Manage server properties** category in the navigation area of the Web Administration Tool, select the **Event notification** tab.

1. Select the **Enable event notification** check box to enable event notification. If **Enable event notification** is disabled, the server ignores all other options on this panel.
2. Set the **Maximum registrations per connection**. Click either the **Registrations** or the **Unlimited** radio button. If you select **Registrations**, you need to specify in the field the maximum number of registrations allowed for each connection. The maximum number of registrations is 2,147,483,647. The default setting is 100 registrations.
3. Set the **Maximum registrations total**. This selection sets how many registrations the server can have at any one time. Click either the **Registrations** or the **Unlimited** radio button. If you select **Registrations**, you need to specify in the field the maximum number of registrations allowed for each connection. The maximum number of registrations is 2,147,483,647. The default number of registrations is **Unlimited**.
4. When you are finished, click **Apply** to save your changes without exiting, or click **OK** to apply your changes and exit, or click **Cancel** to exit this panel without making any changes.
5. If you have enabled event notification, you must restart the server for the changes to take effect. If you were modifying only the settings, the server does not need to be restarted.

Using the command line:

To perform the same operations using the command line, issue the following command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where <filename> contains:

```
dn: cn=Event Notification,cn=Configuration
changetype: modify
replace: ibm-slapdEnableEventNotification
ibm-slapdEnableEventNotification: TRUE
-
replace: ibm-slapdMaxEventsPerConnection
ibm-slapdMaxEventsPerConnection: 100
-
replace: ibm-slapdMaxEventsTotal
ibm-slapdMaxEventsTotal: 0
```

If you have enabled event notification, you must restart the server for the changes to take effect. If you were modifying only the settings, the server does not need to be restarted.

To update the settings dynamically, issue the following **idsldapexop** command:
`idsldapexop -D <adminDN> -w<adminPW> -op readconfig -scope entire`

The **idsldapexop** command updates only those attributes that are dynamic. For other changes to take effect you must stop and restart the server. See “Dynamically-changed attributes” on page 578 for a list of the attributes that can be updated dynamically.

Disabling event notification

To disable event notification, use one of the following procedures.

Using Web Administration:

Expand the **Manage server properties** category in the navigation area of the Web Administration Tool, select the **Event notification** tab.

1. Deselect the **Enable event notification** check box to enable transaction processing.
2. When you are finished, click **Apply** to save your changes without exiting, or click **OK** to apply your changes and exit, or click **Cancel** to exit this panel without making any changes.
3. You must restart the server for the changes to take effect.

Using the command line:

To perform the same operations using the command line, issue the following command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where *<filename>* contains:

```
dn: cn=Event Notification,cn=Configuration
changetype: modify
replace: ibm-slapdEnableEventNotification
ibm-slapdEnableEventNotification: FALSE
```

You must restart the server for the changes to take effect.

See the *IBM Tivoli Directory Server Version C-Client SDK Programming Reference* for more information about event notification.

Enabling and disabling transaction support

Transaction processing enables an application to group a set of entry updates together in one operation. Normally each individual LDAP operation is treated as a separate transaction with the database. Grouping operations together is useful when one operation is dependent on another operation because if one of the operations fails, the entire transaction fails. Transaction settings determine the limits on the transaction activity allowed on the server.

Enabling transaction support

To enable transaction support use one of the following procedures.

Using Web Administration:

Expand the **Manage server properties** category in the navigation area of the Web Administration Tool, select the **Transactions** tab.

1. Select the **Enable transaction processing** check box to enable transaction processing. If **Enable transaction processing** is disabled, all other options on this panel, such as **Maximum number of operations per transaction** and **Pending time limit**, are ignored by the server.
2. Set the **Maximum number of transactions**. Click either the **Transactions** or the **Unlimited** radio button. If you select **Transactions**, you need to specify in the field the maximum number of transactions. The maximum number of transactions is 2,147,483,647. The default setting is 20 transactions.
3. Set the **Maximum number of operations per transaction**. Click either the **Operations** or the **Unlimited** radio button. If you select **Operations**, you need to specify in the field the maximum number of operations allowed for each transaction. The maximum number of operations is 2,147,483,647. The smaller the number, the better the performance. The default is 5 operations.
4. Set the **Pending time limit**. This selection sets the maximum timeout value of a pending transaction in seconds. Click either the **Seconds** or the **Unlimited** radio button. If you select **Seconds**, you need to specify in the field the maximum number of seconds allowed for each transaction. The maximum number of seconds is 2,147,483,647. Transactions left uncompleted for longer than this time are cancelled (rolled back). The default is 300 seconds.
5. When you are finished, click **Apply** to save your changes without exiting, or click **OK** to apply your changes and exit, or click **Cancel** to exit this panel without making any changes.
6. If you have enabled transaction support, you must restart the server for the changes to take effect. If you were modifying only the settings, the server does not need to be restarted.

Using the command line:

To perform the same operations using the command line, issue the following command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where <filename> contains:

```
dn: cn=Transaction,cn=Configuration
changetype: modify
replace: ibm-slapdTransactionEnable
ibm-slapdTransactionEnable: TRUE
-
replace: ibm-slapdMaxNumOfTransactions
ibm-slapdMaxNumOfTransactions: 20
-
replace: ibm-slapdMaxOpPerTransaction
ibm-slapdMaxOpPerTransaction: 5
-
replace: ibm-slapdMaxTimeLimitOfTransactions
ibm-slapdMaxTimeLimitOfTransactions: 300
```

To update the settings dynamically, issue the following **idsldapexop** command:

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

The **idsldapexop** command updates only those attributes that are dynamic. For other changes to take effect you must stop and restart the server. See “Dynamically-changed attributes” on page 578 for a list of the attributes that can be updated dynamically.

Disabling transaction support

To disable transaction processing, use one of the following procedures.

Using Web Administration:

Expand the **Manage server properties** category in the navigation area of the Web Administration Tool, select the **Transactions** tab.

1. Deselect the **Enable transaction processing** check box to enable transaction processing.
2. When you are finished, click **Apply** to save your changes without exiting, or click **OK** to apply your changes and exit, or click **Cancel** to exit this panel without making any changes.
3. You must restart the server for the changes to take effect.

Using the command line:

To perform the same operations using the command line, issue the following command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where <filename> contains:

```
dn: cn=Transaction,cn=Configuration
changetype: modify
replace: ibm-slapdTransactionEnable
ibm-slapdTransactionEnable: False
```

You must restart the server for the changes to take effect.

See the *IBM Tivoli Directory Server Version C-Client SDK Programming Reference* for more information about transaction support.

Adding and removing suffixes

A suffix is a DN that identifies the top entry in a locally held directory hierarchy. Because of the relative naming scheme used in LDAP, this DN is also the suffix of every other entry within that directory hierarchy. A directory server can have multiple suffixes, each identifying a locally held directory hierarchy, for example, o=ibm,c=us.

Note: The specific entry that matches the suffix must be added to the directory.

Entries to be added to the directory must have a suffix that matches the DN value, such as 'ou=Marketing,o=ibm,c=us'. If a query contains a suffix that does not match any suffix configured for the local database, the query is referred to the LDAP server that is identified by the default referral. If no LDAP default referral is specified, the result returned indicates that the object does not exist.

Creating or adding suffixes

To create or add a suffix, use one of the following methods.

Using Web Administration:

Note: Defined suffixes such as cn=localhost, cn=pwdpolicy, cn=schema and cn=ibmpolicies cannot be added or removed. Consequently, they are not displayed in the panel.

Expand the **Manage server properties** category in the navigation area of the Web Administration Tool, select the **Suffixes** tab.

1. Enter the Suffix DN, for example, **c=Italy**. The maximum is 1000 characters for a suffix.
2. Click **Add**.
3. Repeat this process for as many suffixes as you want to add.
4. When you are finished, click **Apply** to save your changes without exiting, or click **OK** to apply your changes and exit, or click **Cancel** to exit this panel without making any changes.

Using the command line:

To add suffixes using the command line, issue the following command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where *<filename>* contains:

```
DN: cn=Directory, cn=RDBM Backends, cn=IBM Directory, cn=Schemas, cn=Configuration
changetype: modify
add: ibm-slapdSuffix
ibm-slapdSuffix: <suffixname>
ibm-slapdSuffix: <suffix2>
ibm-slapdSuffix: <suffix3>
```

To update the settings dynamically, issue the following **idsldapexop** command:

```
idsldapexop -D <adminDN> -w<adminPW> -op readconfig -scope single "cn=Directory,
cn=RDBM Backends,cn=IBM Directory,cn=Schemas,cn=Configuration" ibm-slapdSuffix
```

You can also use the **idscfgsuf** command to add suffixes one at a time:

```
idscfgsuf -I <instancename> -s <suffixname>
```

See “idscfgsuf” on page 433.

Removing a suffix

To remove a suffix use, one of the following methods.

Using Web Administration:

Note: Defined suffixes such as **cn=localhost**, **cn=pwdpolicy**, **cn=schema** and **cn=ibmpolicies** cannot be added or removed. Consequently, they are not displayed in the panel.

Expand the **Manage server properties** category in the navigation area of the Web Administration Tool, select the **Suffixes** tab.

1. From the **Current suffix DNs** list box, select the suffixes you want to remove.
2. Click **Remove**.
3. When you are finished, click **Apply** to save your changes without exiting, or click **OK** to apply your changes and exit, or click **Cancel** to exit this panel without making any changes.

Using the command line:

Note: The removal of system defined suffixes such as **cn=localhost**, **cn=pwdpolicy**, **cn=schema** and **cn=ibmpolicies** is not supported.

To perform the same operations using the command line, issue the following command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where <filename> contains:

```
DN: cn=Directory, cn=RDBM Backends, cn=IBM Directory, cn=Schemas, cn=Configuration
changetype: modify
delete: ibm-slapdSuffix
ibm-slapdSuffix: <suffixname>
ibm-slapdSuffix: <suffix2>
ibm-slapdSuffix: <suffix3>
```

You must restart the server for the change to take effect.

To update the settings dynamically, issue the following **idsldapexop** command:

```
idsldapexop -D <adminDN> -w<adminPW> -op readconfig -scope single "cn=Directory,
cn=RDBM Backends,cn=IBM Directory,cn=Schemas,cn=Configuration" ibm-slapdSuffix
```

You can also use the **idsucfgsuf** command to add suffixes one at a time:

```
idsucfgsuf -I <instancename> -s <suffixname>
```

See “idsucfgsuf” on page 459.

Note: You can also use the configuration utility, **idsxcfg**, to add and remove suffixes. See the *IBM Tivoli Directory Server Version 6.0 Installation and Configuration Guide* for more information.

Adding attributes to and removing attributes from the attribute cache

The attribute cache has the advantage of being able to resolve filters in memory rather than in the database. It also has the advantage of not being flushed like the filter cache each time an LDAP add, delete, modify, or modrdn operation is performed.

In deciding which attributes you want to store in memory, you need to consider:

- The amount of memory available to the server
- The size of the directory
- The types of search filters the application typically uses

Typically you want to put a limited number of attributes into the attribute cache because of memory constraints. To help determine which attributes you want to cache, view the Directory cache candidate list and Changelog cache candidate list for the 10 most frequently used attribute search filters by your applications. See “Checking server status” on page 69. Also, see “Determining which attributes to cache” in the *IBM Tivoli Directory Server Version 6.0 Performance Tuning Guide* for more information.

Setting up and adding attributes to the attribute cache

To set up and add attributes to the attribute cache, use one of the following methods.

Using Web Administration:

Expand the **Manage server properties** category in the navigation area of the Web Administration Tool, select the **Attribute cache** tab.

1. You can change the amount of memory in kilobytes available to the directory cache. The default is 16384 kilobytes (16 MB).

2. You can change the amount of memory in kilobytes available to the changelog cache. The default is 16384 kilobytes (16 MB).

Note: This selection is disabled if a changelog has not been configured.

3. Select the attribute that you want to cache from the **Available attributes** menu. Only those attributes that can be designated as cached attributes are displayed in this menu. For example, sn.

Note: An attribute remains in the list of available attributes until it has been placed in both the cn=directory and the cn=changelog containers.

4. Click either **Add to cn=directory** or **Add to cn=changelog**. The attribute is displayed in the appropriate list box. You can list the same attribute in both containers.

Notes:

- a. **Add to cn=changelog** is disabled if a changelog has not been configured.
 - b. Typically, there is no benefit from configuring attribute caching for the changelog database unless you perform very frequent searches of the changelog.
5. Repeat this process for each attribute you want to cache.

Note: The attribute is removed from the drop-down list when the attribute is added to both cn=directory and cn=changelog. If cn=changelog is not enabled, then the **Add to cn=changelog** button is disabled and the entry cannot be added to cn=changelog. The attribute is removed from the available attributes list when it is added to cn=directory.

6. When you are finished, click **Apply** to save your changes without exiting, or click **OK** to apply your changes and exit, or click **Cancel** to exit this panel without making any changes.

Using the command line:

To create the directory attribute caches with the same attributes, issue the following command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where <filename> contains:

```
dn: cn=Directory, cn=RDBM Backends, cn=IBM Directory, cn=Schemas, cn=Configuration
changetype: modify
add: ibm-slapdCachedAttribute
ibm-slapdCachedAttribute: sn
-
add: ibm-slapdCachedAttribute
ibm-slapdCachedAttribute: cn
-
replace: ibm-slapdcachedattributesize
ibm-slapdcachedattributesize: 16384
```

Removing attributes from the attribute cache

To remove an attribute from the attribute cache, perform either of the following tasks.

Using Web Administration

1. Select the attribute that you want to remove from the attributes cache by clicking the attribute in the appropriate list box. For example AIXAdminGroupId from the previous task.

2. Click **Remove**.
3. Repeat this process for each attribute you want to remove from the list.
4. When you are finished, click **Apply** to save your changes without exiting, or click **OK** to apply your changes and exit, or click **Cancel** to exit this panel without making any changes.

Using the command line:

To perform the same operations using the command line, issue the following command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where <filename> contains:

```
DN: cn=Directory, cn=RDBM Backends, cn=IBM Directory, cn=Schemas, cn=Configuration
changetype: modify
delete: ibm-slapdCachedAttribute
ibm-slapdCachedAttribute: sn
```

Chapter 11. Securing the directory

This section describes the steps necessary for keeping the data in your directory secure.

Configuring security settings

The IBM Tivoli Directory Server has the ability to protect LDAP access by encrypting data with either Secure Sockets Layer (SSL) security or Transaction Layer Security (TLS) or both. When using SSL or TLS to secure LDAP communications with the IBM Directory, both server authentication and client authentication are supported. See “Secure Sockets Layer” on page 115 and “Transaction Layer Security” on page 115 for more information.

Note: To use SSL or TLS you must have GSKit installed on your system. Before you can use SSL or TLS you must first use GSKit to create the key database file and certificates. See “Using gsk7ikm” on page 121.

Using Web Administration:

Do the following:

1. Go to the Web Administration console.
2. Click **Server administration**.
3. Click **Manage security properties**.
4. Click **Settings**.
5. Enable the type of security connections, select one of the following radio buttons:

None Enables the server to receive only unsecure communications from the client. The default port is 389.

SSL Enables the server to receive either secure (default port 636) or unsecure (default port 389) communications from the client. The default port is 636.

SSL only

Enables the server to receive only secure communications from the client. This is the most secure way to configure your server. The default port is 636.

TLS Enables the server to receive secure and unsecure communications from the client over the default port, 389. For secure communications the client must start the TLS extended operation. See “Transaction Layer Security” on page 115 for more information.

SSL and TLS

Enables the server to receive secure and unsecure communications from the client over the default port, 389. For secure communications on the default port, the client must start the TLS extended operation. The server also receives secure communications over the SSL port, 636. See “Transaction Layer Security” on page 115 for more information.

Notes:

- a. The TLS and the SSL and TLS options are only available if your server supports TLS.

- b. TLS and SSL do not interoperate. Sending a start TLS request over the secure port results in an operations error.
6. Select the authentication method.

Note: You must distribute the server certificate to each client. For server and client authentication you also must add the certificate for each client to the server's key database.

Select the radio button for either:

Server authentication

For server authentication the IBM Tivoli Directory Server supplies the client with the IBM Tivoli Directory Server's X.509 certificate during the initial SSL handshake. If the client validates the server's certificate, then a secure, encrypted communication channel is established between the IBM Tivoli Directory Server and the client application.

For server authentication to work, the IBM Tivoli Directory Server must have a private key and associated server certificate in the server's key database file.

Server and client authentication

This type of authentication provides for two-way authentication between the LDAP client and the LDAP server. With client authentication, the LDAP client must have a digital certificate (based on the X.509 standard). This digital certificate is used to authenticate the LDAP client to the IBM Tivoli Directory Server. See "Client authentication" on page 120.

7. When you are finished, click **Apply** to save your changes without exiting, or click **OK** to apply your changes and exit, or click **Cancel** to exit this panel without making any changes.

8. You must stop and restart both the IBM Tivoli Directory Server and the administration daemon for the changes to take effect.

- a. Stop the server. See "Starting and stopping the server" on page 68, if you need information about performing this task.

- b. Stop the administration daemon using one of the following methods.

- Remotely, issue the command:

```
ibmdirctl -D <adminDN> -w <adminPW> admstop
```

- Locally issue the command:

```
idsdiradm <instancename> -k
```

See "Stopping an instance of the directory administration daemon" on page 17, if you need information about performing this task.

- c. Start the administration daemon. This must be done locally.

- Issue the command:

```
idsdiradm <instancename>
```

See "Starting an instance of the directory administration daemon" on page 17, if you need information about performing this task.

- d. Start the server. See "Starting and stopping the server" on page 68, if you need information about performing this task.

Using the command line:

To use the command line to configure SSL communications, issue the command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```



```
where <filename> contains:
dn: cn=SSL,cn=Configuration
changetype: modify
replace: ibm-slapdSslAuth
ibm-slapdSslAuth: {serverAuth | serverClientAuth}
-
replace: ibm-slapdSecurity
ibm-slapdSecurity: {none | SSL | SS10nly | TLS | SSLTLS}
```

You must restart the server and the administration daemon for the changes to take effect.

Transaction Layer Security

Transport Layer Security (TLS) is a protocol that ensures privacy and data integrity in communications between the client and server.

TLS is composed of two layers:

The TLS Record Protocol

Provides connection security with data encryption methods such as the Data Encryption Standard (DES) or RC4 without encryption. The keys for this symmetric encryption are generated uniquely for each connection and are based on a secret negotiated by the TLS Handshake Protocol. The Record Protocol can also be used without encryption.

The TLS Handshake Protocol

Enables the server and client to authenticate each other and to negotiate an encryption algorithm and cryptographic keys before data is exchanged.

TLS is invoked by using the -Y option from the client utilities.

Note: TLS and SSL are not interoperable. Issuing a start TLS request (the -Y option) over an SSL port causes an operations error.

Secure Sockets Layer

The IBM Tivoli Directory Server has the ability to protect LDAP access by encrypting data with Secure Sockets Layer (SSL) security. When using SSL to secure LDAP communications with the IBM Directory, both server authentication and client authentication are supported.

With server authentication, the IBM Tivoli Directory Server must have a digital certificate (based on the X.509 standard). This digital certificate is used to authenticate the IBM Tivoli Directory Server to the client application (such as the Directory Management Tool or **idsldapsearch**) or an application built from the application development package, for LDAP access over SSL.

For server authentication the IBM Tivoli Directory Server supplies the client with the IBM Tivoli Directory Server's X.509 certificate during the initial SSL handshake. If the client validates the server's certificate, then a secure, encrypted communication channel is established between the IBM Tivoli Directory Server and the client application.

For server authentication to work, the IBM Tivoli Directory Server must have a private key and associated server certificate in the server's key database file.

Client authentication provides for two-way authentication between the LDAP client and the LDAP server.

With client authentication, the LDAP client must have a digital certificate (based on the X.509 standard). This digital certificate is used to authenticate the LDAP client to the IBM Tivoli Directory Server. See “Client authentication” on page 120.

To conduct commercial business on the Internet, you might use a widely known Certification Authority (CA), such as VeriSign, to get a high assurance server certificate.

Securing your server with SSL

The following high-level steps are required to enable SSL support for IBM Directory for server authentication. These steps assume you have already installed and configured the IBM Tivoli Directory Server:

1. Install the IBM Directory GSKit package if it is not installed. See the *IBM Tivoli Directory Server Version Installation and Configuration Guide* for information on installing the GSKit package.
2. Generate the IBM Tivoli Directory Server private key and server certificate using the **gsk7ikm** utility (installed with GSKit). The server’s certificate can be signed by a commercial CA, such as VeriSign, or it can be self-signed with the **gsk7ikm** tool. The CA’s public certificate (or the self-signed certificate) must also be distributed to the client application’s key database file.
3. Store the server’s key database file and associated password stash file on the server. The default path for the key database, *<instance_directory>\etc* directory, is a typical location.
4. Access the Web-based LDAP administrative interface to configure the LDAP server. See “Using Web Administration:” on page 113 for the procedures.

If you also want to have secure communications between a master IBM Tivoli Directory Server and one or more replica servers, you must complete the following additional steps:

1. Configure the replica directory server.

Note: Follow the steps shown above for the master, except perform them for each replica. When configuring a replica for SSL, the replica is like the master with respect to its role when using SSL. The master is an LDAP client (using SSL) when communicating with a replica.

2. Configure the master directory server:
 - a. Add the replica’s signed server certificate to the master directory server’s key database file, as a trusted root. In this situation, the master directory is actually an LDAP client. If using self-signed certificates, you must extract all the self-signed certificates from each replica IBM Tivoli Directory Server, add them to the master’s key database, and ensure they are marked as trusted-roots. Essentially, you are configuring the master as an SSL client of the replica server.
 - b. Configure the master IBM Tivoli Directory Server to be aware of the replica server. Be sure to set the replicaPort attribute to use the port that the replica IBM Tivoli Directory Server uses for SSL communication.
3. Restart both the master server and each replica server.

Note: Only one key database is permitted per ldap server.

Setting Server authentication: For server authentication, you can modify the *ibmslapd.conf* file under the *cn=SSL, cn=Configuration* entry. To use the Web Administration Tool, see “Using Web Administration:” on page 113.

To use the command line:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where <filename> contains:

```
dn: cn=SSL,cn=Configuration
changetype: modify
replace: ibm-slapdSSLAuth
ibm-slapdSSLAuth: serverAuth
```

You must restart the server and the administration daemon for the changes to take effect.

Server certificate from an external Certificate Authority (CA)

In order to provide a secure connection between IBM Directory and its clients, the server must have an X.509 certificate and a private key.

The steps required to generate a private key, obtain the required server certificate from an external CA, and prepare them for use by the IBM Directory are outlined in the following:

1. Logon as administrator or root.

Note: If the administrator DN which was created during the server configuration process was `cn=root`, then enter the full administrator DN. Do not just use `root`.

2. Change to the directory where you want to create the key database file and where your private key and certificate will be stored.
3. Run `gsk7ikm` to create a new key database file. You can use any valid value for the key database file name that you want. Whatever file name you use, you need to provide it when configuring the LDAP server to use SSL. Consider providing a full path name. The `gsk7ikm` utility is used to generate a private-public key pair and a certificate request. See “Using `gsk7ikm`” on page 121 for additional information.

Note: By default, the new KDB created by GSKit is not readable by the server. You must change the owner to `idsldap`.

```
chown idsldap:idsldap <mykeyring>.*
```

See the *IBM Tivoli Directory Server version 6.0 Problem Determination Guide* for a more detailed explanation about the Kerberos service name change.

4. If VeriSign is your external CA, obtain a certificate from VeriSign, as follows:
 - a. Access the following VeriSign Web site:
<http://www.verisign.com/server/index.html>
 - b. Click on **IBM internet connection servers**.
 - c. After reviewing the information at this site, click on **Begin**.
 - d. Provide the required information and follow the steps required to request your server certificate. VeriSign is the primary Certification Authority supported for obtaining externally generated, high-assurance server certificates.
5. If you have another CA that you want to use, follow the directions for that CA to submit the contents of the certificate request file to them.

When you receive the resulting certificate from the CA:

1. Logon using your server identity.

2. Change to the directory where you created the key database file.
3. Place the signed certificate from the CA into a file in this directory. The file is used in the next step.
4. From the same directory, run **gsk7ikm** to receive the certificate into your key database file.
5. Access the LDAP server's Web administrative interface, and configure the various SSL parameters, including the file specification for the key database file. See "Using Web Administration:" on page 113.
6. If you have more than one certificate in the key database file, the certificate you want to use for IBM Directory must be the default.
7. Start the IBM Directory.

Note: If you instruct **gsk7ikm** to save the password in a password stash file, it is not necessary to change or set the password in the `ibmslapd.conf` file.

Using a self-signed server certificate

If you are using the IBM Directory in an intranet environment, use **gsk7ikm** to create your own server certificates. You can also use **gsk7ikm** to test IBM Directory with SSL without purchasing a VeriSign high-assurance server certificate. These types of certificates are known as self-signed certificates.

Follow these steps to create a key database file using self-signed certificates.

1. On each server:
 - a. Change to the directory where you want to create the key database file and where your private key and certificate is to be stored.
 - b. Create a new key database file and the self-sign certificate request that is to be used as your CA certificate.
 - Use the largest key size available.
 - Use a secure server certificate, not a low-assurance certificate.
 - c. Obtain the certificate request file. The certificate is put into the key database file automatically by the **gsk7ikm** tool.
2. If you are using an application created for the client, do the following on each client machine:
 - a. Place the CA certificate request file in an accessible location on the client machine.
 - b. Receive the CA certificate request file into the client's key database.
 - c. Mark the received certificate as a trusted root.

See "Using **gsk7ikm**" on page 121 for additional information.

Notes:

1. You must always receive the CA certificate into the server's key database file and mark it as a trusted root before receiving the server certificate into the server's key database file.
2. Whenever you use **gsk7ikm** to manage the IBM Tivoli Directory Server's key database file, remember to change to the directory in which the key database file exists.
3. Each IBM Tivoli Directory Server must have its own private key and certificate. Sharing the private key and certificate across multiple IBM Tivoli Directory Servers increases security risks. By using different certificates and private keys for each server, security exposure is minimized if a key database file for one of the servers is compromised.

Setting up your LDAP client to access IBM Directory

The following steps are required to create a key database file for an LDAP client that contains one or more self-signed server certificates that are marked as trusted by the client. The process can also be used to import CA certificates from other sources, such as VeriSign, into the client's key database file for use as trusted roots. A trusted root is simply an X.509 certificate signed by a trusted entity (for example VeriSign, or the creator of a self-signed server certificate), imported into the client's key database file, and marked as trusted.

1. Copy the server's certificate file (cert.arm) to your client workstation.
2. Run **gsk7ikm** to create a new client key database file or to access an existing one. For a new client key database, choose a file name associated with the client for ease of management. For example, if the LDAP client runs on Fred's machine, you might choose to name the file FRED.KDB.
3. If adding a server's certificate to the existing client key database:
 - a. Click **Key database file** and select **Open**.
 - b. Enter the path and name of the existing key database file then click **OK**.
 - c. Enter the password.
 - d. **Ensure signer certificates** is chosen. Click **Add**.
 - e. Enter the name and location of the server's certificate file.
 - f. Enter a label for the server certificate entry in the client's key database file, for example, Corporate Directory Server, and then click **OK**.
4. If creating the new Client key database:
 - a. Click **Key database file** and select **New**.
 - b. Enter the name and location for the new Client Key DataBase file, and then click **OK**.
 - c. Enter the password.
 - d. After the new client key database is created, repeat the previous steps for adding the server's certificate to the existing key database file.
5. Exit **gsk7ikm**.

See "Using gsk7ikm" on page 121 for additional information.

When the LDAP client creates a secure SSL connection with the server, it uses the server's self-signed certificate to verify that it is connecting to the proper server.

Repeat the preceding steps for each IBM Tivoli Directory Server that the LDAP client needs to connect to in a secure fashion.

Migrate the key ring file to key database file

To migrate the old key ring file that was created from MKKF utility:

1. Start **gsk7ikm**.
2. Click **Key database file** and select **Open**.
3. Enter the path and filename of your key ring file and then click **OK**.
4. Enter the password of your key ring file. If the key ring file is created without a password, you must use the old MKKF to assign a password for it.
5. After the old key ring file is opened, click **Key database file** and select **Save as**.
6. Ensure the key database type is set to CMS key database file. Fill out the name and location of the key database file, and then click **OK**.

Client authentication

Client authentication provides for two-way authentication between the LDAP client and the LDAP server.

With client authentication, the LDAP client must have a digital certificate (based on the X.509 standard). This digital certificate is used to authenticate the LDAP client to the IBM Tivoli Directory Server.

The Simple Authentication and Security Layer (SASL) can be used to add authentication support to connection protocols. A protocol includes a command for identifying and authenticating a user to a server. It can optionally negotiate a security layer for subsequent protocol interactions.

After a server receives the authentication command or any client response, it may issue a challenge or indicate failure or completion. If a client receives a challenge it may issue a response or end the exchange, depending on the profile of the protocol.

During the authentication protocol exchange, the SASL mechanism performs authentication, transmits an authorization identity (known as userid) from the client to the server, and negotiates the use of a mechanism-specific security layer.

When the LDAP server receives an LDAP bind request from a client, it processes the request in the following order:

1. The server parses the LDAP bind request and retrieves the following information:
 - The DN that the client is attempting to authenticate as.
 - The method of authentication used.
 - Any credentials, such as a password included in the request.
 - If the method of authentication is SASL, the server also retrieves the name of the SASL mechanism used from the LDAP bind request.
2. The server normalizes the DN retrieved from the request.
3. The server retrieves any LDAP control included with the LDAP bind request.
4. If the method of authentication is SASL, the server determines whether or not the SASL mechanism (specified in the request) is supported. If the SASL mechanism is not supported by the server, the server sends an error return code to the client and ends the bind process.
5. If the SASL mechanism is supported (=EXTERNAL) and the SSL authentication type is server and client authentication, the server verifies that the client's certificate is valid, issued by a known CA, and that none of the certificates on the client's certificate chain are invalid or revoked. If the client DN and password, as specified in the `ldap_sasl_bind`, are NULL, then the DN contained within the client's x.509v3 certificate is used as the authenticated identity on subsequent LDAP operations. Otherwise, the client is authenticated anonymously (if DN and password are NULL), or the client is authenticated based on the bind information provided by the client.
6. If the method of authentication is Simple, the server checks to see if the DN is an empty string or if there are no credentials.
7. If the DN is an empty string, or if the DN or no credentials are specified, the server assumes that the client is binding anonymously and returns a good result to the client. The DN and authentication method for the connection are left as NULL and LDAP_AUTH_NONE respectively.

8. If the client has not bound beforehand, and does not present a certificate during the bind operation, the connection is refused.

Setting client authentication: For client authentication, you can modify the `ibmslapd.conf` file under the `cn=SSL, cn=Configuration` entry. To use the Web Administration Tool, see “Using Web Administration:” on page 113.

To use the command line:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where `<filename>` contains:

```
dn: cn=SSL,cn=Configuration
cn: SSL
changetype: modify
replace: ibm-slapdSSLAuth
ibm-slapdSSLAuth: serverClientAuth
```

You must restart the server and the administration daemon for the changes to take effect.

Using `gsk7ikm`

The following key-management program, `gsk7ikm`, is provided with the IBM Global Security Kit (GSKit). It is a user-friendly GUI for managing key files, implemented as a Java applet.

Note: If you are prompted to set `JAVA_HOME`, you can set it to either the system-installed Java or the Java version included with the IBM Tivoli Directory Server. If you use the IBM Tivoli Directory Server version, you also need to set the `LIBPATH` environment variable as follows:

```
export LIBPATH=$JAVA_HOME/bin:$JAVA_HOME/bin/classic:$LIBPATH
```

Use `gsk7ikm` to create public-private key pairs and certificate requests, receive certificate requests into a key database file, and manage keys in a key database file.

Note: When setting up Secure Sockets Layer communications, ensure that you use the correct key database file type for your application. For example, Java-based applications such as the Web Administration Console require `.jks` file types, while C-applications like the IBM Tivoli Directory Server require `.cms` key database file types.

The tasks you can perform with `gsk7ikm` include:

- Creating a key pair and requesting a certificate from a certificate authority
- Receiving a certificate into a key database file
- Managing keys and certificates
 - Changing a key database password
 - Showing information about a key
 - Deleting a key
 - Making a key the default key in the key database
 - Creating a key pair and certificate request for self-signing
 - Exporting a key
 - Importing a key into a key database
 - Designating a key as a trusted root
 - Removing trusted root key designation

- Requesting a certificate for an existing key
- Migrating a keyring file to the key database format

Creating a key pair and requesting a certificate from a Certificate Authority

If your client application is connecting to an LDAP server that requires client and server authentication, then you need to create a public-private key pair and a certificate.

If your client application is connecting to an LDAP server that requires only server authentication, it is not necessary to create a public-private key pair and a certificate. It is sufficient to have a certificate in your client key database file that is marked as a trusted root. If the Certification Authority (CA) that issued the server's certificate is not already defined in your client key database, you need to request the CA's certificate from the CA, receive it into your key database, and mark it as trusted. See "Designating a key as a trusted root" on page 127.

Your client uses its private key to sign messages sent to servers. The server sends its public key to clients so that they can encrypt messages to the server, which the server decrypts with its private key.

To send its public key to a server, the client needs a certificate. The certificate contains the client's public key, the Distinguished Name associated with the client's certificate, the serial number of the certificate, and the expiration date of the certificate. A certificate is issued by a CA, which verifies the identity of the client.

The basic steps to create a certificate that is signed by a CA are:

1. Create a certificate request using **gsk7ikm**.
2. Submit the certificate request to the CA. This can be done using e-mail or an online submission from the CA's Web page.
3. Receive the response from the CA to an accessible location on the file system of your server.
4. Receive the certificate into your key database file.

Note: If you are obtaining a signed client certificate from a CA that is not in the default list of trusted CAs, you need to obtain the CA's certificate, receive it into your key database and mark it as trusted. This must be done before receiving your signed client certificate into the key database file.

To create a public-private key pair and request a certificate:

1. Start the **gsk7ikm** Java utility by typing:

```
gsk7ikm
```
2. Select **Key database file**.
3. Select **New** (or **Open** if the key database already exists).
4. Specify key database file name and location. Click **OK**.

Note: A key database is a file that the client or server uses to store one or more key pairs and certificates.

5. When prompted, supply a password for the key database file. Click **OK**.
6. Select **Create**.
7. Select **New certificate request**.
8. Supply user-assigned label for key pair. The label identifies the key pair and certificate in the key database file.

9. If you are requesting a low-assurance client certificate, enter the common name. This must be unique and the full name of the user.
10. If you are requesting a high-assurance secure server certificate, then:
 - Enter the X.500 common name of the server. Usually this is the TCP/IP fully qualified host name, for example, www.ibm.com. For a VeriSign server certificate, it must be the fully qualified host name.
 - Enter the organization name. This is the name of your organization. For a VeriSign secure server certificate, if you already have an account with VeriSign, the name in this field must match the name on that account.
 - Enter the organizational unit name. This is an optional field.
 - Enter the locality/city where the server is located. This is an optional field.
 - Enter a three-character abbreviation of the state/province where the server is located.
 - Enter the postal code appropriate for the server's location.
 - Enter the two-character country code where the server is located.
11. Click **OK**.
12. A message identifying the name and location of the certificate request file is displayed. Click **OK**.
13. Send the certificate request to the CA.
 If this is a request for a VeriSign low assurance certificate or secure server certificate, you must e-mail the certificate request to VeriSign.
 You can mail the low assurance certificate request to VeriSign immediately. A secure server certificate request requires more documentation. To find out what VeriSign requires for a secure server certificate request, go to the following URL: <http://www.verisign.com/server/index.html>.
14. When you receive the certificate from the CA, use **gsk7ikm** to receive it into the key database where you stored the key pair. See "Receiving a certificate into a key database."

Note: Change the key database password frequently. If you specify an expiration date, you need to keep track of when you need to change the password. If the password expires before you change it, the key database is not usable until the password is changed.

Receiving a certificate into a key database

After receiving a response from your CA, you need to receive the certificate into a key database.

To receive a certificate into a key database:

1. Type **gsk7ikm** to start the Java utility.
2. Select **Key database file**.
3. Select **Open**.
4. Specify the key database file name and location. Click **OK**.
5. When prompted, supply a password for the key database file. Click **OK**.
6. Select **Create**.
7. Select **Personal certificates** in the middle window.
8. Click **Receive**.
9. Enter the name and location of the certificate file that contains the signed certificate, as received from the CA. Click **OK**.

Changing a key database password

To change a key database password:

1. Type `gsk7ikm` to start the Java utility.
2. Select **Key database file**.
3. Select **Open**.
4. Specify the key database file name and location. Click **OK**.
5. When prompted, supply the password for the key database file. Click **OK**.
6. Select **Key database file**.
7. Select **Change password**.
8. Enter `<New password>`.
9. Confirm `<New password>`.
10. Select and set an optional password expiration time.
11. Select **Stash the password to a file?** if you want the password to be encrypted and stored on disk.
12. Click **OK**.
13. A message is displayed with the file name and location of the stash password file. Click **OK**.

Note: The password is important because it protects the private key. The private key is the only key that can sign documents or decrypt messages encrypted with the public key.

Showing information about a key

To show information about a key, such as its name, size or whether it is a trusted root:

1. Type `gsk7ikm` to start the Java utility.
2. Select **Key database file**.
3. Select **Open**.
4. Specify a key database file name and location. Click **OK**.
5. When prompted, supply the password for the key database file. Click **OK**.
6. To see information about keys designated as Personal certificates:
 - Select **Personal certificates** at the top of the **Key database content** window.
 - Select a certificate.
 - Click **View/Edit** to display information about the selected key.
 - Click **OK** to return to the list of Personal Certificates.
7. To see information about keys that are designated as Signer Certificates:
 - Select **Signer certificates** at the top of the **Key database content** window.
 - Select a certificate .
 - Click **View/Edit** to display information about the selected key.
 - Click **OK** to return to the list of Signer Certificates.

Deleting a Key

To delete a key:

1. Type `gsk7ikm` to start the Java utility.
2. Select **Key database file**.
3. Select **Open**.
4. Specify key a database file name and location. Click **OK**.
5. When prompted, supply the password for the key database file. Click **OK**.

6. Select the type of key you want to delete at the top of the **Key database content** window (Personal certificates, Signer certificates, or Personal certificate requests).
7. Select a certificate.
8. Click **Delete**.
9. Click **Yes** to confirm.

Making a key the default key in the key ring

The default key must be the private key that the server uses for its secure communications.

To make a key the default key in the key ring:

1. Type `gsk7ikm` to start the Java utility.
2. Select **Key database file**.
3. Select **Open**.
4. Specify a key database file name and location. Click **OK**.
5. When prompted, supply the password for the key database file. Click **OK**.
6. Select **Personal certificates** at the top of the **Key database content** window.
7. Select the desired certificate.
8. Click **View/Edit**.
9. Select the **Set the certificates as the default** box. Click **OK**.

Creating a key pair and certificate request for self-signing

By definition, a secure server must have a public-private key pair and a certificate.

The server uses its private key to sign messages to clients. The server sends its public key to clients so they can encrypt messages to the server, which the server decrypts with its private key.

The server needs a certificate to send its public key to clients. The certificate contains the server's public key, the distinguished name associated with the server's certificate, the serial number of the certificate, and the expiration date of the certificate. A certificate is issued by a CA, who verifies the identity of the server.

You can request one of the following certificates:

- A low assurance certificate from VeriSign, best for non-commercial purposes, such as a beta test of your secure environment
- A server certificate to do commercial business on the Internet from VeriSign or some other CA
- A self-signed server certificate if you plan to act as your own CA for a private Web network

For information about using a CA such as VeriSign to sign the server certificate, see "Creating a key pair and requesting a certificate from a Certificate Authority" on page 122.

The basic steps to creating a self-signed certificate are:

1. Type `gsk7ikm` to start the Java utility.
2. Select **Key database file**.
3. Select **New**, or **Open** if the key database already exists.
4. Specify a key database file name and location. Click **OK**.

Note: A key database is a file that the client or server uses to store one or more key pairs and certificates.

5. When prompted, supply the password for the key database file. Click **OK**.
6. Click **New self-signed**.
7. Supply the following:
 - User-assigned label for the key pair. The label identifies the key pair and certificate in the key database file.
 - The desired certificate Version.
 - The desired Key Size.
 - The X.500 common name of the server. Usually this is the TCP/IP fully qualified host name, for example, `www.ibm.com`.
 - The organization name. This is the name of your organization.
 - The organizational unit name. This is an optional field.
 - The locality/city where the server is located. This is an optional field.
 - A three-character abbreviation of the state/province where the server is located.
 - The zip code appropriate for the server's location.
 - The two-character country code where the server is located.
 - The Validity Period for the certificate.
8. Click **OK**.

Exporting a key

If you need to transfer a key pair or certificate to another computer, you can export the key pair from its key database to a file. On the other computer, you can import the key pair into a key ring.

To export a key from a key database:

1. Type `gsk7ikm` to start the Java utility.
2. Select **Key database file**.
3. Select **Open**.
4. Specify a key database file name and location. Click **OK**.
5. When prompted, supply the password for the key database file. Click **OK**.
6. Select **Personal certificates** at the top of the **Key database content** window.
7. Select the desired certificate.
8. Click **Export/Import**.
9. For **Action type**, select **Export key**.
10. Select the Key file type:

Note: The IBM Tivoli Directory Server requires `.cms` key database file types.

- PKCS12 file
 - CMS Key database file
 - Keyring file (as used by `mkkf`)
 - SSLight key database class
11. Specify a file name.
 12. Specify the location.
 13. Click **OK**.
 14. Enter the required password for the file. Click **OK**.

Importing a key

To import a key into a key ring:

1. Type `gsk7ikm` to start the Java utility.
2. Select **Key database file**.
3. Select **Open**.
4. Specify the key database file name and location. Click **OK**.
5. When prompted, supply the password for the key database file. Click **OK**.
6. Select **Personal certificates** at the top of the **Key database content** window.
7. Select the desired certificate.
8. Click **Export/Import**.
9. For **Action type**, select **Import key**.
10. Select the desired Key file type.

Note: When setting up Secure Sockets Layer communications, ensure that you use the correct key database file type for your application. For example, Java-based applications such as the Web Administration Console require .jks file types, while C-applications like the IBM Tivoli Directory Server require .cms key database file types.

11. Enter the file name and location.
12. Click **OK**.
13. Enter the required password for the source file. Click **OK**.

Designating a key as a trusted root

A trusted root key is the public key and associated distinguished name of a CA. The following trusted roots are automatically defined in each new key database:

- Integriion Certification Authority Root
- IBM World Registry™ Certification Authority
- Thawte Personal Premium CA
- Thawte Personal Freemail CA
- Thawte Personal Basic CA
- Thawte Premium Server CA
- VeriSign Test CA Root Certificate
- RSA Secure Server Certification Authority
- VeriSign Class 1 Public Primary Certification Authority
- VeriSign Class 2 Public Primary Certification Authority
- VeriSign Class 3 Public Primary Certification Authority
- VeriSign Class 4 Public Primary Certification Authority

Note: Each of these trusted roots are initially set to be trusted roots by default.

To designate a key as a trusted root:

1. Type `gsk7ikm` to start the Java utility.
2. Select **Key database file**.
3. Select **Open**.
4. Specify a key database file name and location. Click **OK**.
5. When prompted, supply the password for the key database file. Click **OK**.
6. Select **Signer certificates** at the top of the **Key database content** window.
7. Select the desired certificate.

8. Click **View/Edit**.
9. Check the **Set the certificate as a trusted root** box, and click **OK**.
10. Select **Key database file** and then select **Close**.

Removing a key as a trusted root

A trusted root key is the public key and associated distinguished name of a CA. The following trusted roots are automatically defined in each new key database:

- Integrion Certification Authority Root
- IBM World Registry Certification Authority
- Thawte Personal Premium CA
- Thawte Personal Freemail CA
- Thawte Personal Basic CA
- Thawte Premium Server CA
- VeriSign Test CA Root Certificate
- RSA Secure Server Certification Authority
- VeriSign Class 1 Public Primary Certification Authority
- VeriSign Class 2 Public Primary Certification Authority
- VeriSign Class 3 Public Primary Certification Authority
- VeriSign Class 4 Public Primary Certification Authority

Note: Each of these trusted roots are initially set to be trusted roots by default.

To remove the trusted root status of a key:

1. Type `gsk7ikm` to start the Java utility.
2. Select **Key database file**.
3. Select **Open**.
4. Specify the key database file name and location. Click **OK**.
5. When prompted, supply the password for the key database file. Click **OK**.
6. Select **Signer certificates** at the top of the **Key database content** window.
7. Select the desired certificate.
8. Click **View/Edit**.
9. Clear the **Set the certificate as a trusted root** box. Click **OK**.
10. Select **Key database file** and then select **Close**.

Requesting a certificate for an existing key

To create a certificate request for an existing key:

1. Type `gsk7ikm` to start the Java utility.
2. Select **Key database file**.
3. Select **Open**.
4. Specify the key database file name and location. Click **OK**.
5. When prompted, supply the password for the key database file. Click **OK**.
6. Select **Personal Certificates** at the top of the **Key database content** window.
7. Select the desired certificate.
8. Click **Export/Import**.
9. For **Action type**, select **Export key**.
10. Select the desired Data type:
 - Base 64-encoded ASCII data

- Binary DER data
 - SSLight Key Database Class
11. Enter the certificate file name and location.
 12. Click **OK**.
 13. Select **Key database file** and then select **Close**.

Send the certificate request to the CA.

If this is a request for a VeriSign low assurance certificate or secure server certificate, you must e-mail the certificate request to VeriSign.

You can mail the low assurance certificate request to VeriSign immediately. A secure server certificate request requires more documentation. To find out what VeriSign requires for a secure server certificate request, go to the following URL: <http://www.verisign.com/server/index.html>.

Migrating a key ring file to the key database format

The `gsk7ikm` program can be used to migrate an existing key ring file, as created with `mkkf`, to the format used by `gsk7ikm`.

To migrate a key ring file:

1. Type `gsk7ikm` to start the Java utility.
2. Select **Key database file**.
3. Select **Open**.
4. Specify the key database file name and location. Click **OK**.
5. When prompted, supply the password for the key ring file. Click **OK**.
6. Select **Key database file**.
7. Select **Save as...**
8. Select **CMS key database file** as the Key database type.
9. Specify a file name.
10. Specify location.
11. Click **OK**.

Setting the key database

To set the key database, use one of the following procedures.

Using Web Administration:

Expand the **Manage security properties** category in the navigation area of the Web Administration Tool, select the **Key database** tab.

1. Specify the **Key label**. This administrator-defined key label indicates what part of the key database to use.
2. Specify the **Key database path and file name**. This is the fully qualified file specification of the key database file. If a password stash file is defined, it is assumed to have the same file specification, with an extension of `.sth`.
3. Specify the **Key password**. If a password stash file is not being used, the password for the key database file must be specified here. Then specify the password again in the **Confirm password** field.
4. When you are finished, click **Apply** to save your changes without exiting, or click **OK** to apply your changes and exit, or click **Cancel** to exit this panel without making any changes.

Note: In order for the server to use this file, it must be readable by the user ID **ldap**. See the *IBM Tivoli Directory Server version 6.0 Problem Determination Guide* for information about file permissions.

Using the command line:

To use the command line to set the key database for SSL and TLS, issue the command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where <filename> contains:

```
dn: cn=SSL,cn=Configuration
changetype: modify
replace: ibm-slapdSSLKeyDatabase
ibm-slapdSSLKeyDatabase: <databasename>
-
replace: ibm-slapdSSLKeyDatabasePW
ibm-slapdSSLKeyDatabasePW: <password>
-
replace: ibm-slapdSslKeyRingFilePW
ibm-slapdSslKeyRingFilePW: <password>
```

You must restart the server and the administration daemon for the changes to take effect.

Setting the level of encryption for SSL and TLS communications

By default the SSL and TLS versions of IBM Tivoli Directory Server uses the following list of ciphers when performing cipher negotiation with the client (during the SSL or TLS handshake).

Note: Although the password policy feature is not available in configuration only mode, you can change your level of password encryption in configuration only mode.

Using Web Administration:

Expand the Server administration category in the navigation area in the Web Administration Tool.

1. Click **Manage security properties**.
2. Click **Encryption**.
3. Select the method of encryption that you want to use based on the clients accessing the server. AES-128 is the default level of encryption. If you select multiple encryption methods, the highest level of encryption is used by default, however clients using the selected lower encryption levels still have access to the server.

Note: The IBM Tivoli Directory Server Version supports the Advanced Encryption Standard (AES) level of encryption. For information on AES, see the NIST Web page at <http://csrc.nist.gov/encryption/aes/>.

Table 11. Supported levels of encryption

Encryption level	Attribute
Triple DES encryption with a 168-bit key and a SHA-1 MAC	ibm-slapdSslCipherSpec: TripleDES-168
DES encryption with a 56-bit key and a SHA-1 MAC	ibm-slapdSslCipherSpec: DES-56

Table 11. Supported levels of encryption (continued)

Encryption level	Attribute
RC4 encryption with a 128-bit key and a SHA-1 MAC	ibm-slapdSslCipherSpec: RC4-128-SHA
RC4 encryption with a 128-bit key and a MD5 MAC	ibm-slapdSslCipherSpec: RC4-128-MD5
RC2 encryption with a 40-bit key and a MD5 MAC	ibm-slapdSslCipherSpec: RC2-40-MD5
RC4 encryption with a 40-bit key and a MD5 MAC	ibm-slapdSslCipherSpec: RC4-40-MD5
AES 128-bit encryption	ibm-slapdSslCipherSpec: AES-128
AES 256-bit encryption	ibm-slapdSslCipherSpec: AES

Note: SSL and TLS do not support AES 192 encryption.

The selected ciphers are stored in the configuration file using the `ibm-slapdsslCipherSpec` keyword and the attribute defined from the preceding table. For example, to use only Triple DES, select **Triple DES encryption with a 168-bit key and an SHA-1 MAC**. The attribute `ibm-slapdSslCipherSpec: TripleDES-168` is added to the `ibmslapd.conf` file. In this case, only clients that also support Triple DES are able to establish an SSL connection with the server. You can select multiple ciphers.

- If your server supports the Federal Information Processing Standards (FIPS) mode enablement feature, under the heading "Implementation" a preselected **Use FIPS certified implementation** check box is displayed. This enables the server to use the encryption algorithms from the ICC FIPS-certified library. If you deselect this check box the encryption algorithms from a non-FIPS certified library are used.

Note: The server can be configured to turn FIPS Processing Mode on. It requires the FIPS-enabled libraries to also be on.

- When you are finished, click **Apply** to save your changes without exiting, or click **OK** to apply your changes and exit, or click **Cancel** to exit this panel without making any changes.

Using the command line:

To use the command line to set the SSL level of encryption (in this example to Triple DES encryption with a 168-bit key and an SHA-1 MAC) issue the command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where `<filename>` contains:

```
dn: cn=SSL,cn=Configuration
changetype: modify
replace: ibm-slapdSslCipherSpec
ibm-slapdSslCipherSpec: TripleDES-168
```

See Table 11 on page 130 for other encryption values.

To add more than one level of encryption, your `<filename>` might contain:

```
dn: cn=SSL,cn=Configuration
changetype: modify
replace: ibm-slapdSslCipherSpec
ibm-slapdSslCipherSpec: RC2-40-MD5
ibm-slapdSslCipherSpec: AES
```

```
ibm-slapdSslCipherSpec: AES-128
ibm-slapdSslCipherSpec: RC4-128-MD5
ibm-slapdSslCipherSpec: RC4-128-SHA
ibm-slapdSslCipherSpec: TripleDES-168
ibm-slapdSslCipherSpec: DES-56
ibm-slapdSslCipherSpec: RC4-40-MD5
```

To use the command line to turn off FIPS mode, issue the command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where <filename> contains:

```
dn: cn=SSL,cn=Configuration
changetype: modify
replace: ibm-slapdSslFIPSMODEEnabled
ibm-slapdSslFIPSMODEEnabled: false
```

You must restart the server and the administration daemon for the changes to take effect.

Password encryption

IBM Directory enables you to prevent unauthorized access to user passwords. Using one-way encryption formats, user passwords may be encrypted and stored in the directory, which prevents clear passwords from being accessed by any users including the system administrators.

The administrator may configure the server to encrypt userPassword attribute values in either a one-way encrypting format or a two-way encrypting format.

One-way encrypting formats:

- SHA-1
- crypt

After the server is configured, any new passwords (for new users) or modified passwords (for existing users) are encrypted before they are stored in the directory database. The encrypted passwords are tagged with the encrypting algorithm name so that passwords encrypted in different formats can coexist in the directory. When the encrypting configuration is changed, existing encrypted passwords remain unchanged and continue to work.

For applications that require retrieval of clear passwords, such as middle-tier authentication agents, the directory administrator needs to configure the server to perform either a two-way encrypting or no encryption on user passwords. In this instance, the clear passwords stored in the directory are protected by the directory ACL mechanism.

Two-way encrypting format:

- AES

A two-way encryption option, AES, is provided to allow values of the userPassword attribute to be encrypted in the directory and retrieved as part of an entry in the original clear format. It can be configured to use 128-, 192-, and 256-bit key lengths. Some applications such as middle-tier authentication servers require passwords to be retrieved in clear text format, however, corporate security policies might prohibit storing clear passwords in a secondary permanent storage. This option satisfies both requirements.

A simple bind will succeed if the password provided in the bind request matches any of the multiple values of the userPassword attribute.

When you configure the server using Web Administration, you can select one of the following encryption options:

None No encryption. Passwords are stored in the clear text format.

crypt Passwords are encrypted by the UNIX crypt encrypting algorithm before they are stored in the directory.

SHA-1

Passwords are encrypted by the SHA-1 encrypting algorithm before they are stored in the directory.

AES128

Passwords are encrypted by the AES128 algorithm before they are stored in the directory and are retrieved as part of an entry in the original clear format.

AES192

Passwords are encrypted by the AES192 algorithm before they are stored in the directory and are retrieved as part of an entry in the original clear format.

AES256

Passwords are encrypted by the AES256 algorithm before they are stored in the directory and are retrieved as part of an entry in the original clear format.

Note: The imask format that was available in previous releases is no longer an encryption option. However, any existing imask encrypted values still work.

The default option is AES256. A change is registered in a password encryption directive of the server configuration file:

```
ibm-SlapdPwEncryption: AES256
```

The server configuration file is located in:

```
<instance_directory>\etc\ibmslapd.conf
```

In addition to userPassword, values of the secretKey attribute are always "AES256" encrypted in the directory. Unlike userPassword, this encrypting is enforced for values of secretKey. No other option is provided. The secretKey attribute is an IBM defined schema. Applications may use this attribute to store sensitive data that need to be always encrypted in the directory and to retrieve the data in clear text format using the directory access control.

Consult the *IBM Tivoli Directory Server Version 6.0 Installation and Configuration Guide* for additional information about the configuration file.

To change the type of encryption using the command line, for example to crypt, issue the following command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where <filename> contains:

```
dn: cn=configuration
changetype: modify
replace: ibm-slapdPwEncryption
ibm-slapdPwEncryption: crypt
```

To cause the updated settings to take effect dynamically, issue the following `idsldapexop` command:

```
idsldapexop -D <adminDN> -w <adminPW> -op readconfig -scope single
"cn=configuration" ibm-slapdPWEncryption
```

Notes:

1. If the UNIX crypt method is used, only the first 8 characters are effective.
2. A one-way encrypted password can be used for password matching but it cannot be decrypted. During user login, the login password is encrypted and compared with the stored version for matching verification.

Password policy attributes

The password policy feature provides several operational attributes containing the password policy state information for a given directory entry. These attributes can be used to query for entries in a particular state (password has expired) and by an administrator to override certain policy conditions (unlock a locked account). See Appendix G, “Password policy operational attributes,” on page 495

Setting password policy

Password policy is a set of rules that controls how passwords are used and administered in the IBM Directory. These rules are made to ensure that users change their passwords periodically, and that the passwords meet the organization’s syntactic password requirements. These rules also can restrict the reuse of old passwords and ensure that users are locked out after a defined number of failed bind attempts.

For additional information about passwords see “Password Guidelines” on page 139.

Summary of settings for an EAL4 secure configuration

These settings are the minimum required for your user’s password policy configuration to be considered secure by Common Criteria Evaluation Assurance Level 4 (EAL4) standards.

Table 12. User password policy settings

Web Administration Tool parameter	Default setting	EAL4 setting
Password policy enabled: <code>ibm-pwdPolicy</code>	false	true
Password encryption: <code>ibm-slapdPwEncryption</code> :	AES256	AES256
Users must specify old password when changing the password: <code>pwdSafeModify</code>	false	true
User must change password after reset: <code>pwdMustChange</code>	true	true
Password expiration: <code>pwdMaxAge</code>	0	7776000
Number of grace logins after expiration: <code>pwdGraceLoginLimit</code>	0	0
Account is locked out after a specified number of consecutive failed bind attempts: <code>pwdLockout</code>	false	true
Number of consecutive failed bind attempts before locking out the account: <code>pwdMaxFailure</code>	0	3

Table 12. User password policy settings (continued)

Web Administration Tool parameter	Default setting	EAL4 setting
Minimum time between password changes: pwdMinAge	0	86400
Amount of time before an account lockout expires or lockouts never expire: pwdLockoutDuration	0	0
Amount of time before an incorrect login expires or incorrect login is cleared only with correct password: pwdFailureCountInterval	0	0
Minimum number of passwords before reuse: pwdInHistory	0	10
Check password syntax: pwdCheckSyntax	0	1
Minimum length: pwdMinLength	0	8
Minimum number of alphabetic characters: passwordMinAlphaChars	0	4
Minimum number of numeric and special characters: passwordMinOtherChars	0	2
Maximum number of repeated characters: passwordMaxRepeatedChars	0	2
Minimum number of characters that must be different from the old password: passwordMinDiffChars	0	2

All users except the directory administrator, members of the administrative group and the master server DN are forced to comply with the configured user password policy. The passwords for the administrator, members of the administrative group and the master server DN never expire. The directory administrator, members of the administrative group and the master server DN have sufficient access control privileges to modify users' passwords and the user password policy. Global administration group members are subject to user password policy and have the authority to modify the user password policy settings.

The password policy for administrators, members of the administrative group and the master server DN is set in the configuration file.

Table 13. Administration Password Policy Settings

Administration password requirements	Default setting	EAL4 setting
Password policy enabled: ibm- slapdConfigPwdPolicyOn	false	true
Account is locked out after a specified number of consecutive failed bind attempts: pwdLockout	true	true
Maximum number of incorrect logins until password lockout: pwdMaxFailure	10	10
Amount of time before an account lockout expires or lockouts never expire: pwdLockoutDuration	300	300
Amount of time before an incorrect login expires or incorrect login is cleared only with correct password: pwdFailureCountInterval	0	0
Minimum length: pwdMinLength	8	8

Table 13. Administration Password Policy Settings (continued)

Administration password requirements	Default setting	EAL4 setting
Minimum number of alphabetic characters: passwordMinAlphaChars	2	2
Minimum number of numeric and special characters: passwordMinOtherChars	2	2
Maximum number of repeated characters: passwordMaxRepeatedChars	2	2
Minimum number of characters that must be different from the old password: passwordMinDiffChars	2	2

Administration password policy is set to false by default. Turning on the administration password policy, enables the other attributes with the default settings. The default settings are the minimum requirements for an EAL4 secure configuration.

Setting the administration password and lockout policy

Note: The administration password policy is set using the command line only. The Web administration tool does not support administration password policy.

To turn on the administration password policy with an EAL4 secure configuration, issue the following command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where <filename> contains:

```
dn: cn=pwdPolicy Admin,cn=Configuration
changetype: modify
replace: ibm-slapdConfigPwdPolicyOn
ibm-slapdConfigPwdPolicyOn: true
```

To enable the administration password policy and modify the default settings, issue the following command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where <filename> contains:

```
dn: cn=pwdPolicyAdmin,cn=Configuration
changetype: modify
replace: ibm-slapdConfigPwdPolicyOn
ibm-slapdConfigPwdPolicyOn: TRUE
-
replace: pwdlockout
pwdlockout: TRUE
#select TRUE to enable, FALSE to disable
-
replace:pwdmaxfailure
pwdmaxfailure: 10
-
replace:pwdlockoutduration
pwdlockoutduration: 300
-
replace:pwdfailurecountinterval
pwdfailurecountinterval: 0
-
replace:pwdminlength
```

```
pwdminlength: 8
-
replace:passwordminalphachars
passwordminalphachars: 2
-
replace:passwordminotherchars
passwordminotherchars: 2
-
replace:passwordmaxrepeatedchars
passwordmaxrepeatedchars: 2
-
replace:passwordmindiffchars
passwordmindiffchars: 2
```

Setting the user password policy

To set the password policy for a user password, use one of the following procedures.

Using Web Administration:

Expand the **Manage security properties** category in the navigation area of the Web Administration Tool, select the **Password policy** tab. This panel displays a noneditable **Password attribute** field that contains the name of the attribute that the password policy is using.

1. Select the type of password encryption from the drop-down list:

- AES128
- AES192
- AES256
- crypt
- SHA
- None

For an EAL4 secure password configuration, select **AES256**. See “Password encryption” on page 132 for additional information.

2. Select the **Password policy enabled** check box to enable password policy.

Note: If Password policy is not enabled, none of the other functions on this or the other password panels are available until the check box is enabled. By default password policy is disabled.

For an EAL4 secure password configuration, **Password policy enabled** must be selected.

3. Select the **User can change password** check box to specify whether the user can change the password.

4. Select the **User must change password after reset** check box to specify whether the user must change the password after logging on with a reset password. For an EAL4 secure password configuration, **User must change password after reset** must be selected.

Note: If you are running applications that have not been coded using the LDAP password policy controls, this policy option is not enforced on binds, however no subsequent operations are allowed unless the password is changed.

5. Select the **User must send password when changing** check box to specify whether the user, after the initial log on, needs to specify the existing password before being able to change the password. For an EAL4 secure password configuration, **User must send password when changing** must be selected.

6. Set the password expiration limit. Click the **Password Never Expires** radio button to specify that the password does not have to be changed at a specific time interval or click the **Days** radio button and specify the time interval, in days, when the password needs to be reset. For an EAL4 secure password configuration, click **Days** and specify **90**.
7. Specify whether the system issues a password expiration warning, before the password expires. If you click the **Never warn** radio button, the user is not warned before the previous password expires. The user cannot access the directory until the administrator has created a new password. If you click the **Days before expiration** radio button and specify a number of days (*n*), the user receives a warning prompt to change the password each time the user logs on, starting *n* days before the password expires. The user can still access the directory until the password expires.
8. Specify the number of times, if any, that the user can log on after the password has expired. This selection enables the user to access the directory with an expired password. For an EAL4 secure password configuration, specify **0**.
9. When you are finished, click **Apply** to save your changes without exiting, or click **OK** to apply your changes and exit, or click **Cancel** to exit this panel without making any changes.

Note: Password safemodify setting applies only to a user attempting to update their password. If one user has permission to change the password of another user, then the first user has the authority to become the second user by binding as the second user with the newly modified password. When configuring ACLs, an administrator must be sure that they really want the first user to have authority to bind as another user. Any user that has the authority to modify another user's password must be a trusted user.

Using the command line:

To perform the same operations using the command line, issue the following command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where <filename> contains:

```
dn: cn=configuration
changetype: modify
replace: ibm-slapdPwEncryption
ibm-slapdPwEncryption: AES256
#select AES128, AES192, AES256, crypt, SHA, or none
-
dn: cn=pwdpolicy
changetype: modify
replace: ibm-pwdpolicy
ibm-pwdpolicy: TRUE
#select TRUE to enable, FALSE to disable
-
replace:pwdallowuserchange
pwdallowuserchange: {TRUE|FALSE}
#select TRUE to enable, FALSE to disable
-
replace:pwdmustchange
pwdmustchange: TRUE
#select TRUE to enable, FALSE to disable
#Note:If you are running applications that have not been coded
#using the LDAP password policy controls, this policy option is
#not enforced on binds, however no subsequent operations are allowed
#unless the password is changed.
-
replace:pwdsafemodify
```



```

pwsafemodify: TRUE
#select TRUE to enable, FALSE to disable
-
replace:pwdmaxage
pwdmaxage: 7776000
#in seconds 776000=90days
-
replace:pwdexpirewarning
pwdexpirewarning: 86400
#in seconds 86400=1day
-
replace:pwdgracelogleinlimit
pwdgracelogleinlimit: 0

```

Note: Those settings in **bold** type are required for an EAL4 secure password configuration.

If you change the encryption setting, you must issue a dynamic configuration extended operation:

```

idsldapexop -D <admin DN> -w <admin pw> -op readconfig -scope single
cn=configuration ibm-slapdPwEncryption

```

Unlocking administrative accounts

When an administrator unlocks an account by modifying a local admin group member or master server DN password, the account is unlocked immediately. However, the password modification for a local admin group member does not take effect until a dynamic configuration update request is made. The password modification for a master server DN password does not happen until the server is stopped and restarted. When an administrator changes a configuration file, the administrator must issue a dynamic update request immediately.

```

idsldapmodify -D <adminDN> -w <adminPW> -i <filename>

```

where <filename> contains:

```

dn: cn=admin1,cn=admingroup,cn=configuration
changetype: modify
replace: ibm-slapdadminpw
ibm-slapdadminpw: newpassword123

```

Note: When the administrator's account is locked, the only way to unlock the account is by logging on to the local console.

Password Guidelines

The following section provides details of the supported values of the IBM Tivoli Directory Server password attribute for user entries in the IBM Tivoli Directory Server, as well as the accounts used to administer the LDAP environment. It also provides guidelines of what characters to avoid to reduce confusion when attempting to run the Directory Server command line tools and C-API interfaces.

The IBM Tivoli Directory Server has two types of user accounts:

- Administration accounts (LDAP Administrator (cn=root), members of the Administrator Group, or the master server DN) that are stored in the <instance_directory>/etc/ibmslapd.conf file.
- User entries (iNetOrgPerson) that have a password attribute used with Directory Server C and Java (JNDI) APIs. These are the interfaces that applications, such as Tivoli Access Manager and WebSphere use. While the Directory Server supports a wide variety of values for password entries, you need to review the application documentation to confirm what guidelines or restrictions apply.

Note: Global administration group members are stored in the directory.

Details of the supported password values using the IBM Tivoli Directory Server 6.0 release are explained in the following sections.

Note: The LDAP DB2 user is stored in the configuration file, but is not subject to password policy.

Passwords for user entries (InetOrgPerson)

Using the 6.0 release, the following characters are supported for the userPassword attribute field to be stored in the Directory Server using the C and java APIs. Applications, such as Policy Director, WebSphere, and so on, that are using the Directory Server might have additional restrictions on password values. For details, review the product documentation for these specific products.

- All upper and lower case English alpha and numeric characters.
- All other ASCII single-byte characters are supported.
- Double-byte characters are supported for languages specified in the *IBM Tivoli Directory Server Version 6.0 Installation and Configuration Guide*.
- Passwords are case sensitive. (For example, if the password = TeSt, using a password of TEST or test fails. Only the exact case, TeSt, works.)

LDAP ibmslapd.conf users:

Using the 6.0 release, the following characters are supported for passwords of users that are in the `<instance_directory>/etc/ibmslapd.conf` file.

- All uppercase and lowercase English alpha and numeric characters are supported.
- All other ASCII single-byte characters are supported.
- Passwords are case sensitive. (For example, if the password = TeSt, using a password of TEST or test fails. Only the exact case, TeSt, works.)

Notes:

1. The defined users in the `ibmslapd.conf` file can include the following:
 - LDAP Administrator (cn=root)
 - Members of the Administrator Group
 - Master ID for Replication (cn=MASTER)
 - LDAP DB2 users for LDAP DB entry and change log databases (LDAPDB2)

Note: The user LDAP password guidelines do not apply to these users.

2. Double-byte characters in the administrator passwords are not supported.

Using the Web Administration Tool to modify password attributes:

Using the Web Administration Tool in the 6.0 release, the following characters are supported for adding or modifying the password attribute field:

- All uppercase and lowercase English alpha and numeric characters are supported.
- All other ASCII single-byte characters are supported.
- Passwords are case sensitive. (For example, if the password = TeSt, using a password of TEST or test fails. Only the exact case, TeSt, works.)

Notes:

1. Double-byte characters are not supported for the administrator password.
2. Double-byte characters are supported for user passwords.

Special characters

Avoid using the following characters because the operating shell might interpret these "special" characters:

```
~  
,  
\  
"  
|
```

For example, using the 6.0 Web Administration Tool to assign a user password attribute to the value:

```
"\test\'
```

requires the following password from the command line to be used:

```
-w\\"\\test\'
```

Here is an example search:

```
idsldapsearch -b" " -sbase -Dcn=newEntry,o=ibm,c=us -w\\"\\test\' objectclass=*
```

Note: This password works in the Web Administration Tool's Java application using the original password without the escape character. In the previous example, the Web Administration Tool bind password is the same as the one that was entered when assigning the password in the Web Administration Tool:

```
"\test\'
```

Setting user password lockout

To set the conditions that lock a password, use one of the following procedures.

Note: If password policy is not enabled on the server, the password lockout functions do not take effect.

Using Web Administration:

Expand the **Manage server properties** category in the navigation area of the Web Administration Tool, select the **Password lockout** tab.

Note: If password policy is not enabled on the server, the functions on this panel do not take effect.

1. Specify the number of seconds, minutes, hours or days that must expire after a password has been changed before the password can be changed again. For an EAL4 secure password configuration, specify **1 day** or its equivalent.
2. Specify whether incorrect logins lockout the password.
 - Select the **Passwords are never locked out** radio button if you want to allow unlimited log in attempts. This selection disables the password lockout function.
 - Select the **Attempts** radio button and specify the number of log in attempts that are allowed before locking out the password. This selection enables the password lockout function.

For an EAL4 secure password configuration, select **Attempts** and specify **3**.

3. Specify the duration of the lockout. Select the **Lockouts never expires** radio button to specify that the system administrator must reset the password or select the **Seconds** radio button and specify the number of seconds before the

lockout expires and log in attempts can resume. For an EAL4 secure password, configuration select **Lockouts never expires**.

4. Specify the expiration time for an incorrect login. Click the **Incorrect logins only cleared with correct password** radio button to specify that incorrect logins are cleared only by a successful login or click the **Seconds** radio button and specify the number of seconds before an unsuccessful login attempt is cleared from memory. For an EAL4 secure password configuration, click **Incorrect logins only cleared with correct password**.

Note: This option works only if the password is not locked out.

5. When you are finished, click **Apply** to save your changes without exiting, or click **OK** to apply your changes and exit, or click **Cancel** to exit this panel without making any changes.

Using the command line:

To perform the same operations using the command line, issue the following command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where <filename> contains:

```
dn: cn=pwdpolicy
changetype: modify
replace: pwdlockout
pwdlockout: TRUE
#select TRUE to enable, FALSE to disable
-
replace:pwdminage
pwdminage: 86400
#specify in seconds 86400=1day
-
replace:pwdmaxfailure
pwdmaxfailure: 3
-
replace:pwdlockoutduration
pwdlockoutduration: 0
-
replace:pwdfailurecountinterval
pwdfailurecountinterval: 0
```

Note: Those settings in **bold** type are required for an EAL4 secure password configuration.

Setting user password validation

To set the requirements and limitations for validating a user password, use one of the following procedures.

Using Web Administration:

Expand the **Manage server properties** category in the navigation area of the Web Administration Tool, select the **Password validation** tab.

Note: If password policy is not enabled on the server, the functions on this panel do not take effect.

1. Set the number of passwords that must be used before a password can be reused. Enter a number from 0 to 30. If you enter zero, a password may be reused without restriction. For an EAL4 secure password configuration, specify **10**.

2. From the drop-down menu, select whether the password is checked for the syntax defined in the following entry fields. You can select:

Do not check syntax

No syntax checking is performed.

Check syntax (two-way encrypted only)

The syntax checking is performed on all two-way encrypted passwords.

Check syntax

The syntax checking is performed on all passwords.

Note: If this option is selected and the password encryption is set either to sha or crypt, all password modifications fail. This failure is caused by the fact that the server cannot read these one-way encrypted passwords and therefore cannot validate them.

For an EAL4 secure password configuration, select **Check syntax (except encrypted)**.

3. Specify a number value to set the minimum length of the password. If the value is set to zero, no syntax checking is performed. For an EAL4 secure password configuration, specify 8.
 - Specify a number value to set the minimum number of alphabetic characters required for the password. For an EAL4 secure password configuration, specify 4.
 - Specify a number value to set the minimum number of numeric and special characters required for the password. For an EAL4 secure password configuration, specify 2.

Note: The sum of the minimum number of alphabetic, numeric, and special characters must be equal to or less than the number specified as the minimum length of the password.

4. Specify the maximum number of times that characters can be repeated in the password. This option limits the total number of times a specific character can appear in the password. If the value is set to zero, the number of repeated characters is not checked. For an EAL4 secure password configuration, specify 2.
5. Specify the minimum number of characters that must be different from the previous password and the number of previous passwords specified in the **Minimum number of passwords before reuse** field. If the value is set to zero, the number of different characters is not checked.
6. When you are finished, click **Apply** to save your changes without exiting, or click **OK** to apply your changes and exit, or click **Cancel** to exit this panel without making any changes.

Using the command line:

To perform the same operations using the command line, issue the following command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where <filename> contains:

```
dn: cn=pwdpolicy
changetype: modify
replace: pwdinhistory
pwdinhistory: 10
```

```

-
replace:pwdchecksyntax
pwdchecksyntax: 1
#0=No syntax check
#1=Check syntax (except encrypted)
#2=Check syntax on all
-
replace:pwdminlength
pwdminlength: 8
-
replace:passwordminalphachars
passwordminalphachars: 4
-
replace:passwordminotherchars
passwordminotherchars: 2
-
replace:passwordmaxrepeatedchars
passwordmaxrepeatedchars: 2
-
replace:passwordmindiffchars
passwordmindiffchars: 4

```

Note: Those settings in **bold** type are required for an EAL4 secure password configuration.

Setting Kerberos

The IBM Tivoli Directory server supports Kerberos Version 1.4 servers, such as the IBM Network Authentication Service, for AIX servers and AIX 64-bit clients.

Note: You must have the IBM Network Authentication Service client installed to use Kerberos authentication.

Under Network Authentication Service, a client (generally either a user or a service) sends a request for a ticket to the Key Distribution Center (KDC). The KDC creates a ticket-granting ticket (TGT) for the client, encrypts it using the client's password as the key, and sends the encrypted TGT back to the client. The client then attempts to decrypt the TGT, using its password. If the decryption is successful, the client retains the decrypted TGT, indicating proof of the client's identity.

The TGT, which expires at a specified time, permits the client to obtain additional tickets that give permission for specific services. The requesting and granting of these additional tickets does not require user intervention.

Network Authentication Service negotiates authenticated, optionally encrypted communications between two points on the network. It can enable applications to provide a layer of security that is not dependent on which side of a firewall either client is on. Because of this, Network Authentication Service can play a vital role in the security of your network.

You need to create an LDAP server servicename in the key distribution center (KDC) using the principal name `ldap/<hostname>.<mylocation>.<mycompany>.com`.

Note: An environment variable "LDAP_KRB_SERVICE_NAME" is used to determine the case of the LDAP Kerberos service name. If the variable is set to 'LDAP' then the uppercase LDAP Kerberos service name is used. If the variable is not set, then the lowercase ldap is used. This environment variable is used by both the LDAP client and the server. By default this

variable is not set. See the *IBM Tivoli Directory Server version 6.0 Problem Determination Guide* for more detailed information about the Kerberos service name change.

Network Authentication Service provides the following components:

Key distribution center

The KDC is a trusted server that has access to the private keys of all the principals in a realm. The KDC is composed of two parts: the Authentication Server (AS) and the Ticket Granting Server (TGS). The AS handles initial client authentication by issuing a TGT. The TGS issues service tickets that can be used by the client to authenticate to a service.

Administration server

The administration server provides administrative access to the Network Authentication Service database. This database contains the principals, keys, policies, and other administrative information for the realm. The administration server allows adding, modifying, deleting, and viewing principals and policies.

Password change service

The password change service allows users to change their passwords. The password change service is provided by the administration server.

Client programs

Client programs are provided to manipulate credentials (tickets), manipulate keytab files, change passwords, and perform other basic Network Authentication Service operations.

Application programming interfaces (APIs)

Libraries and header files are provided to allow the development of secure distributed applications. The APIs provided are described in the Application Development Reference.

Using Web Administration:

Under **Server administration** expand the **Manage security properties** category in the navigation area of the Web Administration Tool. If your server supports Kerberos, that is, it has the kerberos supported capabilities OID 1.3.18.0.2.32.30, select the **Kerberos** tab. If your server does not support Kerberos, this tab is not displayed.

1. Select the **Enable Kerberos authentication** check box to enable Kerberos authentication.

Note: You must have a Kerberos client installed to use Kerberos authentication.

2. Select the **Map Kerberos IDs to LDAP DNs** check box to enable the directory administrator to use the existing set of ACL data with the Kerberos authentication method. See “Identity mapping for Kerberos” on page 146 for more information.
3. Enter the Kerberos realm using the format `hostName.domainName`, for example, `TEST.AUSTIN.IBM.COM`. This format is case insensitive.
4. Enter the path and file name of the Kerberos keytab file. This file contains the private key of the LDAP server, as associated with its kerberos account. This file, and the SSL key database file, should be protected.
5. If you are logged in as the directory administrator, enter the Alternate administrator ID using the format `ibm-kn=value@realm` or

ibm-KerberosName=value@realm for example, ibm-kn=root@TEST.AUSTIN.IBM.COM. This field cannot be edited by members of the administrative group.

Note: This ID must be a valid ID in your Kerberos realm. This ID value is case insensitive.

6. When you are finished, click **Apply** to save your changes without exiting, or click **OK** to apply your changes and exit, or click **Cancel** to exit this panel without making any changes.

Using the command line:

To create a Kerberos entry, issue the following command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where <filename> contains:

```
dn: cn=Kerberos, cn=Configuration
cn: Kerberos
ibm-slapdKrbAdminDN: ibm-kn=admin@MYREALM.AUSTIN.IBM.COM
ibm-slapdKrbEnable: true
ibm-slapdKrbIdentityMap: true
ibm-slapdKrbKeyTab: /keytabs/mykeytab.keytab
ibm-slapdKrbRealm: MYREALM.AUSTIN.IBM.COM
objectclass: ibm-slapdKerberos
objectclass: ibm-slapdconfigEntry
objectclass: top
```

To modify a Kerberos entry, for example to change the keytab file, issue the following command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where <filename> contains:

```
dn: cn=Kerberos, cn=Configuration
changetype: modify
replace: ibm-slapdKrbKeyTab
ibm-slapdKrbKeyTab: /keytabs/mynewkeytab.keytab
```

Using Kerberos

Before you can use the command line for Kerberos authentication, you need to do a Kerberos initialization. Issue the following command:

```
kinit <kerberos_principlename>@<realm_name>
```

To use Kerberos authentication you must specify the **-m** option with the GSSAPI parameter on the `idsldapadd` and `idsldapsearch` commands. For example:

```
idsldapsearch -V 3 -m GSSAPI -b <"cn=us"> objectclass=*
```

Identity mapping for Kerberos

Identity mapping enables the directory administrator to use the existing set of ACL data with the Kerberos authentication method. The ACL for the IBM Directory is based on the distinguished name (DN) assigned to the client connected to the directory server. The access rights are based on the permissions granted for that DN and the permissions for any groups containing that DN as a member. If the bind method for GSSAPI is used (that is, Kerberos is used for authenticating to the server), the DN is something like IBM-

KN=your_principal@YOUR_REALM_NAME. This type of DN can be used as

members of access groups or access IDs. You can also use the Kerberos Identity Mapping feature to grant access rights for this DN to an entry already in the directory.

For example, if there is an entry in the directory for Reginald Bender:

```
dn: cn=Reginald Bender, ou=internal users, o=ibm.com, c=US
objectclass: top
objectclass: person
objectclass: organizationalperson
cn: Reginald Bender
sn: Bender
aclentry: access-id:CN=THIS:critical:rWSC
aclentry: group:CN=ANYBODY:normal:rsc
userpassword: cL1eNt
```

The access rights for this entry allow anyone binding with the DN "cn=Reginald Bender, ou=internal users, o=ibm.com, c=US" to view critical data such as the password, but no one else.

If Reginald Bender used Kerberos to bind to the server, his DN could be something like IBM-KN=rbender@SW.REALM_1. If identity mapping is not enabled on the server, he is not allowed to view his own entry's password.

If identity mapping is enabled, he can view the password if this entry were changed to include:

```
dn: cn=Reginald Bender, ou=internal users, o=ibm.com, c=US...
objectclass: ibm-securityidentities
altsecurityidentities: Kerberos:rbender@SW.REALM_1
```

When Reginald Bender binds to the directory server, the server first searches the whole directory to determine if the directory is a KDC (Key Distribution Center) account registry. If it is not, the server searches the directory for any entry containing an altsecurityidentities attribute with a value matching the Kerberos user principal and realm. In this example, rbender is the user principal and SW.REALM_1 is the realm. This is the default for the Kerberos identity mapping. The bind fails if more than one entry has an attribute with this value. The mapping must be one-to-one. If the mapping is successful, Reginald Bender has all of the access rights for "cn=Reginald Bender, ou=internal users, o=ibm.com, c=US", including any access groups that has this as a member.

The IBM Tivoli Directory Server can be used to contain Kerberos account information (krbRealmName-V2 = <realm_name> and krbPrincipalName = <princ_name>@<realm_name>) to serve as the backing store for a KDC.

The server with Kerberos identity mapping enabled first searches the directory for entries with objectclass krbRealm-V2 and krbRealmName-V2 =<realm_name>, such as:

```
dn: krbRealmName-V2=SW.REALM_1, o=ibm.com, c=US
objectclass: krbRealm-V2
krbRealmName-V2: SW.REALM_1
```

If no entries are found, the server uses the default Kerberos identity mapping described previously. If more than one entry is found, the bind fails.

However, if the directory contains the single entry:

```
dn: krbRealmName-V2=SW.REALM_1, ou=Group, o=ibm.com, c=US
objectclass: krbRealm-V2
krbRealmName-V2: SW.REALM_1
krbPrincSubtree: ou=internal users,o=ibm.com, c=US
krbPrincSubtree: ou=external users,o=ibm.com, c=US
```

The server searches each subtree listed as a value of `krbPrincSubtree` for an entry with an attribute `krbPrincipalName`.

In this release, for identity mapping to work for Reginald Bender, you need to add two attributes to the "cn=Reginald Bender, ou=internal users, o=ibm.com, c=US" entry:

```
objectclass: extensibleObject
krbPrincipalName: rbender@SW.REALM_1
```

Depending on whether the directory is a KDC account registry, the final entry is:

```
dn: cn=Reginald Bender, ou=internal users, o=ibm.com, c=US...
objectclass: ibm-securityidentities
altsecurityidentities: Kerberos:rbender@SW.REALM_1...
```

or for a KDC account registry:

```
dn: cn=Reginald Bender, ou=internal users, o=ibm.com, c=US ...
objectclass: extensibleObject
krbPrincipalName: rbender@SW.REALM_1
```

In either case, the client is mapped to "cn=Reginald Bender, ou=internal users, o=ibm.com, c=US".

If a DN is not mapped because no entry is found, the mapping fails but the bind is still successful. However, if more than one DN is mapped, the bind fails.

Identity mapping enables the existing ACLs to work with Kerberos authentication. A client using Kerberos with a mapped identity has two distinct identities, both of which are evaluated in granting access.

Identity mapping has some costs. The internal searches at bind time impact performance and identity mapping requires additional setup to add the appropriate attributes to the entries to be mapped.

In this release, if default identity mapping is used, the administrator (either Kerberos or LDAP) must make sure that the data in the KDC and the data in the LDAP server are synchronized. If the data is not synchronized, incorrect results might be returned because of incorrect ACL evaluation.

Note: The object class, such as **KrbPrincipal** and the attributes such as **KrbPrincSubtree**, **KRbAliasedObjectName**, and **KrbHintAliases** are used to define a IBM Directory as a Kerberos KDC. See the Kerberos documentation for more information.

Certificate revocation verification

If you have selected to use server and client authentication in your SSL settings, you might want to configure your server to check for revoked or expired certificates.

When a client sends an authenticated request to a server, the server reads the certificate and sends a query to an LDAP server with a list that contains revoked

certificates. If the client certificate is not found in the list, communications between the client and server are allowed over SSL. If the certificate is found, communications are not allowed.

To configure SSL certificate revocation verification use one of the following methods:

Using Web Administration:

Under **Server administration**, expand the **Manage security properties** category in the navigation area of the Web Administration Tool, select the **Certificate revocation** tab.

1. Enter the name of the server that contains certificates that have been revoked. This server is designated by the certificate granting authority (CA) that you use, for example VeriSign. The format of the host name is `hostName.domainName`, for example, `myserver.ibm.com`.
2. Enter the port used to communicate with the server, for example 389.
3. Enter the DN used to bind to the verifying server, for example `cn=root`. This is optional if the verifying server allows anonymous searches for certificate revocation lists (CRLs).
4. Enter the password associated with the bind DN. This is required if you specified a DN.
5. Type the bind password again to confirm that there are no typographical errors.
6. When you are finished, click **Apply** to save your changes without exiting, or click **OK** to apply your changes and exit, or click **Cancel** to exit this panel without making any changes.

Note: Expired certificates are not included in the list because the expiration date is contained in the certificate itself.

Using the command line:

To use the command line to configure for SSL certificate revocation verification, issue the command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where *<filename>* contains:

```
dn: cn=CRL,cn=SSL,cn=Configuration
changetype: modify
replace: ibm-slapdCr1Host
ibm-slapdCr1Host: <newhostname>
-
replace: ibm-slapdCr1Password
ibm-slapdCr1Password: <password>
-
replace: ibm-slapdCr1Port
ibm-slapdCr1Port: <portnumber>
-
replace: ibm-slapdCr1User
ibm-slapdCr1User: <username>
```

You must restart the server and the administration daemon for the changes to take effect.

Configuring the DIGEST-MD5 mechanism

DIGEST-MD5 is a SASL authentication mechanism. When a client uses Digest-MD5, the password is not transmitted in clear text and the protocol prevents replay attacks.

To configure the DIGEST-MD5 mechanism use one of the following methods.

Using Web Administration:

Under **Server administration**, expand the **Manage security properties** category in the navigation area of the Web Administration Tool, select the **DIGEST-MD5** tab.

Note: This tab is displayed only if DIGEST-MD5 is supported on your server.

1. Under **Server realm**, you can use the preselected **Default** setting, which is the fully qualified host name of the server, or you can click **Realm** and type the name of the realm that you want to configure the server as.

Note: If the `ibm-slapdDigestRealm` attribute in the configuration entry is set, the server uses that value instead of the default for the realm. In this case, the **Realm** button is preselected and the realm value is displayed in the field.

This realm name is used by the client to determine which user name and password to use.

When using replication, you want to have all the servers configured with the same realm.

2. Under **Username attribute**, you can use the preselected **Default** setting, which is `uid`, or you can click **Attribute** and type the name of the attribute that you want the server to use to uniquely identify the user entry during DIGEST-MD5 SASL binds.

Note: If the `ibm-slapdDigestAttr` attribute in the configuration entry is set, the server uses that value instead of the default for the Username attribute. In this case, the **Attribute** button is preselected and the attribute value is displayed in the field.

3. If you are logged in as the directory administrator, under **Administrator username**, type the administrator user name. This field cannot be edited by members of the administrative group. If the user name specified on a DIGEST-MD5 SASL bind matches this string, the user is the administrator.

Note: The administrator user name is case sensitive.

4. When you are finished, click **Apply** to save your changes without exiting, or click **OK** to apply your changes and exit, or click **Cancel** to exit this panel without making any changes.

Using the command line:

To create the `cn=Digest,cn=configuration` entry, enter the command:

```
idsldapadd -D <adminDN> -w <adminpw> -i <filename>
```

where `<filename>` contains:

```
dn: cn=Digest,cn=configuration
cn: Digest
ibm-slapdDigestRealm: <realm name>
ibm-slapdDigestAttr: <uid>
```

```
ibm-slapedDigestAdminUser: <Adminuser>
objectclass:top
objectclass: ibm-slapedConfigEntry
objectclass: ibm-slapedDigest
```

To change the settings for DIGEST-MD5, issue the following command:

```
idsldapmodify -D <adminDN> -w <adminpw> -i <filename>
```

where <filename> contains:

```
dn: cn=Digest,cn=configuration
changetype: modify
replace: ibm-slapedDigestRealm
ibm-slapedDigestRealm: <newrealmname>
-
replace: ibm-slapedDigestAttr
ibm-slapedDigestAttr: <newattribute>
-
replace: ibm-slapedDigestAdminUser
ibm-slapedDigestAdminUser: <newAdminuser>
```

Chapter 12. Referrals

Referrals provide a way for servers to refer clients to additional directory servers. A referral specifies the URL of an alternate LDAP server. This alternate server handles any requests for objects that are not found within any of the subtrees of the current LDAP server.

A default referral can be used to point to:

- The immediate parent of this server (in a hierarchy)
- A "more knowledgeable" server, such as the uppermost server in the hierarchy
- A "more knowledgeable" server that possibly serves a disjoint portion of the namespace

With referrals you can:

- Distribute namespace information among multiple servers
- Provide knowledge of where data is located within a set of interrelated servers
- Route client requests to the appropriate server

Note: All supported servers and clients for IBM Tivoli Directory Server version 6.0 are enabled to support IPv6 and IPv4 formats. See Appendix E, "IPv6 support," on page 489 for information about these two formats.

Some of the advantages of using referrals are the ability to:

- Distribute processing overhead, providing primitive load balancing
- Distribute administration of data along organizational boundaries
- Provide potential for widespread interconnection, beyond an organization's own boundaries

Note: On the Linux, Solaris, and HP-UX platforms, if a client hangs while chasing referrals, ensure that the environment variable `LDAP_LOCK_REC` has been set in your system environment. No specific value is required.

```
set LDAP_LOCK_REC=anyvalue
```

Setting up referrals to other LDAP directories

This section describes how to use the referral object class and the `ref` attribute to construct entries in an LDAP directory containing references to other LDAP directories. This section also describes how to associate multiple servers using referrals and provides an example.

Using the referral object class and the `ref` attribute

The referral object class and the `ref` attribute are used to facilitate distributed name resolution or to search across multiple servers. The `ref` attribute appears in an entry named in the referencing server. The value of the `ref` attribute points to an entry maintained in the referenced server.

Creating entries

Following is an example configuration that illustrates the use of the `ref` attribute.

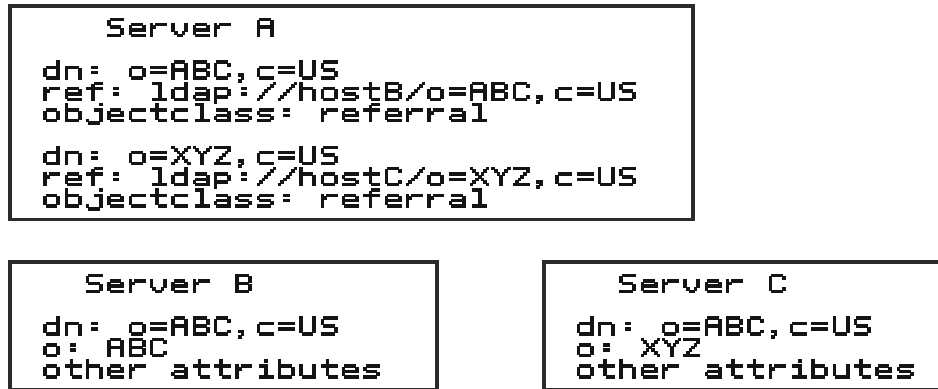


Figure 1. Example of using the referral attribute

In the example, Server A holds references to two entries: o=ABC, c=US and o=XYZ, c=US. For the o=ABC, c=US entry, Server A holds a reference to Server B and for the o=XYZ, c=US entry, Server A holds a reference to Server C.

One setup of referrals is to structure the servers into a hierarchy based on the subtrees they manage. Then, provide "forward" referrals from servers that hold higher (closer to the root of the hierarchy) information and set the default referral to point back to its parent server.

Associating servers with referrals

To associate servers through referrals:

- Use referral objects to point to other servers for subordinate references.
- Define the default referral to point somewhere else, typically to the parent server.

Note: Referral objects can be seen from command line LDAP utilities by specifying the **-M** option.

Pointing to other servers: Use referral objects to point to the other servers for subordinate references, that is, portions of the namespace below this server that it does not service directly.

Referral objects, like other objects, go in the backend (DB2). Referral objects consist of:

dn: Specifies the distinguished name. It is the portion of the namespace served by the referenced server.

objectclass: Specifies the value of the objectclass "referral".

ref: Specifies the LDAP Web address of the server. This Web address consists of the ldap:// identifier, the hostname:port, and a DN. The identifier can be either a host name string or a TCP/IP address. The DN requires a slash (/) before it to delimit it from the hostname:port, and should match the DN of the referral object. The DN specified in the value of the referral attribute should match the DN of the referral object. Typically, it is an entry in a naming context at or below the naming context held by the referencing server.


```
dn: o=IBM,c=US
objectclass: referral
ref: ldap://9.130.25.51:389/o=IBM,c=US
```

Binding with a distributed namespace

When performing searches, the same DN that was used to bind or log in to the original server is used to bind to the referred-to server, unless the IBM Directory application is designed to modify the bind DN and credentials. The correct access must be set up for the same DN to be able to bind to both servers for chasing the referrals. See “Logging on to the console as the server administrator, a member of an administrative group or an LDAP user” on page 22 for additional information.

An example of distributing the namespace through referrals

Following are the steps involved in distributing the namespace using referrals.

1. Plan your namespace hierarchy.
 - country - US
 - company - IBM, Lotus
 - organizationalUnit - IBM Austin, IBM Endicott, IBM HQ
2. Set up multiple servers, each containing portions of the namespace.

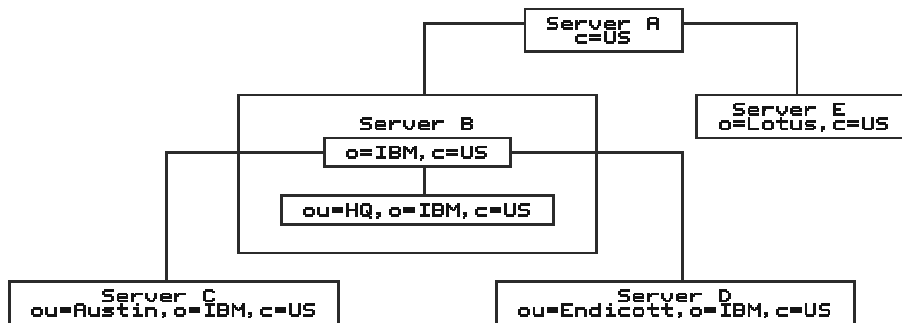


Figure 2. Setting up the servers

Server descriptions:

Server A

A server used to locate other servers in the U.S. With no other knowledge, clients can come here first to locate information for anyone in the U.S.

Server B

A hub for all data pertaining to IBM in the U.S. Holds all HQ information directly. Holds all knowledge (referrals) of where other IBM data is located.

Server C

Holds all IBM Austin information.

Server D

Holds all IBM Endicott information.

Server E

Holds all Lotus® information.

3. Set up referral objects to point to the descendants in other servers.

<pre>dn: o=IBM,c=US objectClass: referral ref: ldap://ibm.com:389/o=IBM,c=US</pre>	←→Pointer to Server B
<pre>dn: o=Lotus,c=US objectClass: referral ref: ldap://lotus.com:389/o=Lotus,c=US</pre>	←→Pointer to Server E

Figure 3. Server A database (LDIF input)

Servers can also define a default referral, which is used to point to a "more knowledgeable" server for anything that is not underneath them in the namespace.

Note: The default referral LDAP Web address does not include the DN portion.

Following is an arrangement of the same five servers, showing the referral objects in the database as well as the default referrals that are used for superior references.



Figure 4. Referral example summary

Creating default referrals

Using the Web Administration Tool is the recommended method to create and remove default referrals.

Using Web Administration:

1. If you have not already done so, use the Web Administration Tool to log on to the master server.

2. Expand the **Server Administration** category in the navigation area of the Web Administration Tool, select **Manage server properties**.
3. Click **Referrals**.

Note: If you are working in another panel and are adding or modifying an entry that has an attribute that contains referrals you can click **Manage referrals** to access this panel.

4. Click **Add**.
5. Enter the host name and port for the server to which this referral value is pointing.
6. Enter the base DN in the directory information tree in the target server. For example `ou=austin,o=ibm,c=us`.
7. Select the attributes you want to include in the referral URL and click **Add**. To remove an attribute from the referral URL, highlight the attribute in the **Selected attributes** field and click **Remove**.
8. Select the scope for the referral search.
 - Select **Object** to search only within the selected object.
 - Select **Single level** to search only within the immediate children of the selected object.
 - Select **Subtree** to search all descendants of the selected entry.
9. Specify a search filter. See “Search filters” on page 305 for more information.
10. Select **Enable SSL**, if the referral is to a secure (SSL) server.
11. Click **OK**.
12. Repeat these steps for additional referrals.
13. When you are finished, click **Apply** to save your changes without exiting, or click **OK** to apply your changes and exit, or click **Cancel** to exit this panel without making any changes.

You must restart the server for the changes to take effect.

Using the command line:

Define a default referral to reference a directory on another server.

Note: The default referral LDAP URL does not include the DN portion. It includes only the `ldap://` identifier and the `hostname:port`.

For example:

Note: This example is of a local LDAP server on port 389.

```
idsldapadd -D <adminDN> -w <adminpw> -i <filename>
```

where `<filename>` contains:

```
# referral
dn: cn=Referral, cn=Configuration
cn: Referral
ibm-slapdReferral: ldap://<additional hostname:port>/<baseDN>?<attributes>?
<scope>?<filter>
ibm-slapdReferral: ldap://<additional hostname:port>/<baseDN>?<attributes>?
<scope>?<filter>
ibm-slapdReferral: ldap://<additional hostname:port>/<baseDN>?<attributes>?
<scope>?<filter>
objectclass: ibm-slapdReferral
objectclass: top
objectclass: ibm-slapdConfigEntry
```

For example, to set up referrals to two servers, server1 and server2 (a secure server), listening on port **389**, with a base of **ou=austin,o=ibm,c=us** , with the attributes **cn**, **sn**, and **description**, a scope of **base**, and a filter of **objectclass=***, the LDIF file is :

```
# referral
dn: cn=Referral, cn=Configuration
cn: Referral
ibm-slapdreferral: ldap://server1.mycity.mycompany.com:389/
    ou=austin,o=ibm,c=us?cn,sn,description?base?objectclass=*
ibm-slapdreferral: ldaps://server2.mycity.mycompany.com:389/
    ou=austin,o=ibm,c=us?cn,sn,description?base?objectclass=*
objectclass: ibm-slapdReferral
objectclass: ibm-slapdConfigEntry
objectclass: top
```

See Appendix E, “IPv6 support,” on page 489 for more information about supported URL formats.

Modifying referrals

To edit a referral, use one of the following methods.

Using Web Administration:

1. If you have not already done so, use the Web Administration Tool to log on to the master server.
2. Expand the **Server Administration** category in the navigation area of the Web Administration Tool, select **Manage server properties**.
3. Click **Referrals**.

Note: If you are working in another panel and are adding or modifying an entry that has an attribute that contains referrals you can click **Manage referrals** to access this panel.

4. From the **Current referrals** section, select the referral you want to edit.
5. Click **Edit**.
6. You can modify the host name and port for the server to which this referral value is pointing.
7. You can modify the base DN in the directory information tree in the target server. For example **ou=austin,o=ibm,c=us**.
8. You can modify the attributes you want to include in the referral URL by adding or removing attributes from the referral URL.
9. You can modify the scope for the referral search.
10. You can modify the search filter. See “Search filters” on page 305 for more information.
11. You can modify **Enable SSL**, if the referral is to a secure (SSL) server or not.
12. Click **OK**.
13. Repeat these steps for each referral you want to modify.
14. When you are finished, click **Apply** to save your changes without exiting, or click **OK** to apply your changes and exit, or click **Cancel** to exit this panel without making any changes.

You must restart the server for the changes to take effect.

Using the command line:

To modify the referral to server1 in order to change the baseDN to ou=raleigh,o=ibm,c=us, issue the command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where <filename> contains:

```
dn: cn=referral, cn= configuration
changetype: modify
replace: ibm-slapdReferral
ibm-slapdreferral: ldap://server1.mycity.mycompany.com:389/
ou=raleigh,o=ibm,c=us?cn,sn,description?base?objectclass=*
```

Removing referrals

To remove a referral, use one of the following methods.

Using Web Administration:

1. If you have not already done so, use the Web Administration Tool to log on to the master server.
2. Expand the **Server Administration** category in the navigation area of the Web Administration Tool, select **Manage server properties**.
3. Click **Referrals**.

Note: If you are working in another panel and are adding or modifying an entry that has an attribute that contains referrals you can click **Manage referrals** to access this panel.

4. From the **Current referrals** section, select the referral you want to remove.
5. Click **Remove**.
6. A confirmation panel is displayed. Click **OK** to remove the referral or click **Cancel** to return to the previous panel without making any changes.
7. Repeat this process for as many referrals as you want to remove or click **Remove all** to remove all of the current referrals.
8. When you are finished, click **Apply** to save your changes without exiting, or click **OK** to apply your changes and exit, or click **Cancel** to exit this panel without making any changes.

You must restart the server for the changes to take effect.

Using the command line:

To delete a single default referral, for example, austin.ibm.com:389, issue the command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where <filename> contains:

```
dn: cn=referral, cn= configuration
changetype: modify
delete: ibm-slapdReferral
ibm-slapdReferral: ldap://referral.austin.ibm.com:389
```

To delete all default referrals:

```
idsldapdelete -D <adminDN> -w <adminPW> "cn=referral,cn=configuration"
```

Chapter 13. Replication

Replication is a technique used by directory servers to improve performance, availability, and reliability. The replication process keeps the data in multiple directory servers synchronized.

Replication provides three main benefits:

- Redundancy of information - Replicas back up the content of their supplier servers.
- Faster searches - Search requests can be spread among several different servers, instead of a single server. This improves the response time for the request completion.
- Security and content filtering - Replicas can contain subsets of the data in a supplier server.

Replication terminology

Cascading replication

A replication topology in which there are multiple tiers of servers. A peer/master server replicates to a set of read-only (forwarding) servers, which in turn replicate to other servers. Such a topology off-loads replication work from the master servers.

Consumer server

A server that receives changes through replication from another (supplier) server.

Credentials

Identify the method and required information that the supplier uses in binding to the consumer. For simple binds, this includes the DN and password. The credentials are stored in an entry the DN of which is specified in the replica agreement.

Forwarding server

A read-only server that replicates all changes sent to it. This contrasts with a peer/master server in that it is read only and it can have no peers.

Gateway server

A server that forwards all replication traffic from the local replication site where it resides to other Gateway servers in the replicating network. Also receives replication traffic from other Gateway servers within the replication network, which it forwards to all servers on its local replication site.

Gateway servers must be masters (writable).

Master server

A server that is writable (can be updated) for a given subtree.

Nested subtree

A subtree within a replicated subtree of the directory.

Peer server

The term used for a master server when there are multiple masters for a

given subtree. A peer server does not replicate changes sent to it from another peer server; it only replicates changes that are originally made on it.

Replica group

The first entry created under a replication context has objectclass `ibm-replicaGroup` and represents a collection of servers participating in replication. It provides a convenient location to set ACL's to protect the replication topology information. The administration tools currently support one replica group under each replication context, named **`ibm-replicagroup=default`**.

Replica subentry

Below a replica group entry, one or more entries with objectclass `ibm-replicaSubentry` can be created; one for each server participating in replication as a supplier. The replica subentry identifies the role the server plays in replication: master or read-only. A read-only server might, in turn, have replication agreements to support cascading replication.

Replicated subtree

A portion of the directory information tree (DIT) that is replicated from one server to another. Under this design, a given subtree can be replicated to some servers and not to others. A subtree can be writable on a given server, while other subtrees may be read-only.

Replicating network

A network that contains connected replication sites.

Replication agreement

Information contained in the directory that defines the 'connection' or 'replication path' between two servers. One server is called the supplier (the one that sends the changes) and the other is the consumer (the one that receives the changes). The agreement contains all the information needed for making a connection from the supplier to the consumer and scheduling replication.

Replication context

Identifies the root of a replicated subtree. The `ibm-replicationContext` auxiliary object class may be added to an entry to mark it as the root of a replicated area. The configuration information related to replication is maintained in a set of entries created below the base of a replication context.

Replication site

A Gateway server and any master, peer, or replica servers configured to replicate together.

Schedule

Replication can be scheduled to occur at particular times, with changes on the supplier accumulated and sent in a batch. The replica agreement contains the DN for the entry that supplies the schedule.

Supplier server

A server that sends changes to another (consumer) server.

Replication topology

The set of objects in a directory that control what kind of information is replicated between LDAP servers and how it is replicated. These objects include:

- Replication contexts

- Replication groups
- Replication subentries
- Replication agreements
- Replication credentials
- Replication schedule entries

All LDAP servers in the replicating network should have the same replication topology.

Replication topology

Specific entries in the directory are identified as the roots of replicated subtrees, by adding the `ibm-replicationContext` objectclass to them. Each subtree is replicated independently. The subtree continues down through the Directory Information Tree (DIT) until reaching the leaf entries or other replicated subtrees (context). Entries are added below the root of the replicated subtree to contain the replication configuration information. These entries are one or more replica group entries, under which are created replica subentries. Associated with each replica subentry are replication agreements that identify the servers that are supplied (replicated to) by each server, as well as defining the credentials and schedule information.

Through replication, a change made to one directory is propagated to one or more additional directories. In effect, a change to one directory shows up on multiple different directories. The IBM Tivoli Directory supports an expanded master-replica replication model. Replication topologies are expanded to include:





- Replication of subtrees of the Directory Information Tree to specific servers
- A multi-tier topology referred to as cascading replication
- Assignment of server role (supplier or consumer) by subtree.
- Multiple master servers, referred to as peer to peer replication.
- Gateway servers that replicate across networks.

The advantage of replicating by subtrees is that a replica does not need to replicate the entire directory. It can be a replica of a part, or subtree, of the directory.

The expanded model changes the concept of master and replica. These terms no longer apply to servers, but rather to the roles that a server has regarding a particular replicated subtree. A server can act as a master for some subtrees and as a replica for others. The term, *master*, is used for a server that accepts client updates for a replicated subtree. The term, *replica*, is used for a server that only accepts updates from other servers designated as a supplier for the replicated subtree.

There are four types of directory roles as defined by function: *master/peer*, *gateway*, *forwarding (cascading)*, and *replica (read-only)*.

Table 14. Server roles

<p>Master/peer </p>	<p>The master/peer server contains the master directory information from where updates are propagated to the replicas. All changes are made and occur on the master server, and the master is responsible for propagating these changes to the replicas.</p> <p>There can be several servers acting as masters for directory information, with each master responsible for updating other master servers and replica servers. This is referred to as peer replication. Peer replication can improve performance and reliability. Performance is improved by providing a local server to handle updates in a widely distributed network. Reliability is improved by providing a backup master server ready to take over immediately if the primary master fails.</p> <p>Notes:</p> <ol style="list-style-type: none"> 1. Master servers replicate all client updates, but do not replicate updates received from other masters. 2. Updates among peer servers can be immediate or scheduled. See “Creating replication schedules” on page 238 for more information.
<p>Forwarding (Cascading) </p>	<p>A forwarding or cascading server is a replica server that replicates all changes sent to it. This contrasts to a master/peer server in that a master/peer server only replicates changes that are made by clients connected to that server. A cascading server can relieve the replication workload from the master servers in a network which contains many widely dispersed replicas.</p>
<p>Gateway </p>	<p>Gateway replication uses Gateway servers to collect and distribute replication information effectively across a replicating network. The primary benefit of Gateway replication is the reduction of network traffic.</p>
<p>Replica (read-only) </p>	<p>An additional server that contains a copy of directory information. The replicas are copies of the master (or the subtree that it is a replica of). The replica provides a backup of the replicated subtree.</p>

You can request updates on a replica server, but the update is actually forwarded to the master server by returning a referral to the client. If the update is successful, the master server then sends the update to the replicas. Until the master has completed replication of the update, the change is not reflected on the replica server where it was originally requested. If the replication fails, it is repeated even if the master is restarted. Changes are replicated in the order in which they are made on the master. See “Replication error handling” on page 168.

If you are no longer using a replica, you must remove the replica agreement from the supplier. Leaving the definition causes the server to queue up all updates and use unnecessary directory space. Also, the supplier continues trying to contact the missing consumer to retry sending the data.

Overview of replication

This section presents a high-level description of the various types of replication topologies.

Simple replication

The basic relationship in replication is that of a master server and its replica server. The master server can contain a directory or a subtree of a directory. The master is writable, which means it can receive updates from clients for a given subtree. The

replica server contains a copy of the directory or a copy of part of the directory of the master server. The replica is read only; it cannot be directly updated by clients. Instead it refers client requests to the master server, which performs the updates and then replicates them to the replica server.

A master server can have several replicas. Each replica can contain a copy of the master's entire directory, or a subtree of the directory. In the following example Replica 2 contains a copy of the complete directory of the Master Server, Replica 1 and Replica 3 each contain a copy of a subtree of the Master Server's directory.

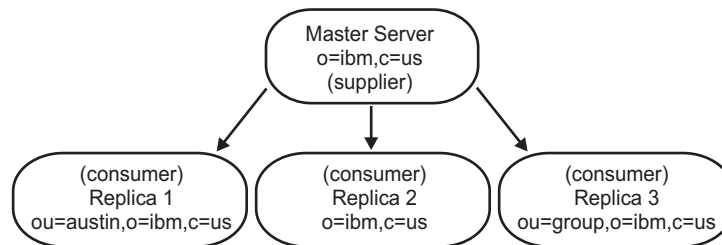


Figure 5. Master-replica replication

The relationship between two servers can also be described in terms of roles, either supplier or consumer. In the previous example the Master Server is a supplier to each of the replicas. Each replica in turn is a consumer of the Master Server.

Cascading replication

Cascading replication is a topology that has multiple tiers of servers. A master server replicates to a set of read-only (forwarding) servers that in turn replicate to other servers. Such a topology off-loads replication work from the master server. In the example of this type of topology, the master server is a supplier to the two forwarding servers. The forwarding servers serve two roles. They are consumers of the master server and suppliers to the replica servers associated with them. The replica servers are consumers of their respective forwarding servers. For example:

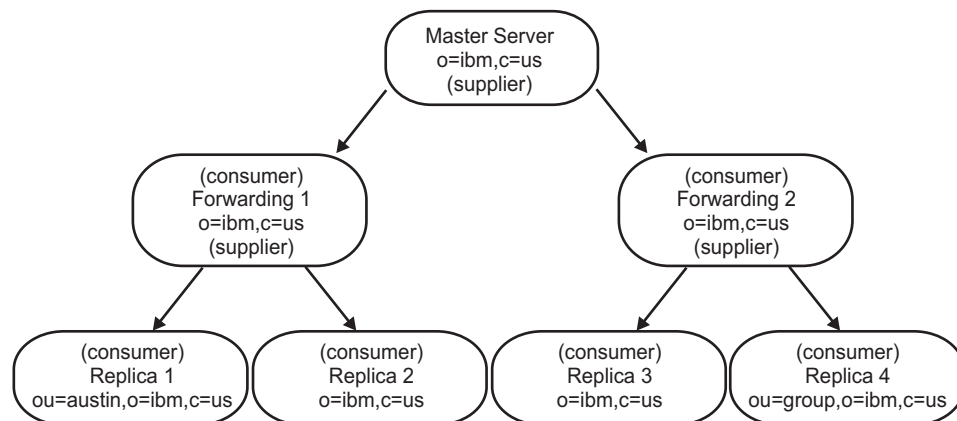


Figure 6. Cascading replication

Peer-to-peer replication

There can be several servers acting as masters for directory information, with each master responsible for updating other master servers and replica servers. This is referred to as peer replication. Peer replication can improve performance, availability, and reliability. Performance is improved by providing a local server to

handle updates in a widely distributed network. Availability and reliability are improved by providing a backup master server ready to take over immediately if the primary master fails. Peer master servers replicate all client updates to the replicas and to the other peer masters, but do not replicate updates received from other master servers.

Note: Conflict resolution for add and modify operations in peer-to-peer replication is based on Timestamp. See “Replication conflict resolution” on page 167.

The following is an example of peer-to-peer replication:

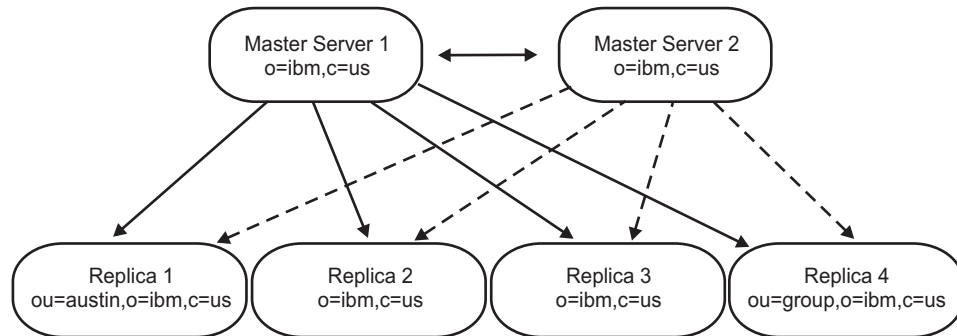


Figure 7. Peer-to-peer replication

Gateway replication

Gateway replication is a more complex adaptation of peer-to-peer replication that extends replication capabilities across networks. The most notable difference is that a gateway server does replicate changes received from other peer servers through the gateway.

A gateway server must be a master server, that is, writable. It acts as a peer server within its own replication site. That is, it can receive and replicate client updates and receive updates from the other peer-master servers within the replication site. It does not replicate the updates received from the other peer-masters to any servers within its own site.

Within the gateway network, the gateway server acts as a two-way forwarding server. In one instance, the peers in its replication site act as the suppliers to the gateway server and the other gateway servers are its consumers. In the other instance the situation is reversed. The other gateway servers act as suppliers to the gateway server and the other servers within its own replication site are the consumers.

Gateway replication uses gateway servers to collect and distribute replication information effectively across a replicating network. The primary benefit of gateway replication is the reduction of network traffic. For example:

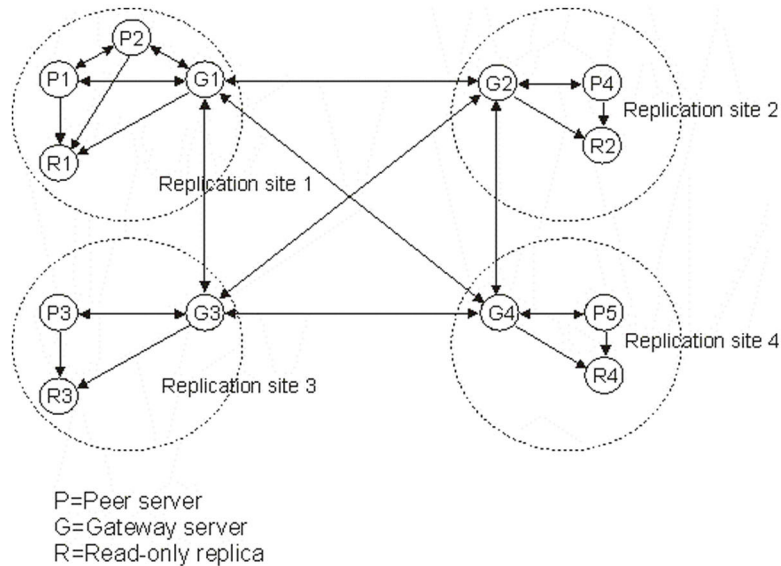


Figure 8. Gateway replication

Replication conflict resolution

If replication conflicts occur involving delete or modifyDN operations, errors that require human intervention might result. For example, if an entry is renamed on one server while it is being modified on a second server, the rename (modifyDN) might arrive at a replica before the modify. Then when the modify arrives, it fails. In this case, the administrator needs to respond to the error by applying the modify to the entry with the new DN. All information necessary to redo the modify with the correct name is preserved in the replication and error logs. Replication errors are rare occurrences in a correctly configured replication topology, but it is not safe to assume that they never occur.

Conflict resolution for add and modify operations in peer-to-peer replication is based on Timestamp. The entry update with the most recent modify TimeStamp on any server in a multi-master replication environment is the one that takes precedence. Replicated delete and rename request are accepted in the order received without conflict resolution. When a replication conflict is detected the replaced entry is archived for recovery purposes in the Lost and Found log. See Chapter 15, "Logging Utilities," on page 267 for more information.

Updates to the same entry made by multiple servers might cause inconsistencies in directory data because conflict resolution is based on the TimeStamp of the entries. The most recent modify TimeStamp takes precedence. If the data on your servers become inconsistent, see "idsldapdiff, ldapdiff" on page 377 for information on resynchronizing servers.

For IBM Tivoli Directory Server 6.0 to resolve replication conflict, it needs the supplier to provide the entry's timestamp before the entry was updated on the supplier. For a previous version of the server, such as IBM Directory Server 5.1 or IBM Tivoli Directory Server 5.2, the server does not have the capability to supply this kind of information. Therefore, replication conflict resolution is not applicable

to cases where the supplier is a downlevel server. The IBM Tivoli Directory Server 6.0 consumer server, in this case, takes the replicated timestamp and update and applies it without conflict checking.

Notes:

1. Earlier versions of the IBM Tivoli Directory Server do not support time stamp conflict resolution. If your topology contains earlier versions of the IBM Tivoli Directory Server, data consistency is not ensured for the network. See Appendix B, "Object Identifiers (OIDs) and attributes in the root DSE," on page 473 and "OIDs for supported and enabled capabilities" on page 475 to find out how to determine, if your servers support conflict resolution.
2. When the `IBMSLAPD_REPL_NO_CONFLICT_RESOLUTION` environment variable is defined, no conflict resolution takes place on a server. If the variable is defined before a server is started, the server operates in a "no replication conflict resolution" mode. In this mode, the server does not try to compare entries' timestamps for replicated entries in an attempt to resolve conflicts between the entries. This environment variable is checked during server startup and therefore changing it while server is running will not have any effect to the server.
3. To resolve replication conflict, a regular database entry which has a later timestamp is not replaced by a replicated entry which has an earlier timestamp. However, conflict resolution does not apply to entry `cn=schema` which is always replaced by a replicated `cn=schema`.

Setting up a load balancer is one method of resolving data conflict resolution.

A load balancer, such as the IBM WebSphere Edge Server, has a virtual host name that applications use when sending updates to the directory. The load balancer is configured to send those updates to only one server. If that server is down, or unavailable because of a network failure, the load balancer sends the updates to the next available peer server until the first server is back on line and available. Refer to your load balancer product documentation for information on how to install and configure the load balancing server.

Replication error handling

Replication errors are any replicated updates for which the consumer returns a result other than `LDAP_SUCCESS`. Replication conflict errors return `LDAP_OTHER` and a special control, and are not treated as errors unless the data is greater than allowed by the server configuration.

Replication errors can be logged in a new table, `REPLERROR`. This replaces non-blocking replication. The size of the replication error log is in the server configuration (`ibm-slapdReplMaxErrors`) and can be updated dynamically. Replication errors are stored and managed per replication agreement, that is, if there are two agreements, then one agreement might have some errors logged, and the other agreement might have no errors logged.

How errors are addressed depends on the replication method. For single-threaded replication, the following occurs:

- `ibm-slapdReplMaxErrors: 0` means that no errors are logged and the first error is retried every minute until it succeeds or is skipped.
- If the number of errors for an agreement reaches the limit, the next error is retried until the error succeeds, is skipped, the number of errors for an

agreement limit is increased, or an error is cleared from the log. The data for an entry that is being retried is displayed by the replication status attribute `ibm-replicationChangeLDIF`.

- The status for the replication agreement is:

```
ibm-replicationStatus: retrying
```

For multi-threaded replication, the following occurs:

- `ibm-slapdReplMaxErrors: 0` means that no errors should be logged, but any errors are logged and replication is suspended until all of the errors are cleared.
- If the number of errors for an agreement exceeds the limit, replication is suspended until at least one error is cleared, or the number of errors for an agreement limit is increased.
- The status for the replication agreement is:

```
ibm-replicationStatus: error log full
```

For more information about viewing replication errors, see the *IBM Tivoli Directory Server Version 6.0 Problem Determination Guide*.

Replication agreements

A replication agreement is an entry in the directory with the object class **`ibm-replicationAgreement`** created beneath a replica subentry to define replication from the server represented by the subentry to another server. These objects are similar to the `replicaObject` entries used by prior versions of the IBM Tivoli Directory Server. The replication agreement consists of the following items:

- A user friendly name, used as the naming attribute for the agreement.
- An LDAP URL specifying the server, port number, and whether SSL should be used.
- The consumer server ID, if known -- 'unknown' for a server whose server ID is not known (if a server is running a release earlier than IBM Tivoli Directory Server version 5.2, or if the server ID could not be retrieved while setting up the topology).
- The DN of an object containing the credentials used by the supplier to bind to the consumer.
- An optional DN pointer to an object containing the schedule information for replication. If the attribute is not present, changes are replicated immediately.
- Replication method (single threaded or multi-threaded).
- Number of consumer: For a replication agreement using the single-threaded replication method, the number of consumer connections is always one, the attribute value is ignored. For an agreement using multi-threaded replication, the number of connections can be configured from 1 to 32. If no value is specified on the agreement, the number of consumer connections is set to one.

Note: For the `cn=ibmpolicies` subtree, all replication agreements will use the single-threaded replication method and one consumer connection, ignoring the attribute values.

The user friendly name might be the consumer server name or some other descriptive string.

To aid in enforcing the accuracy of the data, when the supplier binds to the consumer, it retrieves the server ID from the root DSE and compares it to the value in the agreement. A warning is logged if the server IDs do not match.

The consumer server ID is used by the Web Administration Tool to traverse the topology. Given the consumer's server ID, the Web Administration Tool can find the corresponding subentry and its agreements.

Because the replication agreement can be replicated, a DN to a credentials object is used. This allows the credentials to be stored in a nonreplicated area of the directory. Replicating the credentials objects (from which 'clear text' credentials must be obtainable) represents a potential security exposure. The `cn=localhost` suffix is an appropriate default location for creating credentials objects. Use of a separate object also makes it easier to support various authentication methods; new object classes can be created rather than trying to make sense of numerous optional attributes.

Object classes are defined for each of the supported authentication methods:

- Simple bind
- SASL EXTERNAL mechanism with SSL
- Kerberos authentication

You can designate that part of a replicated subtree not be replicated by adding the `ibm-replicationContext` auxiliary class to the root of the subtree, without defining any replica subentries.

The following sections are examples of setting up replication using either the Web Administration Tool or the command line utilities, and an LDIF file. The scenarios are of increasing complexity:

- One master and one replica
- One master, one forwarder, and one replica
- Two peer/masters, two forwarders, and four replicas.
- Gateway replication.

Things to consider before configuring replication

Before setting up an LDAP replication configuration, there are some administrative responsibilities that you need to consider. In order to ensure that replication is operating smoothly and that your replicas are staying up-to-date, the administrator needs to take some periodic actions to monitor the replication status. After replication is correctly configured, it continues to automatically propagate updates to all defined replica servers. However, if errors occur, human intervention might be required to fully correct the problem.

Interfaces are provided to allow you to view information about updates queued for replication, and to take actions like suspending or resuming replication to a specific replica. See "Managing queues" on page 240 for details. These replication queues should be checked periodically for errors. Read "Viewing server errors" on page 235 to understand how to check for errors that may have occurred during replication to a specified consumer server.

Detailed status and error information is also available to the administrator by reading operational attributes on the replication agreements. See "Monitoring replication status" on page 243 for a description of the information available.

Configuring multiple master servers adds to the potential error cases that an administrator must be aware of. If the same entry is updated at two different master servers at approximately the same time, those updates are likely to conflict

when they are replicated to other servers in the topology. The replication algorithm is designed to detect and resolve any replication conflicts between adds or modifies. See “Replication conflict resolution” on page 167.

You can use a time synchronization product to keep your LDAP servers synchronized. Such a utility is not provided by IBM Tivoli Directory Server.

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 I, “Synchronizing two-way cryptography between server instances,” on page 537 for information about synchronizing directory server instances.

The server does not allow subtree deletion if the subtree contains replication agreements. Because the order of entries to be deleted is not fixed, deleted entries can be replicated randomly. For example, if a replication agreement is deleted first in a subtree, then the delete operation cannot be replicated. This restriction works only when a context is deleted with the **-s** option. If you want to delete the subtree, you must first delete the replication agreements.

Note: You must synchronize the replication topology entries before starting replication. Set up the servers in the network.

Replicating schema and password policy updates

Schema and password policy updates are only updated using the `cn=ibmpolicies` subtree. To ensure that the schemas and password policy is synchronized across all of your servers, you must create an additional replication context for `cn=ibmpolicies`. This replication context needs to include all the servers that are in your directory topology.

Note: If you are using a proxy server, password policy updates are replicated. Schema changes, however, are not. To ensure that schemas are kept synchronized in a proxy environment, the schema update needs to be made on each proxy server and also on a master server in the `cn=ibmpolicies` directory subtree.

Consider the following with respect to replication:

- For best results, replicate changes to the schema before replicating data changes.
- You can use the **idsldapdiff** utility to identify differences in schema, but the **idsldapdiff** utility cannot automatically correct differences in schema.
- You can synchronize schemas by copying the schemas between replicas.

Creating a master-replica topology

Note: Before setting up your replication topology, make a backup copy of your original `ibmslapd.conf`, `ibmslapdcfg.ksf`, and `ibmslapddir.ksf` files. You can use this backup copy to restore your original configuration if you encounter difficulties with replication.

The following diagram shows a basic master-replica topology:

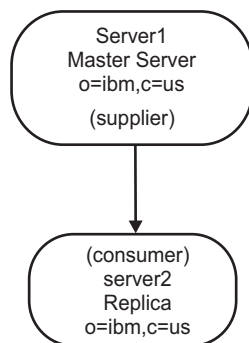


Figure 9. Basic master-replica topology

To define a basic master-replica topology, you must:

1. Create a master server and define what it contains. Select the subtree that you want to be replicated and specify the server as the master.
2. Create credentials to be used by the supplier.
3. Create a replica server.
4. Export data to the replica.

The following sections explain how to accomplish these tasks.

Note: If the entry that you are trying to make the root of a new replication context is not a suffix in the server, before you can use the **Add subtree** function to add the replication configuration information, you must ensure that its ACLs are defined as follows:

For Non-filtered ACLs :

```
ownersource : <the entry DN>
ownerpropagate : TRUE
aclsource : <the entry DN>
aclpropagate: TRUE
```

Filtered ACLs :

```
ownersource : <the entry DN>
ownerpropagate : TRUE
ibm-filteraclinherit : FALSE
ibm-filteraclentry : <any value>
```

To satisfy the ACL requirements, if the entry is not a suffix in the server, edit the ACL for that entry in the **Manage entries** panel:

1. Click **Directory management**→**Manage entries** in the left nav panel.
2. Select an entry and open the **Select Action** menu.
3. Select **Edit ACL** and click **Go**. If you want to add Non-filtered ACLs, select that tab and add an entry **cn=this** with the role **access-id** for both ACLs and owners.

4. Ensure that **Propagate ACLs** and **Propagate owner** are checked. If you want to add Filtered ACLs select that tab and add an entry **cn=this** with the role **access-id** for both ACLs and owners.
5. Ensure that **Accumulate filtered ACLs** is unchecked and that **Propagate owner** is checked. See “Working with ACLs” on page 318 for more detailed information.

Using Web Administration:

Note: These procedures assume that all servers involved are IBM Tivoli Directory Server version 5.x and 6.0 servers. They also assume that you have installed and can use the Web Administration Tool with administrative rights. See the *IBM Tivoli Directory Server Version 6.0 Installation and Configuration Guide* for information about installing the Web Administration Tool.

Creating a master server (replicated subtree)

Note: The server must be running to perform this task.

This task designates an entry as the root of an independently replicated subtree and creates an **ibm-replicasubentry** entry under it representing this server as the single master for the subtree. To create a replicated subtree, you must designate the subtree that you want the server to replicate.

Note: On the Linux, Solaris, and HP-UX platforms, if a referral fails because the server being referred to is not running, ensure that the environment variable `LDAP_LOCK_REC` has been set in your system environment. No specific value is required.

```
set LDAP_LOCK_REC=anyvalue
```

1. Use the Web Administration Tool to log on to the master server.
2. Expand the **Replication management** category in the navigation area of the Web Administration Tool and click **Manage topology**.
3. Click **Add subtree**.
4. Enter the DN of the subtree that you want to replicate or click **Browse** to expand the entries to select the entry that is to be the root of the subtree. For this example, use `c=ibm,o=us`.

Notes:

- a. If you are replicating a subtree, and the subtree is not a suffix, you must replicate the parent of the subtree on the replica first.
5. The master server referral URL is displayed in the form of an LDAP URL, for example:

For non-SSL:

```
ldap://<myservername>.<mylocation>.<mycompany>.com:<port>
```

For SSL:

```
ldaps://<myservername>.<mylocation>.<mycompany>.com:<port>
```

The default URL is `ldap://localhost:389`

Note: The master server referral URL is optional. It is used only:

- If the server contains (or will contain) any read-only subtrees.
- To define a referral URL that is returned for updates to any read-only subtree on the server.

6. Click **OK**.
7. The new subtree is displayed on the Manage topology panel under the heading **Replicated subtrees**.

Creating the credentials

Credentials identify the method and required information, such as a DN and password, that the supplier uses in binding to the consumer.

1. If you have not already done so, use the Web Administration Tool to log on to the master server.
2. Expand the **Replication management** category in the navigation area of the Web Administration Tool and click **Manage credentials**.
3. Select **cn=replication,cn=IBMpolicies** from the list of locations to store the credentials.

Note: The Web Administration Tool allows you to define credentials in three locations. See “Adding credentials” on page 224 for additional information about the different types of credentials that you can create.

4. Click **Add**.
5. Enter the name for the credentials you are creating; for example, **mycreds**, **cn=** is prefilled in the field for you.
6. Select **Simple bind** as the type of authentication and click **Next**.

Note: You can also select **Kerberos** or **SSL with certificates**.

- Enter the DN that the server uses to bind to the replica; for example, **cn=any**

Note: This DN cannot be the same as your server administration DN.

- Enter the password the server uses when it binds to the replica; for example, **secret**.
- Enter the password again to confirm that there are no typographical errors.
- If you want, enter a brief description of the credentials.
- Click **Finish**.

Note: You might want to record the credential’s bind DN and password for future reference.

Creating a replica server

Note: The servers must be running to perform this task.

On the master server:

1. If you have not already done so, use the Web Administration Tool to log on to the master server.
2. Expand the **Replication management** category in the navigation area of the Web Administration Tool and click **Manage topology**.
3. Select the subtree that you want to replicate and click **Show topology**.
4. Select the supplier server and click **Add replica**.
5. On the **Server** tab of the **Add replica** window:
 - Enter the host name and port number of the replica server.
 - Leave the **Enable SSL** check box unchecked.
 - Enter the replica name or leave this field blank to use the host name.

- Enter the replica ID. If the server on which you are creating the replica is running, click **Get replica ID** to automatically prefill this field. This is a required field. If you don't know the replica ID, enter **unknown**.
- Enter a description of the replica server.
- Specify the credentials that the replica uses to communicate with the master.
 - a. Click **Select**.
 - b. Click the radio button next to **cn=replication,cn=IBMpolicies**.
 - c. Click **Show credentials**.
 - d. Select **cn=replication,cn=ibmpolicies**.
 - e. Click **Show credentials**.
 - f.
 - g. Click **OK**.

See "Adding credentials" on page 224 for additional information on agreement credentials.

6. Click the **Additional** tab.
 - a. Keep the **Select a replication schedule or enter DN (optional)** set to **None**. This sets the default as immediate replication.
 - b. Do not deselect any capabilities. See "Adding a replica server" on page 231.
 - c. Keep the **Replication method** set to **Single threaded**.

Note: Only IBM Tivoli Directory Server 6.0 can have the replication method set to either Single threaded or Multi-threaded. IBM Tivoli Directory Server 5.x is always single threaded.

- d. Click to select the **Add credential information on consumer** check box.

Note: If the credential is external, you need to set up the IBM WebSphere Application Server environment variable. See 230.

- e. Enter the administrator DN for the consumer (replica) server. For example `cn=root`.

Note: If the administrator DN which was created during the server configuration process was `cn=root`, then enter the full administrator DN. Do not just use `root`.

- f. Enter the administrator password for the consumer (replica) server. For example `secret`.

Note: The consumer server should be running.

- g. Click **OK** to create the replica.

Note: If the credentials exist, a message is displayed saying the credentials exist. If the credentials don't exist, they are added, and a message prompt is displayed. You are also prompted to restart the server. The panel also shows two port numbers: server port number (this port number cannot be edited) and the admin daemon port number. Make sure you have the correct admin daemon port number for the specific instance used. If the wrong admin daemon port number is specified, the admin daemon fails to restart the server.

- h. Click **OK**.

Note: A message is displayed, indicating that the server attempted to add the topology to the consumer. The message indicates whether this attempt is successful.

- i. Click **OK**.

See “Adding a replica server” on page 231 for more detailed information.

Copying data to the replica: To ensure that the servers are synchronized, you must first quiesce the master. This means that the master does not accept any updates from its clients.

1. If you have not already done so, use the Web Administration Tool to log on to the master server.
2. Expand the **Replication management** category in the navigation area of the Web Administration Tool and click **Manage topology**.
3. Select the subtree you have replicated.
4. Click **Quiesce/Unquiesce** to quiesce the subtree.
5. Click **OK**.

You must now export the data from the master to the replica. This is a manual procedure.

On the master server create an LDIF file for the data. To copy all the data contained on the master server, issue the command:

```
idsdb2ldif -o <masterfile.ldif> -I <instance_name> -k <key seed> -t <key salt>
```

Note: You must use the `-I` option if there is more than one instance. You must use the `-k` and `-t` options if keys on the server are not in sync.

If you want to copy just the data from a single subtree the command is:

```
idsdb2ldif -o <masterfile.ldif> -s <subtreeDN> -I <instance_name>  
-k <key seed> -t <key salt>
```

Note: You must use the `-I` option if there is more than one instance. You must use the `-k` and `-t` options if keys on the server are not in sync.

Note: The four operational attributes, `createTimestamp`, `creatorsName`, `modifiersName`, and `modifyTimestamp` are exported to the LDIF file unless the `-j` option is specified.

On the computer where you are creating the replica:

1. Ensure that the suffixes used by the master are defined in the **ibmslapd.conf** file.
2. Stop the replica server.
3. Copy the `<masterfile.ldif>` file to the replica and issue the command:

```
idsldif2db -r no -i <masterfile.ldif> -I <instance name>
```

The replication agreements, schedules, credentials (if stored in the replicated subtree) and entry data are loaded on the replica.

4. Start the server.

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 I, “Synchronizing two-way cryptography between server instances,” on page 537 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 will be 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 will be 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.

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 D, “ASCII characters from 33 to 126,” on page 487 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`.

Adding the supplier information to the replica: If you did not select to add the credential information to the consumer or if a problem occurred in adding the credential information to the replica, you need to change the replica’s configuration to identify who is authorized to replicate changes to it, and add a referral to a master.

1. Use the Web Administration Tool to log on as the directory administrator to the computer where you are creating the replica.
2. Expand **Replication management** in the navigation area of the Web Administration Tool and click **Manage replication properties**.
3. Under Supplier information, click **Add**.
4. Select a supplier from the **Replicated subtree** drop-down menu, select **Use entry from below**, and enter the name of the replicated subtree for which you want to configure supplier credentials.
5. Enter the replication bindDN. In this example, `cn=any` is used.
6. Enter and confirm the credential password. In this example, `secret` is used. See “Adding credentials” on page 224.
7. Click **OK**.
8. You must restart the replica for the changes to take effect.

See “Adding the supplier information to a replica” on page 236 for more detailed information about supplier information.

Starting replication

The replica is in a suspended state and no replication is occurring. After you have finished setting up your replication topology, on the master you must:

1. If you have not already done so, use the Web Administration Tool to log on to the master server.
2. Expand the **Replication management** category in the navigation area of the Web Administration Tool and click **Manage queues**.
3. Select the new replica.
4. Click **Queue details**.
5. Click **Pending changes**.
6. If there are any pending changes, click **Skip all**. If there are no changes pending click **Cancel**. This prevents duplication of the topology information that was loaded with the `<masterfile.ldif>` file. If you have created multiple new replicas, repeat steps 1 through 6 for each of the new servers.
7. Click **Manage topology** under the **Replication management** category in the navigation area.
8. Select the subtree you have replicated. The status should be **Quiesced**.
9. Click **Quiesce/Unquiesce** to unquiesce the subtree.
10. Click **OK**. The master now receives updates from its clients and places them in the replication queues.
11. Click **Manage queues** under the **Replication management** category in the navigation area.
12. Select the replica.
13. Click **Suspend/resume** to start receiving replication updates for that server. Repeat steps 10 through 13 for each server that was suspended.

Note: If you promote to a master, you need to resume the queues on the master.

See “Managing queues” on page 240 for more detailed information about managing queues.

Using the command line:

This scenario assumes that you are creating a new replicated subtree and that only server1 contains any entry data. All other servers are newly installed and have a configured database.

Note:

```
dn: o=ibm,c=us
objectclass: organization
objectclass: ibm-replicationContext
o: ibm
```

is the subtree you want to create. If this entry already exists, then modify it to add **objectclass=ibm-replicationContext** instead of adding the entire entry.

To create a replica for a subtree, you need to create a replication agreement between the master and the replica, see “Replication agreements” on page 169. This agreement needs to be loaded on both the master and the replica.

The relationship between the two servers is that the master is a supplier to the replica and the replica is a consumer of the master.

To create the master (server1) and replica (server2) for the subtree **o=ibm,c=us**:

1. At the machine where the master is located, create a file to contain the agreement information, for example, *myreplicainfofile*, where *myreplicainfofile* contains:

Note: Replace all occurrences of *<server1-uuid>* in the following files with the value of the **ibm-slappedServerId** attribute from the master server's **cn=Configuration** entry. This value is generated by the server the first time it is started. You can find it either by performing an **idsldapsearch** of the **cn=Configuration** entry or using the **grep** command on the **ibmslapd.conf** file, if you have a UNIX-based system. Similarly, all occurrences of the *<server2-uuid>* must be replaced with the value of the **ibm-slappedServerId** attribute from the replica server's **cn=Configuration** entry.

```
###Replication Context - needs to be on all suppliers and consumers
dn: cn=replication,cn=IBMpolicies
objectclass: container
```

```
dn: o=ibm,c=us
objectclass: organization
objectclass: ibm-replicationContext
```

###Copy the following to servers at v5.x and above.

```
###Replica Group
dn: ibm-replicaGroup=default, o=IBM, c=US
objectclass: top
objectclass: ibm-replicaGroup
ibm-replicaGroup: default
```

```
###Bind Credentials/method to replica server - replication agreement
###points to this.
dn: cn=server2 BindCredentials,cn=replication,cn=IBMpolicies
objectclass: ibm-replicationCredentialsSimple
cn: server2 BindCredentials
replicaBindDN: cn=any
replicaCredentials: secret
description: Bind method of the master (server1) to the replica (server2)
```

```
###Replica SubEntry
dn: ibm-replicaServerId=<server1-uuid>,ibm-replicaGroup=default,o=IBM,c=US
objectclass: top
objectclass: ibm-replicaSubentry
ibm-replicaServerId: <server1-uuid>
ibm-replicationServerIsMaster: true
cn: server1
description: master server
```

```
###Replication Agreement to the replica server
dn: cn=server2,ibm-replicaServerId=<server1-uuid>,
   ibm-replicaGroup=default,o=IBM,c=US
objectclass: top
objectclass: ibm-replicationAgreement
cn: server2
ibm-replicaConsumerId: <server2-uuid>
ibm-replicaUrl: ldap://server2:389
ibm-replicaCredentialsDN: cn=server2 BindCredentials,cn=replication,
   cn=IBMpolicies
description: replica server (server2)
```

2. Stop the master, if it is not already stopped.


```
ibmdirctl -h server1 -D <adminDN> -w <adminPW> -p 389 stop
```
3. To load the new replication topology to the master, issue the command:


```
idsldif2db -r no -i <myreplicainfofile> -I <instance name>
```

4. To generate a file with all of the data necessary to synchronize the new replica, issue the command:

```
idsdb2ldif -o <masterfile.ldif> -I <instance_name> -s o=ibm,c=us  
-k <key seed> -t <key salt>
```

Note: You must use the `-I` option if there is more than one instance. You must use the `-k` and `-t` options if keys on the server are not in sync.

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*, see 177.

See “idsdb2ldif, db2ldif” on page 436 for more information.

Note: Perform steps 5 through 9 on the machine where server2 is located.

5. Copy `<masterfile.ldif>` to the replica.
6. Start the replica, server2, in configuration only mode.

```
idsslapd -I <instance name> -a
```
7. Make sure you have a backup of the original `ibmslapd.conf`, `ibmslapdcfg.ksf`, and `ibmslapddir.ksf` files.
8. You must configure server2 to be a replica server. Use the **idsldapadd** command to add the following entry to the **ibmslapd.conf** file on server2. On server2 issue the following command:

```
idsldapadd -D <adminDN> -w<adminPW> -i<filename>
```

where `<filename>` contains:

```
dn: cn=Master Server, cn=configuration  
objectclass: ibm-slapdReplication  
cn: Master Server  
ibm-slapdMasterDN: cn=any  
ibm-slapdMasterPW: secret  
ibm-slapdMasterReferral: ldap://server1:389/
```

Note: The `ibm-slapdMasterDN` and `ibm-slapdMasterPW` values must match the values stored on the master server, server1, in the entry “cn=server2 BindCredentials” in step 1.

9. Stop the replica, server2. To stop the server issue the command:

```
ibmdirctl -h server2 -D <AdminDN> -w <Adminpwd> -p 389 stop
```

10. Save the **ibmslapd.conf** file as a new backup.
11. Issue the following command:

```
idsldif2db -r no -i <masterfile.ldif> -I <instance name>
```
12. Start the master (server1) and the replica (server2). On each of the servers issue the command:

```
idsslapd -I <LDAPinstance>
```

Note: If you are copying a subtree to a v4.1 or earlier server, you must not copy the **ibm-replicagroup=default** subtree and you must remove the **ibm-replicationcontext** auxiliary class, because neither of these are supported by the 4.1 schema.

Setting up a simple topology with peer replication

Peer replication is a replication topology in which multiple servers are masters. Use peer replication only in environments where the update vectors are well known. Updates to particular objects within the directory must be done only by one peer server. This is intended to prevent a scenario in which one server deletes an object, followed by another server modifying the object. This scenario creates the possibility of a peer server receiving a delete command followed by a modify command for the same object, which creates a conflict. Replicated delete and rename request are accepted in the order received without conflict resolution. See “Replication conflict resolution” on page 167 for more information about conflict resolution.

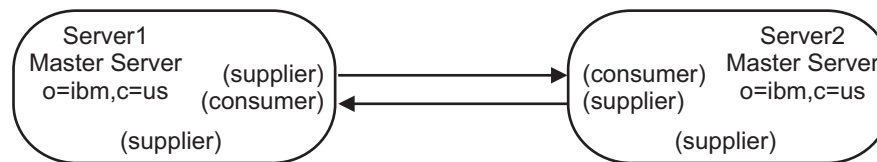


Figure 10. Basic peer-to-peer topology

This section shows how to set up a replication topology between two servers only.

Using Web Administration:

Note: The server must be running to perform this task.

Expand the **Replication management** category in the navigation area and click **Manage topology**.

1. Select the subtree that you want to replicate and click **Show topology**.
2. Click the box next to the existing servers to expand the list of supplier servers, if you want to view the existing topology.
3. Click **Add master**.

On the **Server** tab of the **Add master** window:

- Enter the host name and port number for the server you are creating. The default port is 389 for non-SSL and 636 for SSL. These are required fields.
- Select whether to enable SSL communications.
- Select whether you want to create the server as a gateway server.
- Enter the server name or leave this field blank to use the host name.
- Enter the server ID. If the server on which you are creating the peer-master is running, click **Get server ID** to automatically prefill this field. If you do not know the server ID, enter **unknown**.
- Enter a description of the server.
- You must specify the credentials that the server uses to communicate with the master server. Click **Select**.

Note: The Web Administration Tool allows you to define credentials in the following places:

- **cn=replication,cn=localhost**, which keeps the credentials only on the server that uses them. Placing credentials in **cn=replication,cn=localhost** is considered more secure.

- **cn=replication,cn=IBMpolicies**, which is available even when the server under which you are trying to add a replica is not the same server that you are connected to with the Web Administration Tool. Credentials placed under this location are replicate to the servers.

Note: The location **cn=replication,cn=IBMpolicies** is only available, if the **IBMpolicies** support **OID, 1.3.18.0.2.32.18**, is present under the **ibm-supportedcapabilities** of the root DSE.

- Within the replicated subtree, in which case the credentials are replicated with the rest of the subtree. Credentials placed in the replicated subtree are created beneath the **ibm-replicagroup=default** entry for that subtree.
1. Select the location for the credentials you want to use. Preferably this is **cn=replication,cn=localhost**.
 2. If you have already created a set of credentials, click **Show credentials**.
 3. Expand the list of credentials and select the one you want to use.
 4. Click **OK**.
 5. If you do not have preexisting credentials, click **Add** to create the credentials. See “Adding credentials” on page 224 for additional information on agreement credentials.

On the **Additional** tab:

1. Specify a replication schedule from the drop-down list or click **Add** to create one. See “Creating replication schedules” on page 238
2. From the list of supplier capabilities, you can deselect any capabilities that you do not want replicated to the consumer.
If your network has a mix of servers at different releases, capabilities are available on later releases that are not available on earlier releases. Some capabilities, like filter ACLs and password policy, make use of operational attributes that are replicated with other changes. In most cases, if these features are used, you want all servers to support them. If all of the servers do not support the capability, you do not want to use it. For example, you would not want different ACLs in effect on each server. However, there might be cases where you might want to use a capability on the servers that support it, and not have changes related to the capability replicated to servers that do not support the capability. In such cases, you can use the capabilities list to mark certain capabilities to not be replicated.
3. Check the **Add credential information on consumer** check box, if you want to enable dynamic updates of the supplier credentials. This selection automatically updates the supplier credentials in the configuration file of the consumer server. This enables the topology information to be replicated to the server.
 - Type the Administration DN for the consumer server. For example **cn=root**.

Note: If the administrator DN which was created during the server configuration process was **cn=root**, then enter the full administrator DN. Do not just use **root**.

 - Type the Administration password for the consumer server. For example **secret**.
4. Click **OK**.

5. Supplier and consumer agreements are listed between new master server and any existing servers. Uncheck any agreements that you do not want to be created. This is especially important if you are creating a gateway server.
6. Click **Continue**.
7. Messages might be displayed noting that additional actions must be taken. Perform or take note of the appropriate actions. When you are finished, click **OK**.
8. Add the appropriate credentials.

Note: In some cases the Select credentials panel will pop up asking for a credential that is located in a place other than `cn=replication,cn=localhost`. In such situations you must provide a credential object that is located in a place other than `cn=replication,cn=localhost`. Select the credentials the subtree is going to use from the existing sets of credentials or create new credentials. See “Adding credentials” on page 224.

9. Click **OK** to create the peer-master.
10. Messages might be displayed noting that additional actions must be taken. Perform or take note of the appropriate actions. When you are finished, click **OK**.

Using the command line:

This scenario assumes that you are creating a new replicated subtree and that only `server1` contains any entry data. All other servers are newly installed and have a configured database.

Note:

```
dn: o=ibm,c=us
objectclass: organization
objectclass: ibm-replicationContext
o: ibm
```

is the subtree you want to create. If this entry already exists, then modify it to add **objectclass=ibm-replicationContext** instead of adding the entire entry.

`server1` and `server2` are peer-master servers. That means that while they receive updates from each other, they only replicate entries received from clients. While both masters have the same entry content, only the server that has received the client request replicates the entry. Both masters are suppliers and consumers to each other and suppliers to the other servers.

To create the peer-masters (`server1` and `server2`) for the subtree **o=ibm,c=us**:

1. Start servers `server1` and `server2` in configuration mode. On each of the servers issue the command:


```
idsldapd -I <LDAPinstance> -a
```
2. If `idsdiradm` is not running for each instance, start `idsdiradm`:


```
idsdiradm -I <LDAP_instance>
```
3. You must configure `server1` and `server2` to be peer servers. Use the **idsldapadd** command to add the following entry to the **ibmslapd.conf** file on `server1` and `server2`. On `server1` and `server2` issue the following command:


```
idsldapadd -D <adminDN> -w<adminPW> -i<filename>
```

where *<filename>* contains:

```
dn: cn=Master Server, cn=configuration
objectclass: ibm-slpadReplication
cn: Master Server
ibm-slpadMasterDN: cn=master
ibm-slpadMasterPW: master
```

Note: It is critical that these entries be exactly the same on both servers because this example uses a credentials object that is shared on all the servers. The password is entered in cleartext, but is encrypted in the file.

4. Stop server1 and server2. To stop the servers issue the following command on each of the servers:

```
ibmdirctl -h <serverx> -D <adminDN>-w <adminPW>-p 389 stop
```

where *<serverx>* is the name of the server.

5. Save the **ibmslapd.conf** files.

6. At the machine where the master server, server1, is located, create a file to use for updates to the agreement information, for example *mycredentialsfile* where *mycredentialsfile* contains:

```
dn: cn=replication,cn=IBMpolicies
o: ibm
objectclass: top
objectclass: organization
objectclass: ibm-replicationContext
```

```
dn: ibm-replicaGroup=default, cn=replication,cn=IBMpolicies
objectclass: top
objectclass: ibm-replicaGroup
ibm-replicaGroup: default
```

```
dn: ibm-replicaServerId=<server1-uuid>,ibm-replicaGroup=default,cn=replication,
  cn=IBMpolicies
objectclass: top
objectclass: ibm-replicaSubentry
ibm-replicaServerId: <server1-uuid>
ibm-replicationServerIsMaster: true
cn: server1
description: server 1 (peer master) ibm-replicaSubentry
```

```
dn: ibm-replicaServerId=<server2-uuid>,ibm-replicaGroup=default,cn=replication,
  cn=IBMpolicies
objectclass: top
objectclass: ibm-replicaSubentry
ibm-replicaServerId: <server2-uuid>
ibm-replicationServerIsMaster: true
cn: server2
description: server2 (peer master) ibm-replicaSubentry
```

#server1 to server2 agreement

```
dn: cn=server2,ibm-replicaServerId=<server1-uuid>,
  ibm-replicaGroup=default,cn=replication,cn=IBMpolicies
objectclass: top
objectclass: ibm-replicationAgreement
cn: server2
ibm-replicaConsumerId: <server2-uuid>
ibm-replicaUrl: ldap://server2:389
ibm-replicaCredentialsDN: cn=simple,cn=replication,cn=localhost
description: server1(master) to server2(master) agreement
```

#server2 to server1 agreement

```
dn: cn=server1,ibm-replicaServerId=<server2-uuid>,
```

```

    ibm-replicaGroup=default,cn=replication,cn=IBMpolicies
objectclass: top
objectclass: ibm-replicationAgreement
cn: server1
ibm-replicaConsumerId: <server1-uuid>
ibm-replicaUrl: ldap://server1:389
ibm-replicaCredentialsDN: cn=simple,cn=replication,cn=localhost
description: server2(master) to server1(master) agreement

```

7. Issue the command:

```
idsldif2db -r no -i <mycredentialsfile> -I <instance_name>
```

8. Copy <mycredentialsfile> to the machines where server2 is located and issue the command:

```
idsldif2db -r no -i <mycredentialsfile> -I <instance_name>
```

9. At the machine where server1 is located create a file <mytopologyfile> where <mytopologyfile> includes:

Note: Replace all occurrences of <server1-uuid> in the following files with the value of the **ibm-slappedServerId** attribute from the master server's **cn=Configuration** entry. This value is generated by the server the first time it is started. You can find it either by performing an **idsldapsearch** of the **cn=Configuration** entry or using the **grep** command on the **ibmslapd.conf** file, if you have a UNIX-based system. Similarly, all occurrences of the <serverx-uuid> (where x represents 1 or 2) must be replaced with the value of the **ibm-slappedServerId** attribute from the respective server's **cn=Configuration** entry.

```
dn: o=ibm,c=us
o: ibm
```

```
objectclass: top
objectclass: organization
objectclass: ibm-replicationContext
```

```
dn: ibm-replicaGroup=default, o=ibm,c=us
objectclass: top
objectclass: ibm-replicaGroup
ibm-replicaGroup: default
```

```
dn: ibm-replicaServerId=<server1-uuid>,ibm-replicaGroup=default,o=ibm,c=us
objectclass: top
objectclass: ibm-replicaSubentry
ibm-replicaServerId: <server1-uuid>
ibm-replicationServerIsMaster: true
cn: server1
description: server 1 (peer master) ibm-replicaSubentry
```

```
dn: ibm-replicaServerId=<server2-uuid>,ibm-replicaGroup=default,o=ibm,c=us
objectclass: top
objectclass: ibm-replicaSubentry
ibm-replicaServerId: <server2-uuid>
ibm-replicationServerIsMaster: true
cn: server2
description: server2 (peer master) ibm-replicaSubentry
```

#server1 to server2 agreement

```
dn: cn=server2,ibm-replicaServerId=<server1-uuid>,
    ibm-replicaGroup=default,o=ibm,c=us
objectclass: top
objectclass: ibm-replicationAgreement
cn: server2
ibm-replicaConsumerId: <server2-uuid>
ibm-replicaUrl: ldap://server2:389
ibm-replicaCredentialsDN: cn=simple,cn=replication,cn=localhost
description: server1(master) to server2(master) agreement
```

```
#server2 to server1 agreement
dn: cn=server1,ibm-replicaServerId=<server2-uuid>,
    ibm-replicaGroup=default,o=ibm,c=us
objectclass: top
objectclass: ibm-replicationAgreement
cn: server1
ibm-replicaConsumerId: <server1-uuid>
ibm-replicaUrl: ldap://server1:389
ibm-replicaCredentialsDN: cn=simple,cn=replication,cn=localhost
description: server2(master) to server1(master) agreement
```

10. To load this topology, issue the command:

```
idsldif2db -r no -i <mytopologyfile> -I <instance_name>
```

where `-r no` prevents replication of the set of entries.

11. At this point you might want to load additional data for your subtree.

Note: Use the `-r no` flag to prevent replication of the set of entries.

12. When you have finished loading the data, to be able to export the topology and any additional data for the replication context to populate the other servers, issue the command:

```
idsdb2ldif -s"o=ibm,c=us" -o <mymasterfile.ldif> -I <instance_name>
-k <key seed> -t <key salt>
```

Note: You must use the `-I` option if there is more than one instance. You must use the `-k` and `-t` options if keys on the server are not in sync.

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*, see 177.

See “idsdb2ldif, db2ldif” on page 436 for more information.

13. Restart server1.
14. Copy `<mymasterfile.ldif>` to the machine where server2 is located.
15. Issue the following command:

```
idsldif2db -r no -i <masterfile.ldif> -I <instance_name>
```

16. Start server2:

```
idsslapd -I <instance_name>
```

Creating a master-forwarder-replica topology

The following diagram shows a master-forwarder-replica topology:

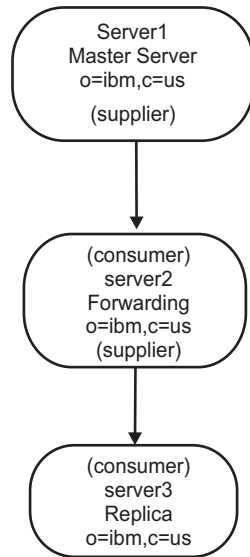


Figure 11. Master-forwarding server-replica topology

To define a master-forwarder-replica topology, you must:

1. Create a master server and a replica server. You have already done this; see “Creating a master-replica topology” on page 172.
2. Create a new replica server for the original replica. See “Adding a replica server” on page 231.
3. Copy data to the replicas. See “Copying data to the replica” on page 176.

Changing the replica to a forwarding server

Note: Before starting to set up your replication topology, make a backup copy of your original `ibmslapd.conf` file for each server. You can use this backup copy to restore your original configuration if you encounter difficulties with replication.

If you have set up a replication topology (see “Adding a subtree” on page 222) with a master (server1) and a replica (server2), you can change the role of server2 to that of a forwarding server. To do this you need to create a new replica (server3) under server2.

Using Web Administration:

1. Start all the servers.
2. If you have not already done so, use the Web Administration Tool to log on to the master server (server1).
3. Expand the Replication management category in the navigation area and click **Manage topology**.
4. Select the subtree that you want to replicate and click **Show topology**.
5. Click the box next to the **server1** selection to expand the list of servers.
6. Select server2 and click **Add replica**.
7. On the **Server** tab of the **Add replica** window:
 - Enter the host name and port number of the replica server.
 - Leave the **Enable SSL** check box unchecked.
 - Enter the replica name or leave this field blank to use the host name.

- Enter the replica ID. If the server on which you are creating the replica is running, click **Get replica ID** to automatically prefill this field. This is a required field.
- Enter a description of the replica server.
- Specify the credentials that the replica uses to communicate with the master.
 - a. Click **Select**.
 - b. Click the radio button next to **cn=replication,cn=IBMpolicies**.

Note: The **mycreds** credentials need to be created under `cn=replication,cn=ibmpolicies` on the forwarder, unless `cn=ibmpolicies` is replicated.

- c. Click **Show credentials**.
- d. Expand the list of credentials and select **mycreds**.
- e. Click **OK**.

See “Adding credentials” on page 224 for additional information on agreement credentials.

8. Click the **Additional** tab.
 - a. Keep the **Specify a replication schedule or enter DN (optional)** set to **None**. This sets the default as immediate replication.
 - b. Do not deselect any capabilities.
 - c. Keep the **Replication method** set to **Single threaded**.
 - d. Click to select the **Add credential information on consumer** check box.
 - e. Enter the administrator’s DN for the consumer (replica) server. For example `cn=root`.

Note: If the administrator DN which was created during the server configuration process was `cn=root`, then enter the full administrator DN. Do not just use `root`.

- f. Enter the administrator’s password for the consumer (replica) server. For example `secret`.
- g. Click **OK** to create the replica. A message is displayed noting that additional actions must be taken, including restarting the replica server. Take the appropriate actions.
- h. Click **OK**.

9. Copy data from server1 to the new replica server3. See “Copying data to the replica” on page 176 for information about how to do that.

Note: The topology changes are replicated to server2 by the master server1.

10. Add the supplier agreement to server3 that makes server2 a supplier to server3 and server3 a consumer to server2. See “Adding the supplier information to a replica” on page 236 for information about how to do this.

Note: This step needs to be performed only if you did not select the **Add credential information on consumer** check box, or supplier information failed to be added to the consumer configuration file.

The server roles are represented by icons in the Web Administration Tool. Your topology is now:

- server1 (master)
 - server2 (forwarder)
 - server3 (replica)

Using the command line:

This scenario assumes that you are creating a new replicated subtree and that only server1 contains any entry data. All other servers are newly installed and have a configured database.

Note:

```
dn: o=ibm,c=us
objectclass: organization
objectclass: ibm-replicationContext
o: ibm
```

is the subtree you want to create. If this entry already exists, then modify it to add **objectclass=ibm-replicationContext** instead of adding the entire entry.

This procedure is similar to the one for a single master and replica, except that the entire topology must be added to each of the servers and the content of the agreement information file is more complex. The file now includes information for the forwarding server and supplier-consumer information.

The supplier-consumer relationship for this scenario is:

- The master is a supplier to the forwarder.
- The forwarder has two roles:
 1. A consumer of the master
 2. A supplier to the replica
- The replica is a consumer of the forwarder.

To create the master (server1), a forwarder (server2), and replica (server3) for the subtree **o=ibm,c=us**:

1. At the machine where the master server is located, create a file to contain the agreement information, for example *myreplicainfofile* where *myreplicainfofile* contains:

Note: Replace all occurrences of *<server1-uuid>* in the following files with the value of the **ibm-slappedServerId** attribute from the master server's **cn=Configuration** entry. This value is generated by the server the first time it is started. You can find it either by performing an **idsldapsearch** of the **cn=Configuration** entry or using the **grep** command on the **ibmslapd.conf** file, if you have a UNIX-based system. Similarly, all occurrences of the *<server2-uuid>* and the *<server3-uuid>* must be replaced with the value of the **ibm-slappedServerId** attribute from the respective server's **cn=Configuration** entry.

```
dn: cn=replication,cn=IBMpolicies
objectclass: container
```

```
dn: o=ibm,c=us
objectclass: organization
objectclass: ibm-replicationContext
```

```
dn: ibm-replicaGroup=default, o=ibm,c=us
objectclass: top
objectclass: ibm-replicaGroup
ibm-replicaGroup: default
```

```
dn: cn=server2 BindCredentials,cn=replication,cn=IBMpolicies
objectclass: ibm-replicationCredentialsSimple
#or ibm-replicationCredentialsExternal or
```

```

#ibm-replicationCredentialsKerberos
cn: server2 BindCredentials
replicaBindDN: cn=any
replicaCredentials: secret
description: Bindmethod of server 1 (the master) to server2

dn: cn=server3 BindCredentials,cn=replication,cn=IBMpolicies
objectclass: ibm-replicationCredentialsSimple
cn: server3 BindCredentials
replicaBindDN: cn=any
replicaCredentials: secret
description: Bindmethod of server2 (the forwarder) to server3 (the replica)

dn: ibm-replicaServerId=<server1-uuid>,ibm-replicaGroup=default,o=ibm,c=us
objectclass: top
objectclass: ibm-replicaSubentry
ibm-replicaServerId: <server1-uuid> #whatever the ID is in the config
ibm-replicationServerIsMaster: true #true if master, false if forwarder
cn: server1
description: master ibm-replicaSubentry

dn: ibm-replicaServerId=<server2-uuid>,ibm-replicaGroup=default,o=ibm,c=us
objectclass: top
objectclass: ibm-replicaSubentry
ibm-replicaServerId: <server2-uuid>
ibm-replicationServerIsMaster: false
cn: server2
description: forwarder ibm-replicaSubentry

dn: cn=forwarder1,ibm-replicaServerId=<server1-uuid>,
    ibm-replicaGroup=default,o=ibm,c=us
objectclass: top
objectclass: ibm-replicationAgreement
cn: server2
ibm-replicaConsumerId: <server2-uuid>
ibm-replicaUrl: ldap://server2:389
ibm-replicaCredentialsDN: cn=server2 BindCredentials,cn=replication,
    cn=IBMpolicies
description: server1 (the master) to server2 (the forwarder) agreement

dn: cn=server3,ibm-replicaServerId=<server2-uuid>,
    ibm-replicaGroup=default,o=ibm,c=us
objectclass: top
objectclass: ibm-replicationAgreement
cn: server3
ibm-replicaConsumerId: <server3-uuid>-uuid
ibm-replicaUrl: ldap://server3:389
ibm-replicaCredentialsDN: cn=server3 BindCredentials,cn=replication,
    cn=IBMpolicies
description: server2 (the forwarder) to server3 (the replica) agreement

```

2. Stop the master, if it is not already stopped.


```
ibmdirctl -h server1 -D <adminDN> -w <adminPW> -p 389 stop
```
3. To load the new replication topology to the master, issue the command:


```
idsldif2db -r no -i <myreplicainfofile> -I <instance_name>
```
4. To generate a file with all of the data to synchronize the new replica, issue the command:


```
idsdb2ldif -o <masterfile.ldif> -I <instance_name> -s o=ibm,c=us
    -k <key seed> -t <key salt>
```

Note: You must use the `-I` option if there is more than one instance. You must use the `-k` and `-t` options if keys on the server are not in sync.

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*, see 177.

See “idsdb2ldif, db2ldif” on page 436 for more information.

5. Copy `<masterfile.ldif>` to the machine where server2 is located.
6. Start the forwarder, server2, in configuration only mode.
7. You must configure server2 to be a replica server. Use the `idsldapadd` command to add the following entry to the `ibmslapd.conf` file on server2. On server2 issue the following command:

```
idsslapd -I <LDAPinstance> -a
```

```
idsldapadd -D <adminDN> -w<adminPW> -i<filename>
```

where `<filename>` contains:

```
dn: cn=Master Server, cn=configuration
objectclass: ibm-slapdReplication
cn: Master Server
ibm-slapdMasterDN: cn=any
ibm-slapdMasterPW: secret
ibm-slapdMasterReferral: ldap://server1:389/
#referral to master when trying to add to consumer.
#Referral can also be added to replicaContext, which would be
#checked first for a valid server.
```

Note: The `ibm-slapdMasterDN` and `ibm-slapdMasterPW` values must match the values stored on the master server, server1, in the entry “cn=server2 BindCredentials”.

8. Stop server2.

```
ibmdirctl -h server2 -D <adminDN> -w <adminPW> -p 389 stop
```

9. Save the `ibmslapd.conf` file.
10. Copy `<masterfile.ldif>` to the machine where server3 is located.
11. Start the replica, server3, in configuration only mode.

```
idsslapd -I <LDAPinstance> -a
```

12. You must configure server3 to be a replica server. Use the `idsldapadd` command to add the following entry to the `ibmslapd.conf` file on server3. On server3 issue the following command:

```
idsldapadd -D <adminDN> -w<adminPW> -i<filename>
```

where `<filename>` contains:

```
dn: cn=Master Server, cn=configuration
objectclass: ibm-slapdReplication
cn: Master Server
ibm-slapdMasterDN: cn=any
ibm-slapdMasterPW: secret
ibm-slapdMasterReferral: ldap://server2:389/
```

Note: The `ibm-slapdMasterDN` and `ibm-slapdMasterPW` values must match the values stored on the master server, server1, in the entry “cn=server2 BindCredentials”.

13. Stop server3.

```
ibmdirctl -h server3 -D <adminDN> -w <adminPW> -p <port> stop
```

14. Save the `ibmslapd.conf` file.
15. At the machines where server2 and server3 are located, issue the following command:

```
idsldif2db -r no -i <masterfile.ldif> -I <instance_name>
```

16. Start the master (server1), the forwarder (server2) and the replica (server3). On each of the servers issue the command:

```
idsslapd -I <LDAPinstance>
```

Note: Remember to ensure that all the servers have been added to the topology under cn=ibmpolicies.

Setting up a complex topology with peer replication

Initially, the **ibm-replicagroup** object created by this process inherits the ACL of the root entry for the replicated subtree. These ACLs might be inappropriate for controlling access to the replication information in the directory.

For the Add subtree operation to be successful, the entry DN which you are adding must have correct ACLs, if it is not a suffix in the server.

For Non-filtered ACLs :

```
ownsource : <the entry DN>
ownerpropagate : TRUE
aclsource : <the entry DN>
aclpropagate: TRUE
```

Filtered ACLs :

```
ownsource : <the entry DN>
ownerpropagate : TRUE
ibm-filteraclinherit : FALSE
ibm-filteraclentry : <any value>
```

Use the **Edit ACLs** function of the Web Administration Tool to set ACLs for the replication information associated with the newly created replicated subtree (see “Editing access control lists for the subtree” on page 224).

Using the forwarding topology created in “Changing the replica to a forwarding server” on page 187, you are going to create a peer-forwarder-replica topology consisting of two peer-master servers, two forwarding servers, and four replicas. To create this topology, you must:

1. Create two additional replica servers for the master server. See “Adding a replica server” on page 231.
2. Create two replicas under each of the two newly created replica servers.
3. Add a new peer master server. See “Adding a peer-master or gateway server” on page 228.

Note: The server that you want to promote to a master must be a leaf replica with no subordinate replicas.

4. Copy the data from the master to the new master and replicas. See “Copying data to the replica” on page 176.
5. Start replication. See “Managing queues” on page 240.

Using Web Administration:

You can create a server as a peer master. In this example you are going to start with the forwarding topology created in the previous procedure and create four new replicas and one server (server5) that is a peer to the master server (server1).

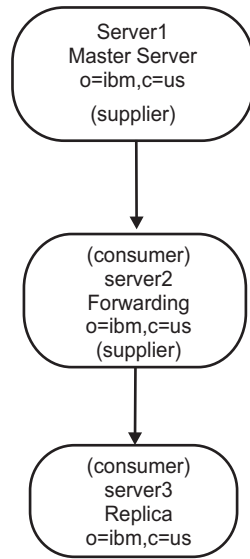


Figure 12. Initial master-forwarding server-replica topology

1. Start all the servers.
2. Use the Web Administration Tool to log on to the master (server1).
3. Expand the Replication management category in the navigation area and click **Manage topology**.
4. Select the subtree that you want to replicate and click **Show topology**.
5. Click the box next to the **server1** selection to expand the list of servers.
6. Click the box next to the **server2** selection to expand the list of servers.
7. Click **server1** and click **Add replica**. Create **server4**. See “Adding a replica server” on page 231 for information about creating replicas, adding credentials and supplier information. The server roles are represented by icons in the Web Administration Tool (see “Replication topology” on page 163). Your topology is now:
 - server1 (master)
 - server2 (forwarder)
 - server3 (replica)
 - server4 (replica)

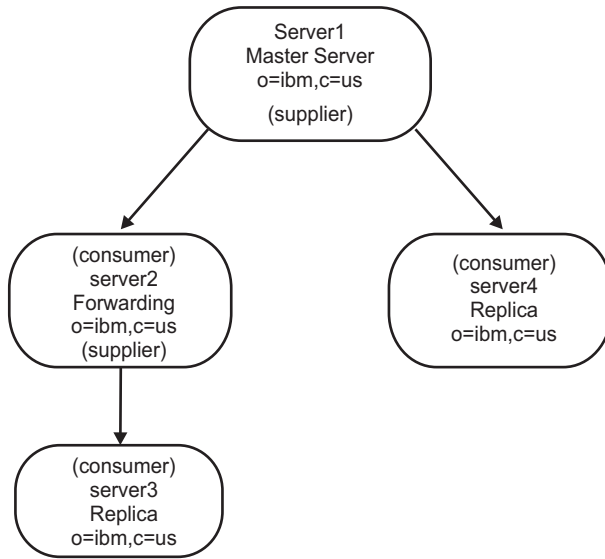


Figure 13. The initial topology with an additional replica

8. Click **server2** and click **Add replica** to create server6. See “Adding a replica server” on page 231 for information about creating replicas, adding credentials and supplier information.
9. Click **server4** and click **Add replica** to create server7. See “Adding a replica server” on page 231 for information about creating replicas, adding credentials and supplier information. Follow the same procedure to create server8. Your topology is now:
 - server1 (master)
 - server2 (forwarder)
 - server3 (replica)
 - server6 (replica)
 - server4 (forwarder)
 - server7 (replica)
 - server8 (replica)

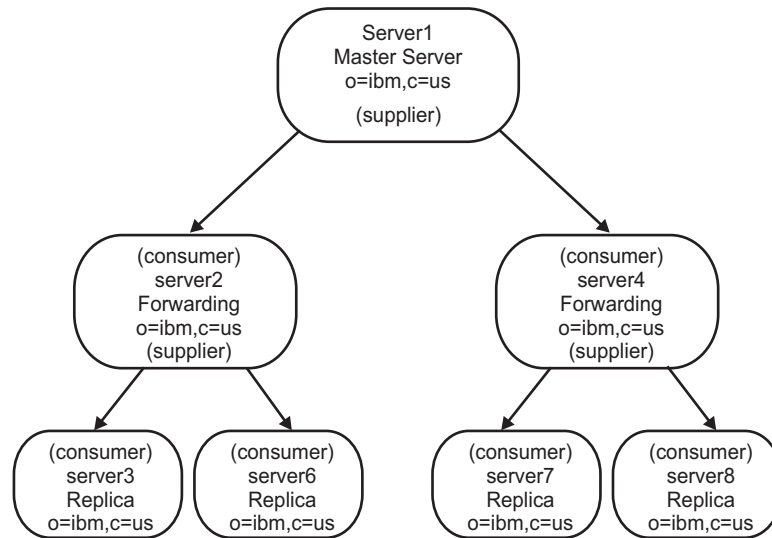


Figure 14. A more complex topology

10. Create a peer server. Select **server1**. Click **Add server**. See “Adding a peer-master or gateway server” on page 228 for information about how to create the server.
11. When the **Create additional supplier agreements** panel is displayed, deselect any agreements that are not appropriate for the server. Peer replication requires each master to be a supplier and consumer to each of the other masters in the topology and supplier to each of the first level replicas, server2 and server4. Server5 needs to be a consumer of server1 and a supplier to server1, server2, and server4. Ensure that the supplier agreement boxes are checked for:

Select	Supplier	Consumer
<input checked="" type="checkbox"/>	server5	server1
<input checked="" type="checkbox"/>	server1	server5
<input checked="" type="checkbox"/>	server5	server2
<input checked="" type="checkbox"/>	server5	server4

Click **Continue**.

12. Add the appropriate credentials and consumer information.

Notes:

- a. In some cases the Select credentials panel will pop up asking for a credential that is located in a place other than `cn=replication,cn=localhost`. In such situations you must provide a credential object that is located in a place other than `cn=replication,cn=localhost`. Select the credentials the subtree is going to use from the existing sets of credentials or create new credentials. See “Adding credentials” on page 224.
 - b. If the credentials are to be created in `cn=replication,cn=localhost`, the credentials must be created on each server after they are restarted. Replication by the peers fails until the credential objects are created. For this example, the credentials are being stored in `cn=replication,cn=ibmpolicies`. Make sure `cn=ibmpolicies` is replicated.
13. Click **OK**. Your topology is now:
 - server1 (master)

- server2 (forwarder)
 - server3 (replica)
 - server6 (replica)
- server4 (forwarder)
 - server7 (replica)
 - server8 (replica)
- server5 (master)
- server5 (master)
 - server1 (master)
 - server2 (forwarder)
 - server3 (replica)
 - server6 (replica)
 - server4 (forwarder)
 - server7 (replica)
 - server8 (replica)

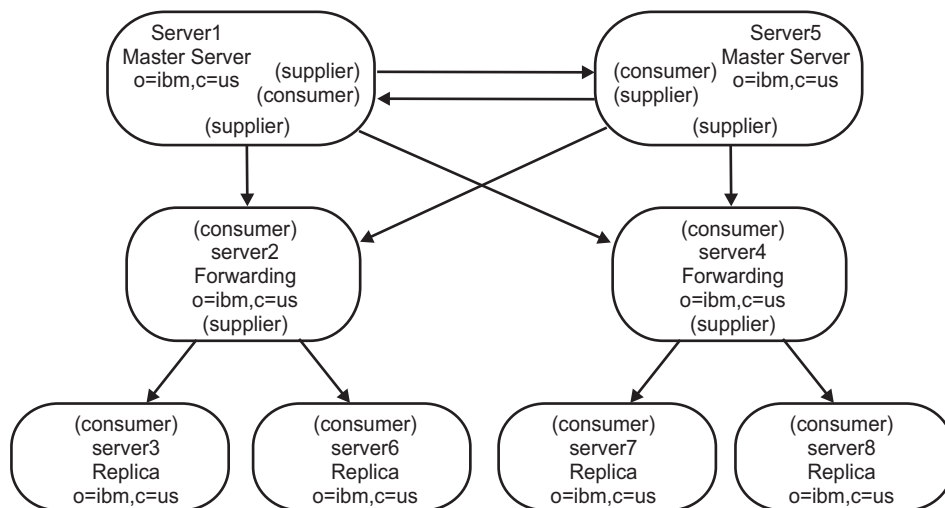


Figure 15. A peer-to-peer topology

14. Copy subtree directory data from server1 to all the new servers. See “Copying data to the replica” on page 176 for information about how to do that.
15. If you did not select **Add the consumer information** when you added the server to the topology, you must add the supplier credential information to each new server. See “Adding the supplier information to a replica” on page 236 for information about how to do that.
16. If it has not already started, start or resume replication. See “Starting replication” on page 177 for information about how to do that.

Using the command line:

This scenario assumes that you are creating a new replicated subtree and that only server1 contains any entry data. All other servers are newly installed and have a configured database.

Note:

```

dn: o=ibm,c=us
objectclass: organization
objectclass: ibm-replicationContext
o: ibm

```

is the subtree you want to create. If this entry already exists, then modify it to add **objectclass=ibm-replicationContext** instead of adding the entire entry.

In this example the topology is more complex. It includes two peer-masters (server1 and server5), two forwarders (server2 and server4) and four replicas (server3, server6, server7, and server8). The relationship among the servers is as follows:

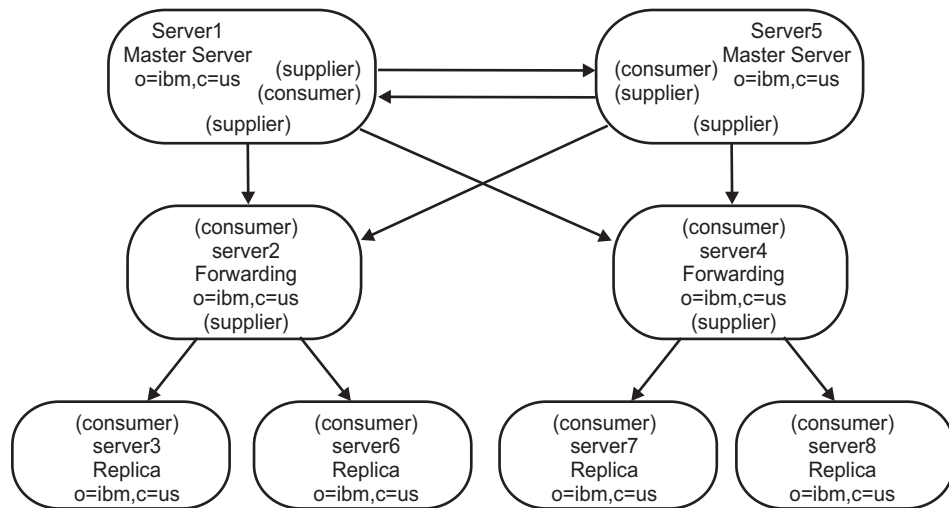


Figure 16. A peer-to-peer topology

- server1 and server5 are peer-master servers. That means that while they receive updates from each other, they only replicate entries received from clients. While both masters have the same entry content, only the server that has received the client request replicates the entry. Both masters are suppliers and consumers to each other and suppliers to the forwarding servers.
 - server2 and server4 have two roles. They are both consumers of server1 and server5 and suppliers to their respective replicas. They do not perform any client updates. They pass replicated updates to their consumers. In this scenario
 - server2 is a supplier to server3 and server6
 - server4 is a supplier to server7 and server8
- There is no interaction between server2 and server4.
- replica 1 and replica 2 are consumers of server2 and server7 and server8 are consumers of server4.

To create the peer-masters (server1 and server5), the forwarders (server2 and server4), and the replicas (server3, server6, server7, and server8) for the subtree **o=ibm,c=us**:

1. Start servers server1 and server5 in configuration mode. On each of the servers issue the command:

```
idsslapd -I <LDAPinstance> -a
```

2. You must configure server1 and server5 to be peer servers. Use the **idsldapadd** command to add the following entry to the **ibmslapd.conf** file on server1 and server5. On server1 and server5 issue the following command:

```
idsldapadd -D <adminDN> -w<adminPW> -i<filename>
```

where *<filename>* contains:

```
dn: cn=Master Server, cn=configuration
objectclass: ibm-slapdReplication
cn: Master Server
ibm-slapdMasterDN: cn=master
ibm-slapdMasterPW: master
```

Note: It is critical that these entries be exactly the same on both servers because this example uses a credentials object that is shared on all the servers.

3. Stop server1 and server5. To stop the servers issue the following command on each of the servers:

```
ibmdirctl -h <serverx> -D <adminDN>-w <adminPW>-p 389 stop
```

where *<serverx>* is the name of the server.

4. Make sure you have a backup of the **ibmslapd.conf** file.
5. At the machine where the master server, server1, is located, create a file to contain the agreement information, for example *mycredentialsfile* where *mycredentialsfile* contains:

```
dn: cn=replication,cn=IBMpolicies
o: ibm
objectclass: top
objectclass: organization
objectclass: ibm-replicationContext
```

```
dn: ibm-replicaGroup=default, cn=replication,cn=IBMpolicies
objectclass: top
objectclass: ibm-replicaGroup
ibm-replicaGroup: default
```

```
dn: ibm-replicaServerId=<server1-uuid>,ibm-replicaGroup=default,
  cn=replication,cn=IBMpolicies
objectclass: top
objectclass: ibm-replicaSubentry
ibm-replicaServerId: <server1-uuid>
ibm-replicationServerIsMaster: true
cn: server1
description: server 1 (peer master) ibm-replicaSubentry
```

```
dn: ibm-replicaServerId=<server5-uuid>,ibm-replicaGroup=default,
  cn=replication,cn=IBMpolicies
objectclass: top
objectclass: ibm-replicaSubentry
ibm-replicaServerId: <server5-uuid>
ibm-replicationServerIsMaster: true
cn: server5
description: server5 (peer master) ibm-replicaSubentry
```

```
dn: ibm-replicaServerId=<server2-uuid>,
  ibm-replicaGroup=default,cn=replication,cn=IBMpolicies
objectclass: top
objectclass: ibm-replicaSubentry
ibm-replicaServerId: <server2-uuid>
ibm-replicationServerIsMaster: false
cn: server2
description: server2 (forwarder) ibm-replicaSubentry
```

```

dn: ibm-replicaServerId=<server4-uuid>,
    ibm-replicaGroup=default,cn=replication,cn=IBMpolicies
objectclass: top
objectclass: ibm-replicaSubentry
ibm-replicaServerId: <server4-uuid>
ibm-replicationServerIsMaster: false
cn: server4
description: server2 (forwarder) ibm-replicaSubentry

#server1 to server5 agreement
dn: cn=server5,ibm-replicaServerId=<server1-uuid>,
    ibm-replicaGroup=default,cn=replication,cn=IBMpolicies
objectclass: top
objectclass: ibm-replicationAgreement
cn: server5
ibm-replicaConsumerId: <server5-uuid>
ibm-replicaUrl: ldap://server5:389
ibm-replicaCredentialsDN: cn=simple,cn=replication,cn=IBMpolicies
description: server1(master) to server5(master) agreement

#server1 to server2 agreement
dn: cn=server2,ibm-replicaServerId=<server1-uuid>,
    ibm-replicaGroup=default,cn=replication,cn=IBMpolicies
objectclass: top
objectclass: ibm-replicationAgreement
cn: server2
ibm-replicaConsumerId: <server2-uuid>
ibm-replicaUrl: ldap://server2:389
ibm-replicaCredentialsDN: cn=simple,cn=replication,cn=IBMpolicies
description: server1(master) to server2(forwarder) agreement

#server1 to server4 agreement
dn: cn=server4,ibm-replicaServerId=<server1-uuid>,
    ibm-replicaGroup=default,cn=replication,cn=IBMpolicies
objectclass: top
objectclass: ibm-replicationAgreement
cn: server4
ibm-replicaConsumerId: <server4-uuid>
ibm-replicaUrl: ldap://server4:389
ibm-replicaCredentialsDN: cn=simple,cn=replication,cn=IBMpolicies
description: server1(master) to server4(forwarder) agreement

#server5 to server1 agreement
dn: cn=server1,ibm-replicaServerId=<server5-uuid>,
    ibm-replicaGroup=default,cn=replication,cn=IBMpolicies
objectclass: top
objectclass: ibm-replicationAgreement
cn: server1
ibm-replicaConsumerId: <server1-uuid>
ibm-replicaUrl: ldap://server1:389
ibm-replicaCredentialsDN: cn=simple,cn=replication,cn=IBMpolicies
description: server5(master) to server1(master) agreement

#server5 to server2 agreement
dn: cn=server2,ibm-replicaServerId=<server5-uuid>,
    ibm-replicaGroup=default,cn=replication,cn=IBMpolicies
objectclass: top
objectclass: ibm-replicationAgreement
cn: server2
ibm-replicaConsumerId: server2-uid
ibm-replicaUrl: ldap://server2:389
ibm-replicaCredentialsDN: cn=simple,cn=replication,cn=IBMpolicies
description: server5(master) to server2(forwarder) agreement

#server5 to server4 agreement

```

```
dn: cn=server4,ibm-replicaServerId=<server5-uuid>,
    ibm-replicaGroup=default,cn=replication,cn=IBMpolicies
objectclass: top
objectclass: ibm-replicationAgreement
cn: server4
ibm-replicaConsumerId: <server4-uuid>
ibm-replicaUrl: ldap://server4:389
ibm-replicaCredentialsDN: cn=simple,cn=replication,cn=IBMpolicies
description: server5(master) to server4(forwarder) agreement
```

```
#server2 to server3 agreement
dn: cn=server3,ibm-replicaServerId=<server2-uuid>,
    ibm-replicaGroup=default,cn=replication,cn=IBMpolicies
objectclass: top
objectclass: ibm-replicationAgreement
cn: server3
ibm-replicaConsumerId: <server3-uuid>
ibm-replicaUrl: ldap://server3:389
ibm-replicaCredentialsDN: cn=simple,cn=replication,cn=IBMpolicies
description: server2(forwarder) to server3(replica) agreement
```

```
#server2 to server6 agreement
dn: cn=server6,ibm-replicaServerId=<server2-uuid>,
    ibm-replicaGroup=default,cn=replication,cn=IBMpolicies
objectclass: top
objectclass: ibm-replicationAgreement
cn: server6
ibm-replicaConsumerId: <server6-uuid>
ibm-replicaUrl: ldap://server6:389
ibm-replicaCredentialsDN: cn=simple,cn=replication,cn=IBMpolicies
description: server2(forwarder) to server6(replica) agreement
```

```
#server4 to server7 agreement
dn: cn=server7,ibm-replicaServerId=<server4-uuid>,
    ibm-replicaGroup=default,cn=replication,cn=IBMpolicies
objectclass: top
objectclass: ibm-replicationAgreement
cn: server7
ibm-replicaConsumerId: <server7-uuid>
ibm-replicaUrl: ldap://server7:389
ibm-replicaCredentialsDN: cn=simple,cn=replication,cn=IBMpolicies
description: server4(forwarder) to server7(replica) agreement
```

```
#server4 to server8 agreement
dn: cn=server8,ibm-replicaServerId=<server4-uuid>,
    ibm-replicaGroup=default,cn=replication,cn=IBMpolicies
objectclass: top
objectclass: ibm-replicationAgreement
cn: server8
ibm-replicaConsumerId: <server8-uuid>
ibm-replicaUrl: ldap://server8:389
ibm-replicaCredentialsDN: cn=simple,cn=replication,cn=IBMpolicies
description: server4(forwarder) to server8(replica) agreement
```

6. Issue the command:

```
idsldif2db -r no -i <mycredentialsfile> -I <instance_name>
```

7. Stop server2 and server4. To stop the servers issue the following command on each of the servers:

```
ibmdirctl -h <serverx> -D <adminDN>-w <adminPW>-p 389 stop
```

where <serverx> is the name of the server.

8. Copy <mycredentialsfile> to the machines where server5, server2, and server4 are located and issue the command:

```
idsldif2db -r no -i <mycredentialsfile> -I <instance_name>
```

on each machine.

9. At the machine where server1 is located create a file *<mytopologyfile>* where *<mytopologyfile>* includes:

Note: Replace all occurrences of *<master-uuid>* in the following files with the value of the **ibm-slappedServerId** attribute from the master server's **cn=Configuration** entry. This value is generated by the server the first time it is started. You can find it either by performing an **idsldapsearch** of the **cn=Configuration** entry or using the **grep** command on the **ibmslapd.conf** file, if you have a UNIX-based system. Similarly, all occurrences of the *<serverx-uuid>* (where x represents a number) must be replaced with the value of the **ibm-slappedServerId** attribute from the respective server's **cn=Configuration** entry.

```
dn: o=ibm,c=us
o: ibm
objectclass: top
objectclass: organization
objectclass: ibm-replicationContext
```

```
dn: ibm-replicaGroup=default, o=ibm,c=us
objectclass: top
objectclass: ibm-replicaGroup
ibm-replicaGroup: default
```

```
dn: ibm-replicaServerId=<server1-uuid>, ibm-replicaGroup=default, o=ibm,c=us
objectclass: top
objectclass: ibm-replicaSubentry
ibm-replicaServerId: <server1-uuid>
ibm-replicationServerIsMaster: true
cn: server1
description: server 1 (peer master) ibm-replicaSubentry
```

```
dn: ibm-replicaServerId=<server5-uuid>, ibm-replicaGroup=default, o=ibm,c=us
objectclass: top
objectclass: ibm-replicaSubentry
ibm-replicaServerId: <server5-uuid>
ibm-replicationServerIsMaster: true
cn: server5
description: server5 (peer master) ibm-replicaSubentry
```

```
dn: ibm-replicaServerId=<server2-uuid>,
   ibm-replicaGroup=default, o=ibm,c=us
objectclass: top
objectclass: ibm-replicaSubentry
ibm-replicaServerId: <server2-uuid>
ibm-replicationServerIsMaster: false
cn: server2
description: server2 (forwarder) ibm-replicaSubentry
```

```
dn: ibm-replicaServerId=<server4-uuid>,
   ibm-replicaGroup=default, o=ibm,c=us
objectclass: top
objectclass: ibm-replicaSubentry
ibm-replicaServerId: <server4-uuid>
ibm-replicationServerIsMaster: false
cn: server4
description: server2 (forwarder) ibm-replicaSubentry
```

```
#server1 to server5 agreement
dn: cn=server5, ibm-replicaServerId=<server1-uuid>,
   ibm-replicaGroup=default, o=ibm,c=us
objectclass: top
objectclass: ibm-replicationAgreement
cn: server5
```

```
ibm-replicaConsumerId: <server5-uuid>
ibm-replicaUrl: ldap://server5:389
ibm-replicaCredentialsDN: cn=simple,cn=replication,cn=IBMpolicies
description: server1(master) to server5(master) agreement
```

```
#server1 to server2 agreement
dn: cn=server2,ibm-replicaServerId=<server1-uuid>,
    ibm-replicaGroup=default,o=ibm,c=us
objectclass: top
objectclass: ibm-replicationAgreement
cn: server2
ibm-replicaConsumerId: <server2-uuid>
ibm-replicaUrl: ldap://server2:389
ibm-replicaCredentialsDN: cn=simple,cn=replication,cn=IBMpolicies
description: server1(master) to server2(forwarder) agreement
```

```
#server1 to server4 agreement
dn: cn=server4,ibm-replicaServerId=<server1-uuid>,
    ibm-replicaGroup=default,o=ibm,c=us
objectclass: top
objectclass: ibm-replicationAgreement
cn: server4
ibm-replicaConsumerId: <server4-uuid>
ibm-replicaUrl: ldap://server4:389
ibm-replicaCredentialsDN: cn=simple,cn=replication,cn=IBMpolicies
description: server1(master) to server4(forwarder) agreement
```

```
#server5 to server1 agreement
dn: cn=server1,ibm-replicaServerId=<server5-uuid>,
    ibm-replicaGroup=default,o=ibm,c=us
objectclass: top
objectclass: ibm-replicationAgreement
cn: server1
ibm-replicaConsumerId: <server1-uuid>
ibm-replicaUrl: ldap://server1:389
ibm-replicaCredentialsDN: cn=simple,cn=replication,cn=IBMpolicies
description: server5(master) to server1(master) agreement
```

```
#server5 to server2 agreement
dn: cn=server2,ibm-replicaServerId=<server5-uuid>,
    ibm-replicaGroup=default,o=ibm,c=us
objectclass: top
objectclass: ibm-replicationAgreement
cn: server2
ibm-replicaConsumerId: server2-uid
ibm-replicaUrl: ldap://server2:389
ibm-replicaCredentialsDN: cn=simple,cn=replication,cn=IBMpolicies
description: server5(master) to server2(forwarder) agreement
```

```
#server5 to server4 agreement
dn: cn=server4,ibm-replicaServerId=<server5-uuid>,
    ibm-replicaGroup=default,o=ibm,c=us
objectclass: top
objectclass: ibm-replicationAgreement
cn: server4
ibm-replicaConsumerId: <server4-uuid>
ibm-replicaUrl: ldap://server4:389
ibm-replicaCredentialsDN: cn=simple,cn=replication,cn=IBMpolicies
description: server5(master) to server4(forwarder) agreement
```

```
#server2 to server3 agreement
dn: cn=server3,ibm-replicaServerId=<server2-uuid>,
    ibm-replicaGroup=default,o=ibm,c=us
objectclass: top
objectclass: ibm-replicationAgreement
cn: server3
```



```
ibm-replicaConsumerId: <server3-uuid>
ibm-replicaUrl: ldap://server3:389
ibm-replicaCredentialsDN: cn=simple,cn=replication,cn=IBMpolicies
description: server2(forwarder) to server3(replica) agreement
```

```
#server2 to server6 agreement
dn: cn=server6,ibm-replicaServerId=<server2-uuid>,
    ibm-replicaGroup=default,o=ibm,c=us
objectclass: top
objectclass: ibm-replicationAgreement
cn: server6
ibm-replicaConsumerId: <server6-uuid>
ibm-replicaUrl: ldap://server6:389
ibm-replicaCredentialsDN: cn=simple,cn=replication,cn=IBMpolicies
description: server2(forwarder) to server6(replica) agreement
```

```
#server4 to server7 agreement
dn: cn=server7,ibm-replicaServerId=<server4-uuid>,
    ibm-replicaGroup=default,o=ibm,c=us
objectclass: top
objectclass: ibm-replicationAgreement
cn: server7
ibm-replicaConsumerId: <server7-uuid>
ibm-replicaUrl: ldap://server7:389
ibm-replicaCredentialsDN: cn=simple,cn=replication,cn=IBMpolicies
description: server4(forwarder) to server7(replica) agreement
```

```
#server4 to server8 agreement
dn: cn=server8,ibm-replicaServerId=<server4-uuid>,
    ibm-replicaGroup=default,o=ibm,c=us
objectclass: top
objectclass: ibm-replicationAgreement
cn: server8
ibm-replicaConsumerId: <server8-uuid>
ibm-replicaUrl: ldap://server8:389
ibm-replicaCredentialsDN: cn=simple,cn=replication,cn=IBMpolicies
description: server4(forwarder) to server8(replica) agreement
```

10. To load this topology, issue the command:

```
idsldif2db -r no -i <mytopologyfile> -I <instance_name>
```

where `-r no` prevents replication of the set of entries.

11. At this point you might want to load additional data for your subtree.
12. When you have finished loading the data, to be able to export the topology to populate the other servers, issue the command:

```
idsdb2ldif -s"o=ibm,c=us" -o <mymasterfile.ldif> -I <instance_name>
-k <key seed> -t <key salt>
```

Note: You must use the `-I` option if there is more than one instance. You must use the `-k` and `-t` options if keys on the server are not in sync.

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*, see 177.

See “idsdb2ldif, db2ldif” on page 436 for more information.

13. Start server2, server3, server4, server6, server7, and server8 in configuration only mode. On each of the servers issue the command:

```
idsslapd -I <LDAPinstance> -a
```

14. You must configure server2 and server4 to be forwarding servers and configure server3, server6, server7, and server8 to be replica servers. Use the **idsldapadd** command to add the following entry to the **ibmslapd.conf** file on each of the servers:

```
idsldapadd -D <adminDN> -w<adminPW> -p <port> -i<filename>
```

where <filename> contains:

```
dn: cn=Master Server, cn=configuration
objectclass: ibm-slapdReplication
cn: Master Server
ibm-slapdMasterDN: cn=any
ibm-slapdMasterPW: secret
ibm-slapdMasterReferral: ldap://server1:389/
```

Note: This ensures that all updates from the clients are referred to server1.

15. Stop server2, server3, server4, server6, server7, and server8. To stop the servers issue the following command on each of the servers:

```
ibmdirctl -h <serverx> -D <adminDN>-w <adminPW> -p <port> stop
```

where <serverx> is the name of the server.

16. Save the **ibmslapd.conf** file as a new backup.
17. Copy <mymasterfile.ldif> to the machines where server2, server3, server4, server5, server6, server7, and server8 are located.
18. At each of these machines, issue the following command:

```
idsldif2db -r no -i <masterfile.ldif> -I <instance_name>
```

19. Start server1, server2, server3, server4, server5, server6, server7, and server8. On each of the servers issue the command:

```
idsslapd -I <instance_name>
```

Summary of steps for creating a complex replication topology

Use this high level overview as a guide for setting up a complex replication topology.

1. Start or place all of the potential peer masters and replicas-to-be in Configuration only mode.
2. Start the 'first' master and configure it as a master for the context.
3. Load the data for the subtree to be replicated on the 'first' master, if the data is not already loaded.
4. Select the subtree to be replicated.
5. Add all of the potential peer masters as replicas of the 'first' master.
6. Add all of the other replicas.
7. Add replica agreements for the replicas to each of the peer masters.

Note: If the credentials are to be created in **cn=replication,cn=localhost**, the credentials must be created on each server after they are restarted. Replication by the peers fails until the credential objects are created. This also applies to **cn=replication,c=ibmpolicies** if **cn=ibmpolicies** is not replicated.

8. Add replica agreements for the other masters to each of the peer masters. The 'first' master already has that information.
9. Quiesce the replicated subtree.
10. Use Queue management to skip all for each queue.

11. Export the data for the replicated subtree from the 'first' master.
12. Unquiesce the subtree.
13. Import the data for the replicated subtree on to each replica and peer master.
14. Manage the replication properties on each replica and peer master to set the credentials to be used by suppliers.
15. Restart the replicas and peer masters as soon as each is ready.

Unconfiguring a master/replica configuration

There are several ways to remove a replica server from a master (supplier)/replica (consumer) topology.

Use the following command to remove all master/replica information by unconfiguring the ldap server's database on both machines and reconfiguring:

```
idsucfgdb -I <instance_name>
```

A message box will display, asking you if you want to remove the database and the database instance. Click **Yes**.

Note: This process unconfigures the entire database on the replica server and all data will be lost.

Alternately, use the following steps to remove your replica from the topology. With this option, you are required to unconfigure and reconfigure one server only (replica):

1. Stop the replica server.
2. Suspend the master server.
3. Remove supplier information from your master server. Go to **Replication management**→ **Manage topology**.
4. Delete a replica server.
 - a. Click **Show topology**.
 - b. Select a replica.
 - c. Click **Delete**.
5. Delete a master server.
 - a. Click **Show topology**.
 - b. Select a master.
 - c. Click **Delete**.
6. Remove a subtree from master server.
 - a. Click **Show topology**.
 - b. Select a subtree.
 - c. Select **Delete subtree** from the drop-down list.
 - d. Click **Go**.
7. Remove credentials from a master server.
 - a. Click **Manage credentials**.
 - b. Select a subtree.
 - c. Click **Show credentials**.
 - d. Select credentials.
 - e. Click **Delete**.
 - f. Click **OK**.

8. Run the following command on the replica server to unconfigure the database and remove all data:

```
idsucfgdb -I <instance_name>
```

A message box will display, asking you if you want to remove the database and the database instance. Click **Yes**. All information or entries will be lost in each of your databases.

You can also do the following to unconfigure your replica server without unconfiguring your entire database:

1. Remove supplier information from your master server. Go to **Replication management**→ **Manage topology**.
2. Delete replica server.
 - a. Click **Show topology**.
 - b. Select a replica.
 - c. Click **Delete**.
3. Delete master server.
 - a. Click **Show topology**.
 - b. Select a master.
 - c. Click **Delete**.
4. Remove subtree from master server.
 - a. Click **Show topology**.
 - b. Select a subtree.
 - c. Select **Delete subtree** from the drop-down list.
 - d. Click **Go**.
5. Remove credentials from the master server.
 - a. Click **Manage credentials**.
 - b. Select a subtree.
 - c. Click **Show credentials**.
 - d. Select credentials.
 - e. Click **Delete**.
 - f. Click **OK**.
6. Remove the credentials from the replica server.
 - a. Click **Manage credentials**.
 - b. Select a subtree.
 - c. Click **Show credentials**.
 - d. Select credentials.
 - e. Click **Delete**.
 - f. Click **OK**.
7. Remove supplier information from your replica server. Click **Manage replication properties**. Click **Delete**.
8. Go to **Directory management**.
9. Select the subtree and expand.
10. Select **ibm-replica Group=default** and expand.
11. Select the **replicaSubentry** entry and expand.
12. Delete all agreements.
13. Collapse and delete **replicaSubentry** entry.

14. Collapse and delete **ibm-replica Group=default**.
15. Select the subtree. From the drop-down list, select **Delete auxiliary objectclass** and click **Go**.
16. A new panel is displayed. In this panel, select the **ibm-replicationContext** and click **Delete**.
17. Click **OK**.
18. Confirm your server no longer has replication information by performing the following searches on the replica server. Nothing should be returned for the second search. If an empty container is returned for the first search, that is acceptable.

```
idsldapsearch -D cn=root -w secret -b " " -s sub
objectclass=ibm-repl*
```

This operation will return any replication topology that remains in the directory.

Note: You can perform this step on the master if there are no replicas left in the topology.

Setting up a gateway topology

Note: A gateway server must be an IBM Tivoli Directory Server 5.2 or later, or an IBM Directory Server version 5.1 FP 1 server that supports gateway replication.

Gateway replication uses gateway servers to collect and distribute replication information effectively across a replicating network. The primary benefit of gateway replication is the reduction of network traffic.

Gateway servers must be masters (writable). The following figure illustrates how gateway replication works:

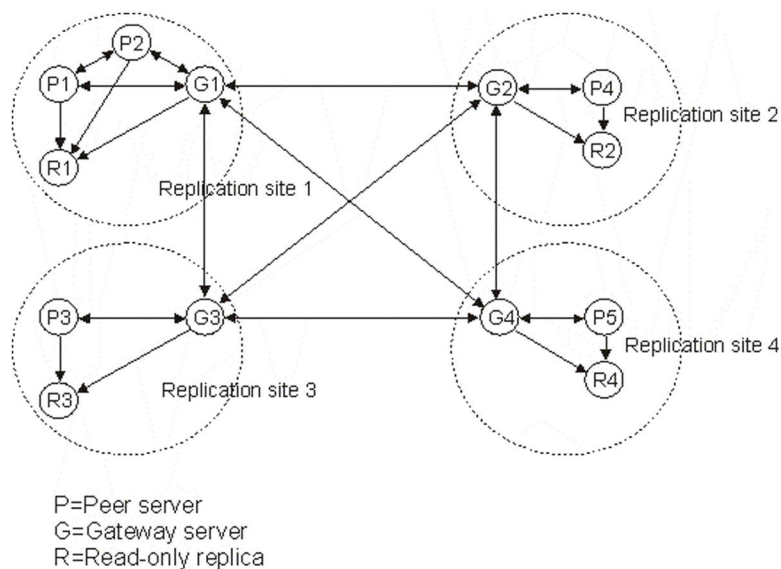


Figure 17. A replicating network with Gateway servers

The replicating network in the preceding figure contains four replication sites, each containing a gateway server. A gateway server:

- Collects replication updates from the peer/master servers in the replication site where it resides and sends the updates to all the other gateway servers within the replicating network.
- Collects replication updates from other gateway servers in the replication network and sends those updates to the peers/masters and replicas in the replication site where it resides.

Gateway servers use server IDs and consumer IDs to determine which updates are sent to other gateway servers in the replicating network and which updates are sent to local servers within the replication site.

To set up gateway replication, you must create at least two gateway servers. The creation of a gateway server establishes a replication site. You must then create replication agreements between the gateway and any masters/peers and replicas you want to include in that gateway's replication site.

Gateway servers must be masters (writable). If you attempt to add the gateway object class, `ibm-replicaGateway`, to a subentry that is not a master, an error message is returned.

There are two methods for creating a gateway server. You can:

- Create a new gateway server
- Convert an existing master server to a gateway server

Note: It is very important that you assign only one gateway server per replication site. The master and replica servers within the replication site can only have agreements with the gateway server for that site.

Using Web Administration:

Note: Before starting to set up your replication topology, make a backup copy of your original configuration file (`ibmslapd.conf`) and the key stash files (`ibmslapddir.ksf` and `ibmslapdconf.ksf`) `ibmslapd.conf` file. You can use this backup copy to restore your original configuration if you encounter difficulties with replication. In addition you need to save the replication topology information stored in the directory. Use the `idsdb2ldif` utility to export the `ibm-replicagroup=default` subtree of the replicated subtree. For example, if you are changing the topology for the subtree `o=ibm,c=us`, you need to export the subtree `ibm-replicagroup=default,o=ibm,c=us`.

Attention: If restoring, you must restore to the same operating system as the operating system on which the failure occurred. If you don't restore to the same operating system, there might be errors.

To set up a gateway using the complex topology with peer replication from the previous scenario:

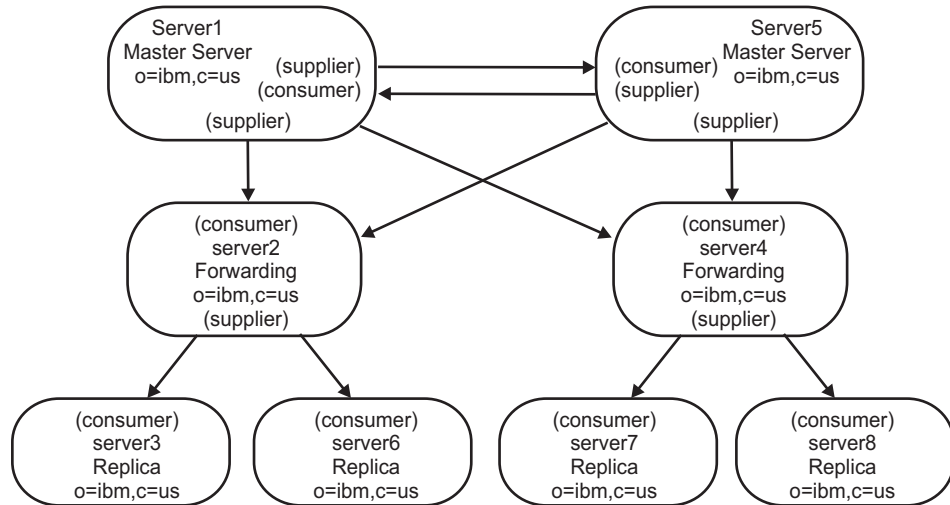


Figure 18. Initial peer-to-peer topology

- Convert an existing peer server (peer1) to a Gateway server to create replication site1.
- Create a new gateway server for replication site 2 and agreements with peer1.
- Create the topology for replication site 2 (not illustrated in this example).
- Copy the data from the master to all the machines in the topology.
 1. Use the Web Administration Tool to log on to the master (server1).
 2. Expand the Replication management category in the navigation area and click **Manage topology**.
 3. Select the subtree that you want and click **Show topology**.
 4. To convert an existing server to a gateway server, click **Manage gateway servers**. Select **server1** or its peer **server5**. For this example use **server1** and click Make gateway.
 5. Click **OK**.

Note: If the server you want to use as a gateway is not already a master, it must be a leaf replica with no subordinate replicas that you can first promote to be a master and then designate as a gateway.

6. To create a new gateway server, Click **Add server**.
7. Create the new server, **server9** as a gateway server. See “Adding a peer-master or gateway server” on page 228 for information about how to do that.
8. The **Create additional supplier agreements panel** is displayed. Ensure that the supplier agreement boxes are checked for server1 only. Deselect the other agreements.

Select	Supplier	Consumer
✓	server1	server9
✓	server9	server1
	server2	server9
	server9	server2
	server4	server9
	server9	server4
	server9	server5

Select	Supplier	Consumer
	server5	server9

Click **Continue**.

9. Click **OK**.
10. Add the appropriate credentials and consumer information.

Note: In some cases the Select credentials panel will pop up asking for a credential that is located in a place other than `cn=replication,cn=localhost`. In such situations you must provide a credential object that is located in a place other than `cn=replication,cn=localhost`. Select the credentials the subtree is going to use from the existing sets of credentials or create new credentials. See "Adding credentials" on page 224.

11. Click **OK**. The server roles are represented by icons in the Web Administration Tool. Your topology is now:
 - server1 (master-gateway for replication site1)
 - server2 (forwarder)
 - server3 (replica)
 - server6 (replica)
 - server4 (forwarder)
 - server7 (replica)
 - server8 (replica)
 - server5 (master)
 - server9 (master-gateway for replication site 2)
 - server5 (master)
 - server1 (master)
 - server2 (forwarder)
 - server3 (replica)
 - server6 (replica)
 - server4 (forwarder)
 - server7 (replica)
 - server8 (replica)
 - server9 (master-gateway)
 - server1 (master-gateway)

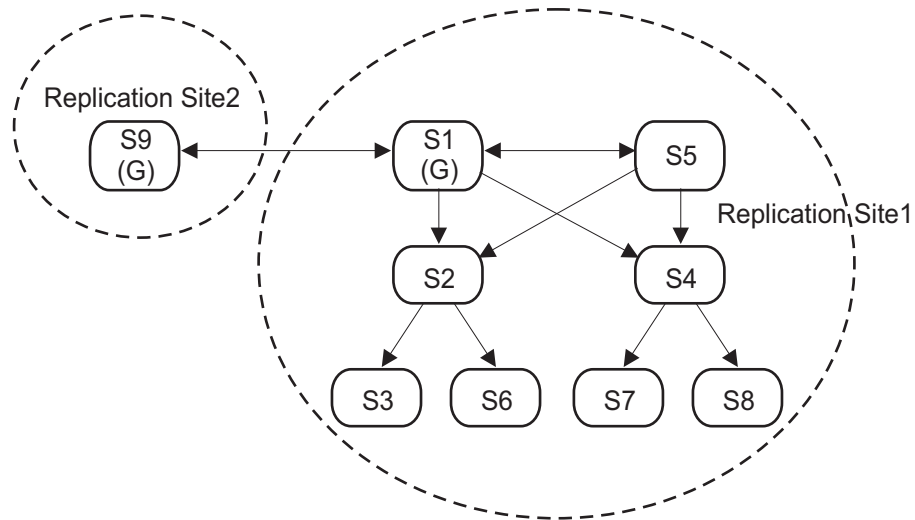


Figure 19. A gateway topology with two replication sites

12. Add servers to **server9** to create the topology for replication site 2. Remember to deselect any agreement for the new servers to any servers outside of replication site 2.
13. Repeat this process to create additional replication sites. Remember to create only one gateway server per replication site. However, each gateway server must be present in the topologies with agreements to the other gateway servers.
14. When you have finished creating the topology, copy the data from server1 to the all the new servers in all the replication sites and if required, add the supplier credential information to all the new servers. See “Copying data to the replica” on page 176 and “Adding the supplier information to a replica” on page 236 for information about how to do that.

Using the command line:

In this example you are going to change the previous two peer, two forwarder, and four replica scenario to:

- Change the role of server1 to a gateway server for its topology (replication site1).
- Create a new gateway server, server9, for replication site2.

Note: Replication site2 has its own topology with server9 as its gateway server. That replication topology is not being illustrated in this example. You can use the topology for replication site1 as a model. However, all the topology does need to be included for all replication sites in your actual topology setup.

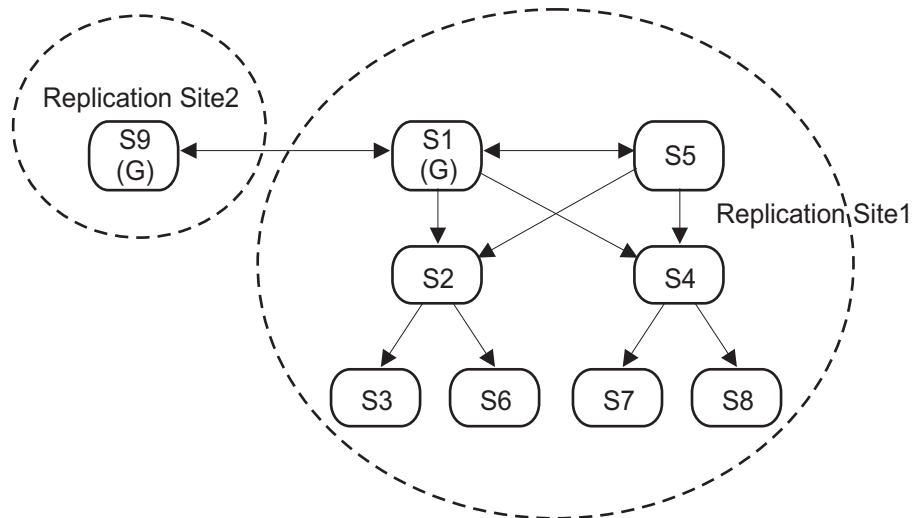


Figure 20. A gateway topology with two replication sites

1. Create server9. Create an instance (see "Creating and administering instances" in the *IBM Tivoli Directory Server Version 6.0 Installation and Configuration Guide*) for server9.

Note: Remember the server ID for this instance. You will use it in this task.

2. Configure server9 as a consumer of server1. Use the `idsldapmodify` command to add the following entry to the `ibmslapd.conf` file on server9:

```
idsldapmodify -D <adminDN> -w<adminPW> -p <port> -i <filename>
```

where `<filename>` contains:

```
dn: cn=Master Server, cn=configuration
objectclass: ibm-slapdReplication
cn: Master Server
ibm-slapdMasterDN: cn=any
ibm-slapdMasterPW: secret
```

3. Make server1 a gateway. Modify the following entry on server1 by adding the `objectclass: ibm-replicaGateway` attribute:

```
dn: ibm-replicaServerId=<server1-uuid>,ibm-replicaGroup=default,
ou=test,o=ibm,c=us
objectclass: top
objectclass: ibm-replicaSubentry
objectclass: ibm-replicaGateway
ibm-replicaServerId: <server1-uuid>
ibm-replicationServerIsMaster: true
cn: server1
description: server1 (gateway server from replication site 1 to
replication site 2)
```

4. Add the server9 subentry to server1:

```
dn: ibm-replicaServerId=<server9-uuid>,ibm-replicaGroup=default,
ou=test,o=ibm,c=us
objectclass: top
objectclass: ibm-replicaSubentry
objectclass: ibm-replicaGateway
ibm-replicaServerId: <server9-uuid>
ibm-replicationServerIsMaster: true
cn: server9
description: server9 (gateway server from replication site 2 to
replication site 1)
```

5. Suspend the server5 to server1 queue:

```
idsldapexop -D <adminDN> -w <admin_password> -h server5 -p <port> -op controlrepl
-action suspend -rc "ou=test,o=ibm,c=us"
```

6. Add the replication agreement from server9 to server1 on server1:

```
#server9 to server1 agreement
dn: cn=server1,ibm-replicaServerId=<server9-uuid>,
    ibm-replicaGroup=default,ou=test,o=ibm,c=us
objectclass: top
objectclass: ibm-replicationAgreement
cn: server1
ibm-replicaConsumerId: <server1-uuid>
ibm-replicaUrl: ldap://server1:389
ibm-replicaCredentialsDN: cn=simple,cn=replication,cn=IBMPolicies
description: supplier agreement from replication site2 to replication site 1
```

7. Add the replication agreement from server1 to server9 on server1:

```
#server1 to server9 agreement
dn: cn=server9,ibm-replicaServerId=<server1-uuid>,
    ibm-replicaGroup=default,ou=test,o=ibm,c=us
objectclass: top
objectclass: ibm-replicationAgreement
cn: server9
ibm-replicaConsumerId: <server9-uuid>
ibm-replicaUrl: ldap://server9:389
ibm-replicaCredentialsDN: cn=simple,cn=replication,cn=IBMPolicies
description: supplier agreement from replication site1 to replication site2
```

8. Quiesce server1:

```
idsldapexop -D <adminDN> -w <admin_password> -h server1 -p <port> -op quiesce
-rc "ou=test,o=ibm,c=us"
```

9. Flush the server1 to server9 queue:

```
idsldapexop -D <adminDN> -w <admin_password> -h server1 -p <port> -op controlqueue
-skip all -ra "cn=server9,ibm-replicaServerId=<server1-uuid>,
    ibm-replicaGroup=default,ou=test,o=ibm,c=us"
```

10. Perform an idsdb2ldif command to create an LDIF file on server1:

```
idsdb2ldif -s "ou=test,o=ibm,c=us" -o <filename1>.ldif
-I <instance_name> -k <key seed>
-t <key salt>
```

where <filename1>.ldif is the first LDIF file. For more information about file contents, see 214.

11. Perform an idsdb2ldif command to create a second LDIF file on server1:

```
idsdb2ldif -s "cn=replication,cn=ibmpolicies" -o <filename2>.ldif
-I <instance_name> -k <key seed>
-t <key salt>
```

where <filename2>.ldif is the second LDIF file. For more information about file contents, see 217.

12. Unquiesce server1:

```
idsldapexop -D <adminDN> -w <admin_password> -h server1 -p <port> -op
quiesce -end -rc "ou=test,o=ibm,c=us"
```

13. Resume the server5 to server1 queue on server5:

```
idsldapexop -D <adminDN> -w <admin_password> -h server5 -p <port> -op
controlrepl -action resume -rc "ou=test,o=ibm,c=us"
```

At this point, server5 and server1 are fully functional.

14. Copy the <filename1>.ldif file to server9.

15. Load the <filename1>.ldif onto server9:

```
idsldif2db -r no -i <filename1>.ldif -I <instance_name>
```

16. Copy the <filename2>.ldif file to server9.
17. Load the <filename2>.ldif onto server9:

```
idsldif2db -r no -i <filename2>.ldif -I <instance_name>
```
18. Start server9:

```
idsslapd -I <instance_name> -a
```

Note: If you want the global policy information replicated, remember to ensure that all the servers have been added to the topology under cn=ibmpolicies.

The following are partial file contents of both the first and second LDIF files loaded onto server9:

<filename1>.ldif

Note: The items in bold are the entries that were modified or added to create this Gateway topology.

```
dn: cn=ou=test,o=ibm,c=us
o: ibm
objectclass: top
objectclass: organization
objectclass: ibm-replicationContext

dn: ibm-replicaGroup=default,ou=test,o=ibm,c=us
objectclass: top
objectclass: ibm-replicaGroup
ibm-replicaGroup: default

#Make server1 a gateway server for site 1
dn: ibm-replicaServerId=<server1-uuid>,ibm-replicaGroup=default,
ou=test,o=ibm,c=us
objectclass: top
objectclass: ibm-replicaSubentry
objectclass: ibm-replicaGateway
ibm-replicaServerId: <server1-uuid>
ibm-replicationServerIsMaster: true
cn: server1
description: server1 (gateway server from replication site 1 to
replication site 2)

#Add server9 as a gateway server for site 2
dn: ibm-replicaServerId=<server9-uuid>,ibm-replicaGroup=default,
ou=test,o=ibm,c=us
objectclass: top
objectclass: ibm-replicaSubentry
objectclass: ibm-replicaGateway
ibm-replicaServerId: <server9-uuid>
ibm-replicationServerIsMaster: true
cn: server9
description: server9 (gateway server from replication site 2 to
replication site 1)

dn: ibm-replicaServerId=<server5-uuid>,ibm-replicaGroup=default,
ou=test,o=ibm,c=us
objectclass: top
objectclass: ibm-replicaSubentry
ibm-replicaServerId: <server5-uuid>
ibm-replicationServerIsMaster: true
cn: server5
description: server5 (master)

dn: ibm-replicaServerId=<server2-uuid>,
ibm-replicaGroup=default,ou=test,o=ibm,c=us
objectclass: top
```

objectclass: ibm-replicaSubentry
ibm-replicaServerId: <server2-uuid>
ibm-replicationServerIsMaster: false
cn: server2
description: server2 (forwarder server number one)

dn: ibm-replicaServerId=<server4-uuid>,
 ibm-replicaGroup=default,ou=test,o=ibm,c=us
objectclass: top
objectclass: ibm-replicaSubentry
ibm-replicaServerId: <server4-uuid>
ibm-replicationServerIsMaster: false
cn: server4
description: server4 (forwarder server number two)

#server1 to server9 agreement

dn: cn=server9,ibm-replicaServerId=<server1-uuid>,
 ibm-replicaGroup=default,ou=test,o=ibm,c=us
objectclass: top
objectclass: ibm-replicationAgreement
cn: server9
ibm-replicaConsumerId: <server9-uuid>
ibm-replicaUrl: ldap://server9:389
ibm-replicaCredentialsDN: cn=simple,cn=replication,cn=IBMPolicies
description: supplier agreement from replication site1 to replication site2

#server9 to server1 agreement

dn: cn=server1,ibm-replicaServerId=<server9-uuid>,
 ibm-replicaGroup=default,ou=test,o=ibm,c=us
objectclass: top
objectclass: ibm-replicationAgreement
cn: server1
ibm-replicaConsumerId: <server1-uuid>
ibm-replicaUrl: ldap://server1:389
ibm-replicaCredentialsDN: cn=simple,cn=replication,cn=IBMPolicies
description: supplier agreement from replication site2 to replication site 1

#server1 to server5 agreement

dn: cn=server5,ibm-replicaServerId=<server1-uuid>,
 ibm-replicaGroup=default,ou=test,o=ibm,c=us
objectclass: top
objectclass: ibm-replicationAgreement
cn: server5
ibm-replicaConsumerId: <server5-uuid>
ibm-replicaUrl: ldap://server5:389
ibm-replicaCredentialsDN: cn=simple,cn=replication,cn=IBMPolicies
description: server1 (gateway-master) to server5 (peer-master) agreement

#server1 to server2 agreement

dn: cn=server2,ibm-replicaServerId=<server1-uuid>,
 ibm-replicaGroup=default,ou=test,o=ibm,c=us
objectclass: top
objectclass: ibm-replicationAgreement
cn: server2
ibm-replicaConsumerId: <server2-uuid>
ibm-replicaUrl: ldap://server2:389
ibm-replicaCredentialsDN: cn=simple,cn=replication,cn=IBMPolicies
description: server1 (gateway-master) to server2 (forwarder) agreement

#server1 to server4 agreement

dn: cn=server4,ibm-replicaServerId=<server1-uuid>,
 ibm-replicaGroup=default,ou=test,o=ibm,c=us
objectclass: top
objectclass: ibm-replicationAgreement
cn: server4
ibm-replicaConsumerId: <server4-uuid>
ibm-replicaUrl: ldap://server4:389

```

ibm-replicaCredentialsDN: cn=simple,cn=replication,cn=IBMPolicies
description: server1 (gateway-master) to server4 (forwarder) agreement

#server5 to server1 agreement
dn: cn=server1,ibm-replicaServerId=<server5-uuid>,
    ibm-replicaGroup=default,ou=test,o=ibm,c=us
objectclass: top
objectclass: ibm-replicationAgreement
cn: server1
ibm-replicaConsumerId: <server1-uuid>
ibm-replicaUrl: ldap://server1:389
ibm-replicaCredentialsDN: cn=simple,cn=replication,cn=IBMPolicies
description: server5 (peer-master) to server1 (gateway-master) agreement

#server5 to server2 agreement
dn: cn=server2,ibm-replicaServerId=<server5-uuid>
    ibm-replicaGroup=default,ou=test,o=ibm,c=us
objectclass: top
objectclass: ibm-replicationAgreement
cn: server2
ibm-replicaConsumerId: server2-uid
ibm-replicaUrl: ldap://server2:389
ibm-replicaCredentialsDN: cn=simple,cn=replication,cn=IBMPolicies
description: server5 (peer-master) to server2 (forwarder) agreement

#server5 to server4 agreement
dn: cn=server4,ibm-replicaServerId=<server5-uuid>,
    ibm-replicaGroup=default,ou=test,o=ibm,c=us
objectclass: top
objectclass: ibm-replicationAgreement
cn: server4
ibm-replicaConsumerId: <server4-uuid>
ibm-replicaUrl: ldap://server4:389
ibm-replicaCredentialsDN: cn=simple,cn=replication,cn=IBMPolicies
description: server5 (peer-master) to server4 (forwarder) agreement

#server2 to server3 agreement
dn: cn=server3,ibm-replicaServerId=<server2-uuid>,
    ibm-replicaGroup=default,ou=test,o=ibm,c=us
objectclass: top
objectclass: ibm-replicationAgreement
cn: server3
ibm-replicaConsumerId: <server3-uuid>
ibm-replicaUrl: ldap://server3:389
ibm-replicaCredentialsDN: cn=simple,cn=replication,cn=IBMPolicies
description: server2 (forwarder) to server3 (replica)agreement

#server2 to server6 agreement
dn: cn=server6,ibm-replicaServerId=<server2-uuid>,
    ibm-replicaGroup=default,ou=test,o=ibm,c=us
objectclass: top
objectclass: ibm-replicationAgreement
cn: server6
ibm-replicaConsumerId: <server6-uuid>
ibm-replicaUrl: ldap://server6:389
ibm-replicaCredentialsDN: cn=simple,cn=replication,cn=IBMPolicies
description: server2 (forwarder) to server6 (replica)agreement

#server4 to server7 agreement
dn: cn=server7,ibm-replicaServerId=<server4-uuid>,
    ibm-replicaGroup=default,ou=test,o=ibm,c=us
objectclass: top
objectclass: ibm-replicationAgreement
cn: server7
ibm-replicaConsumerId: <server7-uuid>
ibm-replicaUrl: ldap://server7:389
ibm-replicaCredentialsDN: cn=simple,cn=replication,cn=IBMPolicies

```

```

description: server4 (forwarder) to server7 (replica)agreement

#server4 to server8 agreement
dn: cn=server8,ibm-replicaServerId=<server4-uuid>,
    ibm-replicaGroup=default,ou=test,o=ibm,c=us
objectclass: top
objectclass: ibm-replicationAgreement
cn: server8
ibm-replicaConsumerId: <server8-uuid>
ibm-replicaUrl: ldap://server8:389
ibm-replicaCredentialsDN: cn=simple,cn=replication,cn=IBMPolicies
description: server4 (forwarder) to server8 (replica)agreement

```

<filename2>.ldif

```

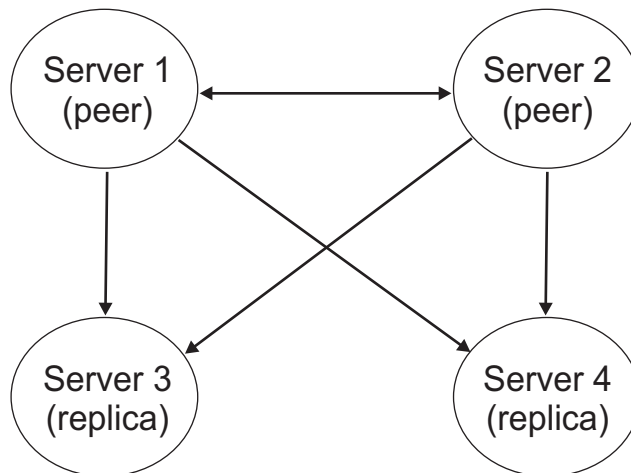
dn: cn=replication,cn=ibmpolicies
o: ibm
objectclass: top
objectclass: container
objectclass: ibm-replicationContext

dn: cn=simple,cn=replication,cn=ibmpolicies
objectclass: ibm-replicationCredentialsSimple
cn: simple
replicaBindDN: cn=any
replicaCredentials: secret

```

Recovery procedures

The following procedures are based on a system topology of IBM Tivoli Directory Server Version 6.0 servers with two peer master servers (server 1 and server 2) and two replica servers (server 3 and server 4). Server 2 is acting as a fail-over master, meaning that it does not accept updates directly from client machines unless server 1 is taken offline.



Required recovery information

After you have created your replication topology, you need to do the following:

1. Make a copy of the configuration file (ibmslapd.conf) and the key stash files (ibmslapddir.ksf and ibmslapdconf.ksf) of each server and store these files in a secure location. This location needs to be on a backup machine that is not part of the replication topology or on a separate media such as a diskette, CD, or tape. This information needs updating only if you change the topology or change your configuration parameters (any entries under cn=Configuration). If

you have made changes to the existing schema or added a new schema you need to make copies of the schema files (V3.* files) as well.

2. Use the **idsdbback** utility to create a nightly backup directory. Tar or zip this directory and store it in a secure location. This location needs to be on a backup machine that is not part of the replication topology or on a separate media such as a CD, or tape. This file contains all the entries in the directory, the server configuration information and the schema files. This backup directory ensures that you can never lose more than 24 hours worth of data. Run this utility against either server 3 or server 4 during off peak hours to get the most current data.

Attention: If restoring, you must restore to the same operating system as the operating system on which the failure occurred. If you don't restore to the same operating system, there might be errors.

Creating the database backup file

Use either the Configuration Tool or the command line utility to create your backup file.

Before you create the backup file, be sure that you have enough space to copy all the data. The space required is approximately the sum of the size of the following directories:

- `<dblocation>/<dbname>`
- `<dblocation>/ldap32kcont_<dbname>`

By default `<dblocation>` is the installation path of the database instance.

The server must be stopped before you can back up the database. To back up the database:

Using the Configuration Tool:

1. At a command prompt, type `idsxcfg -I <instance_name>` to start the Configuration Tool.
2. Click **Backup database** in the task list on the left.
3. In the Backup database window on the right, in the **Backup directory** field, type the directory path in which to back up all directory data and configuration settings. Alternatively, click **Browse** to locate the directory path. Make note of the exact directory path of the back up directory. This location is required for a successful restoration of data.
4. Click **Create backup directory as needed** if you want the directory to be created if it does not exist.
5. Click **Backup**.

Using the command line: On the server that you are using as the source server, if it does not already exist, create the backup directory, `<backupdir>`. Then issue the command:

```
idsdbback -k <backupdir> -I <instance_name>
```

Where `backupdir` is the name of the backup directory you are creating. Make note of the exact directory path of the back up directory. This location is required for a successful restoration of data.

After you have created the backup directory, tar or zip the directory and its contents and store it in a secure location. This location needs to be on a backup machine that is not part of the replication topology or on a separate media such as a CD or tape.

Restoring the database

Use either the Configuration Tool or the command line utility to restore your database and configuration information.

Copy the most current backup directory file to the server and untar or unzip it.

Note: This file must be copied to the exact location where the backup directory was originally created. Otherwise **idsdbrestore** fails.

The server must be stopped before you can back up the database. To restore the database:

Using the Configuration Tool:

1. At a command prompt, type `idsxcfg -I <instance_name>` to start the Configuration Tool.
2. Click **Restore database** in the task list on the left.
3. In the Restore database window on the right, in the **Backup directory** field, type the path in which the directory was previously backed up. Alternatively, click **Browse** to locate the path.
4. Select the **Restore data only (not configuration settings)** check box.
5. Click **Restore**.

Using the command line: On the server that you are restoring the data:

1. Issue the command:

```
idsdbrestore -k <backupdir> -I <instance_name> -n
```

Where *backupdir* is the name of the backup directory you are restoring from.

Your database and configuration information have been restored.

Recovering from a single-server failure

Use this procedure to restore a server that has been repaired, for example had the hard drive replaced. For this example, server 3 is the server that is going to be restored. Server 2 is the server that is going to be used to restore server 3.

Note: If the server is being replaced by a new machine, ensure that you use the same host name as the previous machine.

Attention: The following instructions assume you are recovering to the same operating system as the operating system on which the failure occurred. If you don't recover to the same operating system, there will be errors.

1. Install the IBM Tivoli Directory Server on server 3.
2. Configure a new database on server 3. Use the same instance owner name and database name that was previously used for server 3.
3. Copy the backup configuration file (`ibmslapd.conf`) and the key stash files (`ibmslapddir.ksf` and `ibmslapdconf.ksf`) for server 3 from the recovery source media on to server 3.

Note: If recovering, you must recover to the same operating system as the operating system on which the failure occurred. If you don't recover to the same operating system, there might be errors.

4. Quiesce server 1.
`idsldapexop -D <admin_dn> -w <admin_pw> -op quiesce -rc o=ibm,c=us`
5. Wait for server 1 to replicate all pending updates to server 2, when `ibm-replicationpendingchange`count is zero.
`idsldapsearch -D <admin_dn> -w <admin_pw> -h <server1> -b <dn of agreement with server2> -s base objectclass=* ibm-replicationpendingchange`
6. On server 1, purge the replication queue for server 3.
`idsldapexop -D <admin_dn> -w <admin_pw> -op controlqueue -skip all -ra <dn of agreement with server3>`
7. On server 1, clear all errors logged for replication with server 3.
`idsldapexop -D <admin_dn> -w <admin_pw> -op controlreplerr -delete all -ra <dn of agreement with server3>`
8. On server 1, suspend replication to server 2 and server 3.
`idsldapexop -D <admin_dn> -w <admin_pw> -op controlrepl -action suspend -ra <dn of agreement with server2>`
`idsldapexop -D -D <admin_dn> -w <admin_pw> -op controlrepl -action suspend -ra <dn of agreement with server3>`
9. Unquiesce server 1 so that it can accept updates again.
`idsldapexop -D <admin_dn> -w <admin_pw> -op quiesce -end -rc o=ibm,c=us`
10. Stop server 3.
11. Stop server 2.
12. Use DB2 backup to back up the data on server 2.
13. Start server 2, and resume its replication queue on server 1.
`idsldapexop -op controlrepl -action resume -ra <dn of agreement with server2>`
14. Restore the DB2 data on server 3.
15. Start server 3, and resume its replication queue on server 1.
`idsldapexop -op controlrepl -action resume -ra <dn of agreement with server3>`

Recovering from a catastrophic failure

Use this procedure, if all the servers in the topology are lost and are being replaced.

1. Ensure that the same host names are used on the new machines that were used on the previous ones.
2. Reinstall the IBM Tivoli Directory Server Version 6.0 on all the new servers.
3. Configure a new database on each of the servers. Use the same instance owner names and database names as before.
4. Ensure that all the servers are stopped.
5. Copy the most current backup directory files to each of the servers.

Note: This file must be copied to the exact location where the backup directory was originally created. Otherwise `idsdbrestore` fails.

6. Restore the database on each of the servers using the Configuration Tool or the `idsdbrestore` command. See "Restoring the database" on page 219.
7. Restart all the servers.

Your topology and data are restored to what they were less than 24 hours before the failure.

Multi-threaded replication

The multi-threaded replication function replaces the current single replication thread with a minimum of three threads to service the replication agreement:

- Main thread
- Sender thread
- Receiver thread

You can have anywhere from 1 to 32 consumer connections. Set the number of consumer connections to match the number of processors on your machine.

Using multiple threads enables the supplier to send the updates to the consumer without waiting on the response from the consumer.

Anyone with a replication backlog might consider switching to multi-threaded replication. Candidate environments can include the following:

- A high update rate
- No downlevel servers
- Common AES salt and synchronization if encryption is AES and passwords are updated often
- Small fanout (for example, don't try 8 connections per agreement with 24 replicas)
- Available servers and reliable network
- Data consistency is not critical
- All replication schedules are immediate
- Multiprocessor machines

Multi-threaded replication is difficult to administer if servers or networks are not reliable.

When errors occur, the errors are logged and can be replayed by the administrator, but the error logs must be monitored closely. The following is a search to show the replication backlog for all agreements supplied by one server:

```
idsldapsearch -h supplier-host -D cn=admin -w ? -s sub
  objectclass=ibm-replicationagreement
  ibm-replicationpendingchangeount ibm-replicationstate
```

If the replication state is active, and the pending count is growing, there is a backlog that won't decrease unless the update rate decreases.

Replication error table

The replication error table (REPLERROR) logs failing updates for later recovery. When replication starts, the number of failures logged for each replication agreement is counted. This count is incremented if an update results in a failure, and a new entry is added into the table.

Each entry in the replication error table contains the following:

- The replication agreement ID.
- The replication change ID.
- The timestamp for when the update was attempted.
- The number of attempts made (this value is 1 by default, and increments for each attempt made).

- The result code from the consumer.
- All of the information from the replication operation pertaining to the update, for example, the DN, the actual data, controls, flags, and so forth.

If the value specified by the attribute `ibm-slapdReplMaxErrors` in the server configuration is `0`, replication continues processing updates. The attribute `ibm-slapdReplMaxErrors` is an attribute in the replication configuration entry and it can be changed dynamically.

If the value specified by the attribute `ibm-slapdReplMaxErrors` is greater than `0`, then when the error count for a replication agreement exceeds this value, replication does one of the following:

Single threaded

Replication goes into a loop trying to replicate the failing update.

Multi-threaded

Replication is suspended.

If the server is configured to use a single connection, replication attempts to send the same update after waiting for 60 seconds and keeps trying until replication succeeds or the administrator skips the update.

For a server configured to use multiple connections, replication is suspended for this agreement. The receiver threads continue polling for status from any updates that have been sent, but no more updates are replicated. To resume replication, the directory administrator must clear at least one error for this agreement or increase the limit with a dynamic modification of the server configuration.

For more information, see "Replication error log extended operation" in *IBM Tivoli Directory Server C-Client SDK Programming Reference Version 6.0*.

Web Administration tasks for managing replication

Use the Web Administration Tool to perform the following tasks.

Replicating subtrees

Adding a subtree

Note: The server must be running to perform this task.

Expand the **Replication management** category in the navigation area and click **Manage topology**.

- Click **Add subtree**.
- Enter the DN of the subtree that you want to replicate or click **Browse** to expand the entries to select the entry that is to be the root of the subtree.
- Enter the master server referral URL in the form of an LDAP URL, for example:

For non-SSL:

```
ldap://<myservername>.<mylocation>.<mycompany>.com:<port>
```

For SSL:

```
ldaps://<myservername>.<mylocation>.<mycompany>.com:<port>
```

The default URL is `ldap://localhost:389`

Note: The master server referral URL is optional. It is used only:

- If the server contains (or will contain) any read-only subtrees.
- To define a referral URL that is returned for updates to any read-only subtree on the server.

- Click **OK**.
- The new server is displayed on the Manage topology panel under the heading **Replicated subtrees**.

Note: On the Linux, Solaris, and HP-UX platforms, if a referral fails, ensure that the environment variable `LDAP_LOCK_REC` has been set in your system environment. No specific value is required.

```
set LDAP_LOCK_REC=anyvalue
```

Editing a subtree

Use this option to change the URL of the master server that this subtree and its replicas send updates to. You need do this if you change the port number or host name of the master server, change the master to a different server

1. Select the subtree you want to edit.
2. Expand the **Select Action** menu, select **Edit subtree** and click **Go**.
3. Enter the master server referral URL. This must be in the form of an LDAP URL, for example:

```
ldap://<mynewservername>.<mylocation>.<mycompany>.com:<port>
```

Depending on the role being played by the server on this subtree (whether it is a master, replica or forwarding), different labels and buttons appear on the panel.

- When the subtree's role is replica, a label indicating that the server acts as a replica or forwarder is displayed along with the button **Make server a master**. If this button is clicked then the server which the Web Administration Tool is connected to becomes a master.
- When the subtree is configured for replication only by adding the auxiliary class (no default group and subentry present), then the label **This subtree is not replicated** is displayed along with the button **Replicate subtree**. If this button is clicked, the default group and the subentry is added so that the server with which the Web Administration Tool is connected to becomes a master.
- If no subentries for the master servers are found, the label **No master server is defined for this subtree** is displayed along with the button titled **Make server a master**. If this button is clicked, the missing subentry is added so that the server with which the Web Administration Tool is connected to becomes a master.

Removing a subtree

1. Select the subtree you want to remove
2. Expand the **Select Action** menu, select **Delete subtree** and click **Go**.
3. When asked to confirm the deletion, click **OK**.

The subtree is removed from the **Replicated subtree** list.

Note: This operation succeeds only if the `ibm-replicaGroup=default` is entry is empty.

Quiescing the subtree

This function is useful when you want to perform maintenance on or make changes to the topology. It minimizes or stops completely the number of updates

that can be made to the server. A quiesced server does not accept client requests. It accepts requests only from an administrator using the Server Administration control.

This function is Boolean.

1. Click **Quiesce/Unquiesce** to quiesce the subtree.
2. When asked to confirm the action, click **OK**.
3. Click **Quiesce/Unquiesce** to unquiesce the subtree.
4. When asked to confirm the action, click **OK**.

Editing access control lists for the subtree

Replication information (replica subentries, replication agreements, schedules, possibly credentials) are stored under a special object, **ibm-replicagroup=default**. The **ibm-replicagroup** object is located immediately beneath the root entry of the replicated subtree. By default, this subtree inherits ACL from the root entry of the replicated subtree. This ACL might not be appropriate for controlling access to replication information.

Required authorities:

- Control replication - You must have write access to the **ibm-replicagroup=default** object (or be the owner/administrator).
- Cascading control replication - You must have write access to the **ibm-replicagroup=default** object (or be the owner/administrator).
- Control queue - You must have write access to the replication agreement.

To view ACL properties using the Web Administration Tool utility and to work with ACLs:

1. Select the subtree you want to edit the ACLs on.
2. Expand the **Select Action** menu, select **Edit ACLs** and click **Go**.

See “Working with ACLs” on page 318 for information on how to edit ACLs and see Chapter 17, “Access control lists,” on page 309 for additional information about ACLs.

Working with credentials

You can use the Web Administration Tool to perform the following tasks:

Adding credentials

Expand the **Replication management** category in the navigation area of the Web Administration Tool and click **Manage credentials**

1. Select the location that you want to use to store the credentials from the list of subtrees. The Web Administration Tool allows you to define credentials in three locations.
 - **cn=replication,cn=localhost**, which keeps the credentials only on the current server.

Note: In most replication cases, locating credentials in **cn=replication,cn=localhost** is preferred because it provides greater security than replicated credentials located on the subtree. However, there are certain situations in which credentials located on **cn=replication,cn=localhost** are not available.

If you are trying to add a replica under a server, for example serverA and you are connected to a different server with the Web Administration Tool, serverB, the **Select credentials** field does not display the option **cn=replication,cn=localhost**. This is because you cannot read the information or update any information under **cn=localhost** of the serverA when you are connected to serverB.

The **cn=replication,cn=localhost** is only available when the server under which you are trying to add a replica is the same server that you are connected to with the Web Administration Tool.

- **cn=replication,cn=IBMpolicies**, which is available even when the server under which you are trying to add a replica is not the same server that you are connected to with the Web Administration Tool. Credentials placed under this location are replicate to the servers.

Note: The location **cn=replication,cn=IBMpolicies** is only available, if the **IBMpolicies** support OID, 1.3.18.0.2.32.18, is present under the **ibm-supportedcapabilities** of the root DSE.

- Within the replicated subtree, in which case the credentials are replicated with the rest of the subtree. Credentials placed in the replicated subtree are created beneath the **ibm-replicagroup=default** entry for that subtree.

Note: If no subtrees are displayed, go to “Adding a subtree” on page 222 for instructions about creating the subtree that you want to replicate.

2. Click **Add**.

3. Enter the name for the credentials you are creating, for example, **mycreds**, **cn=** is prefilled in the field for you.

4. Select the type of authentication method you want to use and click **Next**.

- If you selected simple bind authentication:
 - a. Enter the DN that the server uses to bind to the replica, for example, **cn=any**
 - b. Enter the password uses when it binds to the replica, for example, **secret**.
 - c. Enter the password again to confirm that there are no typographical errors.
 - d. If you want, enter a brief description of the credentials.
 - e. Click **Finish**.

Note: You might want to record the credential’s bind DN and password for future reference. You will need this password when you create the replica agreement.

- If you selected Kerberos authentication:
 - a. Enter your Kerberos bind DN.
 - b. Enter a keyfile (the fully-qualified file specification of the key database file). Leave this field blank to use the server’s LDAP service name.

Note: The server’s LDAP service principal name is *service/hostname@realm*. This comes from standard Kerberos conventions. The *service* is always **ldap**. For example, for host **myserver.mytown.mycompany.com** in Kerberos realm **"MYTOWN.MYCOMPANY.COM"**, the server’s principal name is:
`ldap/myserver.mytown.mycompany.com@MYTOWN.MYCOMPANY.COM`

The server gets the host name from the system TCP/IP configuration; the realm name comes from the realm name configured on the **Kerberos** tab on the **Security properties** panel.

- c. If you want, enter a brief description of the credentials. No other information is necessary. See “Setting Kerberos” on page 144 for additional information.
- d. Click **Finish**.

The Kerberos panel takes an optional bind DN of the form `ibm-kn=xxx@realm` and an optional key tab file name (referred to as keyfile on the Web Administration Tool). If a bind DN is specified, the server uses the specified principal name to authenticate to the consumer server. Otherwise, the server’s Kerberos service name (`ldap/host-name@realm`) is used.

If a key tab file is used, the server uses the key tab file to obtain the credentials for the specified principal name. If no key tab file is specified, the server uses the key tab file specified in the server’s Kerberos configuration.

By default, the supplier uses its own service principal to bind with the consumer. For example, if the supplier is named `master.our.org.com` and the realm is `SOME.REALM`, the DN is **ibm-**

Kn=ldap/master.our.org.com@SOME.REALM. The realm value is case insensitive.

Note: If more than one supplier uses Kerberos authentication to replicate to the same consumer, you must configure all suppliers to use the same Kerberos principal rather than letting them default to using their Kerberos service name.

- If you selected SSL with certificate authentication you do not need to provide any additional information, if you are using the server’s certificate. If you choose to use a certificate other than the server’s:
 - a. Enter the key file name.
 - b. Enter the key file password.
 - c. Reenter the key file password to confirm it.
 - d. Enter the key label.
 - e. If you want, enter a brief description.
 - f. Click **Finish**.

See “Secure Sockets Layer” on page 115 for additional information.

Note: If an external credential object is selected while you are adding credentials on consumers during an Add master operation using the Web Administration Tool, then the following settings need to be configured on the machine where the Web server is running:

- The `JAVA_HOME\jre\lib\ext\` has the following jar files:
 - `ibmjceprovider.jar`
 - `ibmpkcs.jar`
 - `ibmjcefw.jar`
 - `local_policy.jar`
 - `US_export_policy.jar`
 - `ibmjlog.jar`
 - `gsk7cls.jar`
- The `JAVA_HOME\jre\lib\security\java.security` file must have the following two lines to register CMS provider and JCE provider:


```
security.provider.2=com.ibm.spi.IBMCMSProvider
security.provider.3=com.ibm.crypto.provider.IBMJCE
```

- Gskit must be installed and gsk7\lib must be in the system path.
- For the Web Administration Tool to read the keyfile containing credentials information that the master server uses to connect to the replica, and create credentials on replica, the keyfile must be present in C:\temp for Windows platforms, and in /tmp for UNIX.

Modifying credentials

Expand the Replication management category in the navigation area of the Web Administration Tool and click **Manage credentials**.

1. Select a subtree and click **Show credentials**.
2. In the credentials box for the selected subtree, select the credentials you want to modify and click **Edit**.
 - If the credential is simple authentication. In the Edit credential panel you can modify:
 - The **Bind DN**
 - The **Password**
 - The **Description** of the credential
 - If the credential is kerberos authentication. In the Edit credential panel you can modify:
 - The **Bind DN**
 - The **Key file**
 - The **Description** of the credential
 - If the credential is SSL with certificate authentication. In the Edit credential panel you can modify:
 - The **Key file**
 - The **Password**
 - The **Key label**
 - The **Description** of the credential
3. When you are finished, click **OK**.

Removing credentials

Expand the Replication management category in the navigation area of the Web Administration Tool and click **Manage credentials**

1. Select a subtree and click **Show credentials**.
2. In the credentials box for the selected subtree, select the credentials you want to remove and click **Delete**.
3. A message confirming that you want to delete the credential object is displayed. Click **OK** to remove the credential or click **Cancel** to return to the **Manage credentials** panel without saving any changes.

Managing credential ACLs

Use this information if you want to enable others to work with credentials. You need to assign ACLs to enable this function.

Expand the Replication management category in the navigation area of the Web Administration Tool and click **Manage credentials**

1. Select a subtree and click **Show credentials**.
2. In the credentials box for the selected subtree, select the credentials you want to modify the ACLs for and click **Edit ACL**.

3. See “Working with ACLs” on page 318 for information on editing ACLs.

Managing topologies

Topologies are specific to the replicated subtrees.

Viewing the topology

Note: The server must be running to perform this task.

Expand the **Replication management** category in the navigation area and click **Manage topology**.

1. Select the subtree that you want to view and click **Show topology**.

The topology is displayed in the Replication topology list. Expand the topologies. From this list you can:

- Add a master
- Add a replica
- Manage gateway servers
- Edit an agreement
- View the replication schedule
- View replication errors
- Move a server to a different role in the topology
- Delete a server.

Adding a peer-master or gateway server

Note: The server must be running to perform this task.

Expand the **Replication management** category in the navigation area and click **Manage topology**.

1. Select the subtree that you want to replicate and click **Show topology**.
2. Click the box next to the **Replication Topology** to expand the list of supplier servers, if you want to view the existing topology.
3. Click **Add master**.

On the **Server** tab of the **Add master** window:

- Enter the host name and port number for the server you are creating. The default port is 389 for non-SSL and 636 for SSL. These are required fields.
- Select whether to enable SSL communications.
- Select whether you want to create the server as a gateway server.
- Enter the server name or leave this field blank to use the host name.
- Enter the server ID. If the server on which you are creating the peer-master is running, click **Get server ID** to automatically prefill this field.
- Enter a description of the server.
- You must specify the credentials that the server uses to communicate with the other master server. Click **Select**.

Note: The Web Administration Tool allows you to define credentials in the following places:

- **cn=replication,cn=localhost**, which keeps the credentials only on the server that uses them. Placing credentials in **cn=replication,cn=localhost** is considered more secure.
- **cn=replication,cn=IBMpolicies**, which is available even when the server under which you are trying to add a replica is not the same server that you are connected to with the Web Administration Tool. Credentials placed under this location are replicate to the servers.

Note: The location **cn=replication,cn=IBMpolicies** is only available, if the **IBMpolicies** support OID, 1.3.18.0.2.32.18, is present under the **ibm-supportedcapabilities** of the root DSE.

- Within the replicated subtree, in which case the credentials are replicated with the rest of the subtree. Credentials placed in the replicated subtree are created beneath the **ibm-replicagroup=default** entry for that subtree.
1. Select the location for the credentials you want to use. Preferably this is **cn=replication,cn=localhost**.
 2. If you have already created a set of credentials, click **Show credentials**.
 3. Expand the list of credentials and select the one you want to use.
 4. Click **OK**.
 5. If you do not have preexisting credentials, click **Add** to create the credentials. See “Adding credentials” on page 224 for additional information on agreement credentials.

On the **Additional** tab:

1. Specify a replication schedule from the drop-down list or click **Add** to create one. See “Creating replication schedules” on page 238
2. From the list of supplier capabilities, you can deselect any capabilities that you do not want replicated to the consumer.

If your network has a mix of servers at different releases, capabilities are available on later releases that are not available on earlier releases. Some capabilities, like filter ACLs (“Filtered ACLs” on page 310) and password policy (“Setting password policy” on page 134), make use of operational attributes that are replicated with other changes. In most cases, if these features are used, you want all servers to support them. If all of the servers do not support the capability, you do not want to use it. For example, you would not want different ACLs in effect on each server. However, there might be cases where you might want to use a capability on the servers that support it, and not have changes related to the capability replicated to servers that do not support the capability. In such cases, you can use the capabilities list to mark certain capabilities to not be replicated.

3. Check the **Add credential information on consumer** check box, if you want to enable dynamic updates of the supplier credentials. This selection automatically updates the supplier information in the configuration file of the server you are creating. This enables the topology information to be replicated to the server.
 - Type the Administration DN for this, the consumer, server. For example **cn=root**.

Note: If the administrator DN which was created during the server configuration process was **cn=root**, then enter the full administrator DN. Do not just use **root**.

- Type the Administration password for this, the consumer, server. For example secret.
4. Click **OK**.
 5. Supplier and consumer agreements are listed between new master server and any existing servers. Uncheck any agreements that you do not want to be created. This is especially important if you are creating a gateway server.
 6. Click **Continue**.
 7. Messages might be displayed noting that additional actions must be taken. Perform or take note of the appropriate actions. When you are finished, click **OK**.
 8. Add the appropriate credentials.

Note: In some cases the Select credentials panel will pop up asking for a credential that is located in a place other than `cn=replication,cn=localhost`. In such situations you must provide a credential object that is located in a place other than `cn=replication,cn=localhost`. Select the credentials the subtree is going to use from the existing sets of credentials or create new credentials. See “Adding credentials” on page 224.

9. Check the **Add credential information on consumer** check box, if you want to enable dynamic updates of the supplier credentials. This selection automatically updates the supplier information in the configuration file of the server you are creating. This enables the topology information to be replicated to the server.

- Type the Administration DN for this, the consumer, server. For example `cn=root`.

Note: If the administrator DN which was created during the server configuration process was `cn=root`, then enter the full administrator DN. Do not just use `root`.

- Type the Administration password for this, the consumer, server. For example secret.
10. Click **OK** to create the peer-master.
 11. Messages might be displayed noting that additional actions must be taken. Perform or take note of the appropriate actions. When you are finished, click **OK**. See “Starting replication” on page 177.

Note: If an external credential object is selected while you are adding credentials on consumers during an Add master operation using the Web Administration Tool, then the following settings need to be configured on the machine where the IBM WebSphere Application Server is running:

- The `WAS_HOME\java\jre\lib\ext\` has the following jar files:
 - `ibmjceprovider.jar`
 - `ibmpkcs.jar`
 - `ibmjcefw.jar`
 - `local_policy.jar`
 - `US_export_policy.jar`
 - `ibmjlog.jar`
 - `gsk7cls.jar`
- The `WAS_HOME\java\jre\lib\security\java.security` file must have the following two lines to register CMS provider and JCE provider:

```
security.provider.2=com.ibm.spi.IBMCMSProvider
security.provider.3=com.ibm.crypto.provider.IBMJCE
```

- Restart IBM WebSphere Application Server.
- Gskit must be installed and gsk7\lib must be in the system path.
- For the Web Administration Tool to read the keyfile containing credentials information that the master server uses to connect to the replica, and create credentials on replica, the keyfile must be present in C:\temp for Windows platforms, and in /tmp for UNIX.

Adding a replica server

Note: The server must be running to perform this task.

Expand the **Replication management** category in the navigation area and click **Manage topology**.

1. Select the subtree that you want to replicate and click **Show topology**.
2. Click the box next to the existing servers to expand the list of supplier servers.
3. Select the supplier server and click **Add replica**.

On the **Server** tab of the **Add replica** window:

- Enter the host name and port number for the replica you are creating. The default port is 389 for non-SSL and 636 for SSL. These are required fields.
- Select whether to enable SSL communications.
- Enter the replica name or leave this field blank to use the host name.
- Enter the replica ID. If the server on which you are creating the replica is running, click **Get replica ID** to automatically prefill this field.
- Enter a description of the replica server.
- You must specify the credentials that the replica uses to communicate with the master. Click **Select**.

Note: The Web Administration Tool allows you to define credentials in the following places:

- **cn=replication,cn=localhost**, which keeps the credentials only on the server that uses them. Placing credentials in **cn=replication,cn=localhost** is considered more secure.
- **cn=replication,cn=IBMpolicies**, which is available even when the server under which you are trying to add a replica is not the same server that you are connected to with the Web Administration Tool. Credentials placed under this location are replicate to the servers.

Note: The location **cn=replication,cn=IBMpolicies** is only available, if the **IBMpolicies** support OID, 1.3.18.0.2.32.18, is present under the **ibm-supportedcapabilities** of the root DSE.

- Within the replicated subtree, in which case the credentials are replicated with the rest of the subtree. Credentials placed in the replicated subtree are created beneath the **ibm-replicagroup=default** entry for that subtree.
1. Select the location for the credentials you want to use. Preferably this is **cn=replication,cn=localhost**.
 2. If you have already created a set of credentials, click **Show credentials**.
 3. Expand the list of credentials and select the one you want to use.
 4. Click **OK**.

5. If you do not have preexisting credentials, click **Add** to create the credentials. See “Adding credentials” on page 224 for additional information on agreement credentials.

On the **Additional** tab:

1. Specify a replication schedule from the drop-down list or click **Add** to create one. See “Creating replication schedules” on page 238
2. From the list of supplier capabilities, you can deselect any capabilities that you do not want replicated to the consumer.

If your network has a mix of servers at different releases, capabilities are available on later releases that are not available on earlier releases. Some capabilities, like filter ACLs (“Filtered ACLs” on page 310) and password policy (“Setting password policy” on page 134), make use of operational attributes that are replicated with other changes. In most cases, if these features are used, you want all servers to support them. If all of the servers do not support the capability, you do not want to use it. For example, you would not want different ACLs in effect on each server. However, there might be cases where you might want to use a capability on the servers that support it, and not have changes related to the capability replicated to servers that do not support the capability. In such cases, you can use the capabilities list to mark certain capabilities to not be replicated.
3. Select the either **Single threaded** or **Multi-threaded** for the method of replication. If you specify **Multi-threaded**, you must also specify the number (between 2 and 32) of connections to use for replication. The default number of connections is 2.
4. Check the **Add credential information on consumer** check box, if you want to enable dynamic updates of the supplier credentials. This selection automatically updates the supplier information in the configuration file of the server you are creating. This enables the topology information to be replicated to the server.
 - Type the Administration DN for this, the consumer, server. For example `cn=root`.

Note: If the administrator DN which was created during the server configuration process was `cn=root`, then enter the full administrator DN. Do not just use `root`.

 - Type the Administration password for this, the consumer, server. For example `secret`.
5. Click **OK** to create the replica.
6. A message is displayed noting that additional actions must be taken. Click **OK**.

Notes:

1. If you are adding more servers as additional replicas or are creating a complex topology, do not proceed with “Copying data to the replica” on page 176 or “Adding the supplier information to a replica” on page 236 until you have finished defining the topology on the master server. If you create the *masterfile.ldif* after you have completed the topology, it contains the directory entries of the master server and a complete copy of the topology agreements. When you load this file on each of the servers, each server then has the same information.
2. If an external credential object is selected while you are adding credentials on consumers during an Add replica operation using the Web Administration Tool, see 230.

Removing a server

Expand the **Replication management** category in the navigation area and click **Manage topology**.

1. Select the subtree that you want and click **Show topology**.
2. Select the server that you want to remove from the topology.
3. Click **Delete**.
4. When asked to confirm the deletion, click **OK**.

Note: When removing a replica from your topology, remember to delete the supplier credential entry from the consumer if no master server will be using this credential entry again. A master server should not have any agreements under it. See “Removing credentials” on page 227.

Moving or promoting a server

Expand the **Replication management** category in the navigation area and click **Manage topology**.

1. Select the subtree that you want and click **Show topology**.
2. Select the server that you want and click **Move**.
3. Select the server that you want to move the replica to, or select **Replication topology** to promote the replica to a master. Click **Move**.
4. The **Create additional supplier agreements** is displayed. Deselect the supplier agreements that are not appropriate for the role of the server. You are prompted to select the credentials and consumer information for each new supplier credential being created. Existing supplier agreements from the other servers to the newly promoted server are still in effect and do not need to be recreated.

Note: In some cases the Select credentials panel will pop up asking for a credential which is located in a place other than `cn=replication,cn=localhost`. In such situations you must provide a credential object which is located in a place other than `cn=replication,cn=localhost`. Select the credentials the subtree is going to use from the existing sets of credentials or create new credentials. The credential entry should exist or be created on the other masters. See “Adding credentials” on page 224.

5. Click **OK**.

The change in the topology tree reflects the moving of the server.

See “Setting up a complex topology with peer replication” on page 192 for more information.

Demoting a master

To change the role of a server from a master to a replica, expand the **Replication management** category in the navigation area and click **Manage topology**.

1. Connect the Web Administration Tool to the server that you want to demote.
2. Click **Manage topology**.
3. Select the subtree and click **Show topology**.
4. Select the server you are demoting and click **Move**.
5. Select the server under which you are going to place the demoted server and click **Move**.
6. Delete all the agreements for the server you want to demote. Click **Yes**.

Managing gateway servers

Note: A gateway server must be an IBM Tivoli Directory Server version 6.0 or version 5.2 server or an IBM Directory Server version 5.1 server with a fix pack that supports gateway replication.

You can designate whether a master server is to have the role of a gateway server in the replication site.

To designate a master to be a gateway server, expand the **Replication management** category in the navigation area and click **Manage topology**.

1. Select the subtree that you want to view and click **Show topology**.
2. Click **Manage gateway servers**.
3. Select the server from the **Master servers** box that you want to make a gateway server.
4. Click **Make gateway**. The server is moved from the **Master servers** box to the **Gateway servers** box
5. Click **OK**.

To remove the role of a gateway server from a master server.

1. Click **Manage gateway servers**.
2. Select the server from the **Gateway servers** box that you want to make a master server.
3. Click **Make master**. The server is moved from the **Gateway servers** box to the **Master servers** box
4. Click **OK**.

Note: Remember that there can be only one gateway server per replication site. When you create additional gateway servers in your topology, the Web Administration Tool treats the gateway as a peer server and creates agreements to all the servers in the topology. Ensure that you deselect any agreements that are not with the other gateway servers or not within the gateways own replication site.

See “Setting up a gateway topology” on page 207 for more information.

Editing an agreement

You can change the following information for the replica:

On the **Server** tab you can only change

- Hostname and port

Note: The port is editable only for switching from non-SSL-enabled to SSL-enabled, and back.

- Enable SSL
- Description
- Credentials - see “Adding credentials” on page 224.

On the **Additional** tab you can change:

- Replication schedules - see “Creating replication schedules” on page 238.

- Change the capabilities replicated to the consumer replica. From the list of supplier capabilities, you can deselect any capabilities that you do not want replicated to the consumer.
- Replication method.
- Consumer information.
- When you are finished, click **OK**.

Viewing the replication schedule

Expand the **Replication management** category in the navigation area of the Web Administration Tool and click **Manage topology**

1. Select the subtree that you want to view and click **Show topology**.
2. Select the master or gateway server that you want to view.
3. Click **View schedule**.

If a replication schedule exists between the selected server and its consumers, they are displayed. You can modify or delete these schedules. If no schedules exist and you want to create one, you must use the **Manage schedules** function from the Web Administration Tool navigation area. See “Creating replication schedules” on page 238 for information about managing schedules.

Viewing server information

Expand the Replication management category in the navigation area of the Web Administration Tool and click **Manage topology**

1. Select the subtree that you want to view and click **Show topology**.
2. Select the server that you want to view.
3. Click **View server**.

The server name, ID and role and consumer information are displayed.

Viewing server errors

You can view replication updates that were not completed because of errors that occurred during replication.

Expand the Replication management category in the navigation area of the Web Administration Tool and click **Manage topology**

1. Select the subtree that you want to view and click **Show topology**.
2. Select the server (replica agreement) that you want to view.
3. Click **View errors**.

The subtree, supplier and consumer information is displayed. Replication errors are displayed in a table that supplies the following information:

Change ID

The ID assigned to the failed update.

Last update time

Indicates the time when the last attempt to replicate the entry was made.

Number of attempts

Indicates the number of attempts made to replicate the entry.

Result code

Result code obtained by the last attempt to replicate the entry.

Note: The order this information is displayed in is defined by failure ID. Failure IDs are assigned as they happen. The failure ID is not the same as the change ID. The change ID remains constant, but the failure ID is changed on every failed attempt.

You can select an error and perform the following actions:

- Click **Show details** to view more information about the error.
- Click **Retry** to attempt the update again.
- Click **Remove** to remove the error from the Replication error management table.

You can also

- Click **Retry all** to attempt all the update again.
- Click **Remove all** to remove all the errors from the Replication error management table.

See “Managing queues” on page 240 for additional information.

Adding the supplier information to a replica

If you did not select to add the credential information to the consumer or if a problem occurred in adding the credential information to the replica, you need to change the replica’s configuration to identify who is authorized to replicate changes to it, and add a referral to a master.

On the machine where you are creating the replica:

1. Expand **Replication management** in the navigation area and click **Manage replication properties**.
2. Click **Add**.
3. Select a supplier from the **Replicated subtree** drop-down menu or enter the name of the replicated subtree for which you want to configure supplier credentials. If you are editing supplier credentials, this field is not editable.
4. Enter the replication bindDN. In this example, cn=any.

Note: You can use either of these two options, depending on your situation.

- Set the replication bind DN (and password) and a default referral for all subtrees replicated to a server using the ‘default credentials and referral’. This might be used when all subtrees are replicated from the same supplier.
 - Set the replication bind DN and password independently for each replicated subtree by adding supplier information for each subtree. This might be used when each subtree has a different supplier (that is, a different master server for each subtree).
5. Depending on the type of credential, enter and confirm the credential password. (You previously recorded this for future use.)
 - **Simple Bind** - Specify the DN and password
 - **Kerberos** - If the credentials on the supplier do not identify the principal and password, that is, the server’s own service principal is to be used, then the bind DN is `ibm-kn=ldap/<yourservername@yourrealm>`. If the credentials has a principal name such as `<myprincipal@myrealm>`, use that as the DN. In either case a password is not needed.
 - **SSL w/ EXTERNAL bind** - Specify the subject DN for the certificate and no password

See “Adding credentials” on page 224.

6. Click **OK**.

7. You must restart the replica for the changes to take effect.

See “Modifying replication properties” for additional information.

The replica is in a suspended state and no replication is occurring. After you have finished setting up your replication topology, you must click **Manage queues**, select the replica and click **Suspend/resume** to start replication. See “Managing queues” on page 240 for more detailed information. The replica now receives updates from the master.

Modifying replication properties

Expand the **Replication management** category in the navigation area and click **Manage replication properties**.

On this panel you can:

- Change the maximum number of pending changes to return from replication status queries. The default is 200.
- Set the maximum number of replication errors a server will log while replicating updates to a consumer. If the server is using single-threaded replication, and the maximum is exceeded, the update is retried periodically until it succeeds or until the administrator clears the log so the failure can be added. If the server is using multi-threaded replication, and the maximum is exceeded, any replication errors that occur for the updates in progress are logged and replication waits for the administrator to clear the log. The log can be cleared by retrying or removing the failed updates. Separate logs are maintained for each consumer. The default is zero as in none.

Note: Logging is enabled if a value greater than zero is specified.

- Change the size in bytes of the replication context cache. The default is 100 000 bytes.
- Set the replication conflict maximum entry size in bytes . If the total size of an entry in bytes exceeds the value in this field, the entry is not sent again by the supplier to resolve a replication conflict on the consumer. The default is 0 for unlimited.
- Add, edit, or delete supplier information.

Adding supplier information

1. Click **Add**.
2. Select a supplier from the drop-down menu or enter the name of the replicated subtree that you want to add as a supplier .
3. Enter the replication bind DN for the credentials.

Note: You can use either of these two options, depending on your situation.

- Set the replication bind DN (and password) and a default referral for all subtrees replicated to a server using the ‘default credentials and referral’. This might be used when all subtrees are replicated from the same supplier.
 - Set the replication bind DN and password independently for each replicated subtree by adding supplier information for each subtree. This might be used when each subtree has a different supplier (that is, a different master server for each subtree).
4. Depending on the type of credential, enter and confirm the credential password. (You previously recorded this for future use.)

- **Simple Bind** - specify the DN and password
- **Kerberos** - specify a pseudo DN of the form 'ibm-kn=LDAP-service-name@realm' without a password
- **SSL w/ EXTERNAL bind** - specify the subject DN for the certificate and no password

See "Adding credentials" on page 224.

5. Click **OK**.

The subtree of the supplier is added to the Supplier information list.

Editing supplier information

1. Select the supplier subtree that you want to edit.
2. Click **Edit**.
3. If you are editing **Default credentials and referral**, which is used to create the cn=Master Server entry under cn=configuration, enter the URL of the server from which the client wants to receive replica updates in the Default supplier's LDAP URL field. This needs to be a valid LDAP URL (ldap://). Otherwise, skip to step 4.
4. Enter the replication bind DN for the new credentials you want to use.

Note: Set the replication bind DN and password independently for each replicated subtree by adding supplier information for each subtree. This might be used when each subtree has a different supplier (that is, a different master server for each subtree).

5. Enter and confirm the credential password.
6. Click **OK**.

Removing supplier information

1. Select the supplier subtree that you want to remove.
2. Click **Delete**.
3. When asked to confirm the deletion, click **OK**.

The subtree is removed from the Supplier information list.

Creating replication schedules

You can optionally define replication schedules to schedule replication for particular times, or to not replicate during certain times. If you do not use a schedule, the server schedules replication whenever a change is made. This is equivalent to specifying a schedule with immediate replication starting at 12:00 AM on all days.

Expand the **Replication management** category in the navigation area and click **Manage schedules**.

On the **Weekly schedule** tab, select the subtree for which you want to create the schedule and click **Show schedules**. If any schedules exist, they are displayed in the **Weekly schedules** box. To create or add a new schedule:

1. Click **Add**.
2. Enter a name for the schedule. For example **schedule1**.
3. For each day, Sunday through Saturday, the daily schedule is specified as **None**. This means that no replication update events are scheduled. The last

replication event, if any, is still in effect. Because this is a new replica, there are no prior replication events, therefore, the schedule defaults to immediate replication.

4. You can select a day and click **Add a daily schedule** to create a daily replication schedule for it. If you create a daily schedule it becomes the default schedule for each day of the week. You can:
 - Keep the daily schedule as the default for each day or select a specific day and change the schedule back to none. Remember that the last replication event that occurred is still in effect for a day that has no replication events scheduled.
 - Modify the daily schedule by selecting a day and clicking **Edit a daily schedule**. Remember changes to a daily schedule affect all days using that schedule, not just the day you selected.
 - Create a different daily schedule by selecting a day and clicking **Add a daily schedule**. After you have created this schedule it is added to the **Daily schedule** drop-down menu. You must select this schedule for each day that you want the schedule to be used.

See “Creating a daily schedule” for more information on setting up daily schedules.

5. When you are finished, click **OK**.

Creating a daily schedule

Expand the **Replication management** category in the navigation area and click **Manage schedules**.

On the **Daily schedule** tab, select the subtree for which you want to create the schedule and click **Show schedules**. If any schedules exist, they are displayed in the **Daily schedules** box. To create or add a new schedule:

1. Click **Add**.
2. Enter a name for the schedule. For example **monday1**.
3. Select the time zone setting, either UTC or local.
4. Select a replication type from the drop-down menu:

Immediate

Performs any pending entry updates since the last replication event and then updates entries continuously until the next scheduled update event is reached.

Once

Performs all pending updates prior to the starting time. Any updates made after the start time wait until the next scheduled replication event.

5. Select a start time for the replication event.
6. Click **Add**. The replication event type and time are displayed.
7. Add or remove events to complete your schedule. The list of events is refreshed in chronological order.
8. When you are finished, click **OK**.

For example:

Replication type	Start time
Immediate	12:00 AM
Once	10:00 AM
Once	2:00 PM

Replication type	Start time
Immediate	4:00 PM
Once	8:00 PM

In this schedule, the first replication event occurs at midnight and updates any pending changes prior to that time. Replication updates continue to be made as they occur until 10:00 AM. Updates made between 10:00 AM and 2:00 PM wait until 2:00 PM to be replicated. Any updates made between 2:00 PM and 4:00 PM wait the replication event scheduled at 4:00 PM, afterwards replication updates continue until the next scheduled replication event at 8:00 PM. Any updates made after 8:00 PM wait until the next scheduled replication event.

Note: If replication events are scheduled too closely together, a replication event might be missed if the updates from the previous event are still in progress when the next event is scheduled.

Managing queues

This task allows you to monitor status of replication for each replication agreement (queue) used by this server.

Expand the **Replication management** category in the navigation area and click **Manage queues**.

The Manage queues table contains the following information in columns:

Select Selects the replica on which you want to perform an action.

Replica

Specifies the name of the replica in the replication queue.

Subtree

Specifies the subtree under which the replica is located.

Last result

Indicates the last return code/status (success/failed)

State Indicates the state of replication with the consumer:

Active Actively sending updates to consumer.

Ready In immediate replication mode, ready to send updates as they occur.

Waiting

Waiting for next scheduled replication time.

Binding

In the process of binding to the consumer.

Connecting

In the process of connecting to the consumer.

On Hold

This replication agreement has been suspended or "held".

Error Log Full

For a server configured to use multiple connections, replication is suspended for this agreement. The receiver threads continue polling for status from any updates that have been sent, but no more updates are replicated.

Retrying

If the server is configured to use a single connection, replication attempts to send the same update after waiting for 60 seconds and keeps trying until replication succeeds or the administrator skips the update.

Queue size

Specifies the number of pending changes returned from replication status queries.

- Select the replica for which you want to manage the queue.
- Depending on the status of the replica, you can click **Suspend/resume** to stop or start replication.
- Click **Force replication** to replicate all the pending changes regardless of when the next replication is scheduled.
- Click **Queue details**, for more complete information about the replica's queue. You can also manage the queue from this selection.
- Click **Refresh** to update the queues, obtain the current status, and clear server messages.

Queue details

If you clicked **Queue details**, three tabs are displayed:

- Status
- Last attempted details
- Pending changes

The **Status** tab displays the replica name, its subtree, its replication status, and a record of replication times. From this panel you can suspend or resume replication by clicking **Suspend** or **Resume**. The non-editable status field changes to reflect the change in status. Click **Refresh** to update the queue information.

The **Last attempted details** tab gives the following information about the last update attempt on the selected replica:

- **Replica** - The name of the replica in the replication queue.
- **Subtree** - The subtree under which the replica is located.
- **Entry DN** - The DN of the updated entry.
- **Last replicated at** - The last time the entry was replicated.
- **Update type** - The type of update, for example, add, delete or modify.
- **Last result** - The error code assigned to the error.
- **Failed LDIF** - The update in LDIF format.
- **Additional error messages** - Any additional information about the error.

If an entry is not able to be loaded press **Skip blocking entry** to continue replication with the next pending entry. Click **Refresh** to update the queue information.

The **Pending changes** tab shows all the pending changes to the replica. The number of pending changes displayed depends on the value you entered on the **Manage replication properties** panel. The default is 200.

If replication is blocked you can delete all the pending changes by clicking **Skip all**. Click **Refresh** to update the list of pending changes to reflect any new update or updates that have been processed.

Note: If you choose to skip blocking changes, you must ensure that the consumer server is eventually updated. See “idsldapdiff, ldapdiff” on page 377 for more information.

Command line tasks for managing replication

Specifying a supplier DN and password for a subtree

You can specify a supplier DN and PW for a particular subtree. To do this the following information is needed on all consumers:

1. Start the consumer servers.
2. You must configure replica1 to be a replica server. Do the following to add an entry to the **ibmslapd.conf** file on replica1:

```
idsldapmodify -D <admin_dn> -w <admin_pw> -I <instance_name> -i <LDIF_file>
```

where *<LDIF_file>* contains the following:

```
dn: cn=Master Server, cn=configuration
cn: Master Server
ibm-slapdMasterDN: cn=master
ibm-slapdMasterPW: <masterserverpassword>
ibm-slapdMasterReferral: ldap://<masterhostname:masterport>
objectclass: ibm-slapdReplication
```

```
dn: cn=Supplier s1, cn=configuration
cn: Supplier s1
ibm-slapdMasterDN: cn=s1
ibm-slapdMasterPW: s1
ibm-slapdReplicaSubtree: ou=Test, o=IBM, c=US
objectclass: ibm-slapdSupplier
```

3. Save the **ibmslapd.conf** file.
4. Restart replica1.

Viewing replication configuration information

A great deal of information related to replication activity is available using searches. To see the replication topology information related to a particular replicated subtree, you can do a subtree search with the base set to the DN of the subtree and the filter set as (**objectclass=ibm-repl***) to find the subentry that is the base of the topology information. If this replication context was created through the web admin interface, the name of the entry will be **ibm-replicaGroup=default**.

```
idsldapsearch -D <adminDN> -w <adminPW> -p <port> -b <suffixentryDN>
objectclass=ibm-repl*
```

The objects returned will include the replica group itself, plus the following:

- An object with **objectclass=ibm-replicaSubentry** for each server that replicates data within this context. Replica subentries contain a server ID attribute and an indication of the role the server plays (**ibm-replicationServerIsMaster**).
- For each replica subentry, there is a replication agreement object for each consumer server that receives replication updates from the server described by the replica subentry. Each replication agreement contains the following information:
 - **ibm-replicaConsumerId**: The server ID of the consumer server.
 - **ibm-replicaURL**: The LDAP URL of the consumer server.

- **ibm-replicaCredentialsDN**: The DN of the entry containing the credentials used to bind to the consumer.

Agreements may also contain the following:

- **ibm-replicaScheduleDN**: The DN of a schedule entry that determines when replication updates are sent to this consumer. If no schedule is specified, replication defaults to "immediate" mode.
- **ibm-replicationOnHold**: A boolean indicating that replication to this consumer is suspended (or not).
- **ibm-replicationExcludedCapability**: The values of this attribute list OIDs of features that the consumer does not support. Operations related to these capabilities are then excluded from the updates sent to this consumer.
- **ibm-replicationMethod**: Single threaded or multi-threaded.
- **ibm-replicationConsumerConnections**: For a replication agreement using the single-threaded replication method, the number of consumer connections is always one, the attribute value is ignored. For an agreement using multi-threaded replication, the number of connections can be configured from 1 to 32. If no value is specified on the agreement, the number of consumer connections is set to one.

Monitoring replication status

In addition, there are many operational attributes that provide replication status information when explicitly requested on a search. One of these attributes is associated with the entry that is the base of the replicated subtree, that is, the entry that the **ibm-replicationContext** objectclass was added to. If you do a base search of that entry, and request that the **ibm-replicationIsQuiesced** attribute is returned. This attribute is a boolean that indicates if the subtree has been quiesced. If the subtree is quiesced, no client updates are allowed (only updates from replication suppliers are accepted). There is an extended operation that can be used to quiesce a subtree, see "idsldapexop, ldapexop" on page 386.

The remainder of the status-related operational attributes are all associated with a replication agreement object. These attributes are only returned when explicitly requested on the search. The attributes available are:

- **ibm-replicationLastActivationTime**: The time that the last replication session started between this supplier and consumer.
- **ibm-replicationLastFinishTime**: The time that the last replication session finished between this supplier and consumer.
- **ibm-replicationLastChangeId**: The change ID of the last update sent to this consumer.
- **ibm-replicationState**: The current state of replication with this consumer. Possible values are:

Active Actively sending updates to consumer.

Ready In immediate replication mode, ready to send updates as they occur.

Waiting

Waiting for next scheduled replication time.

Binding

In the process of binding to the consumer.

Connecting

In the process of connecting to the consumer.

On Hold

This replication agreement has been suspended or "held".

Error Log Full

For a server configured to use multiple connections, replication is suspended for this agreement. The receiver threads continue polling for status from any updates that have been sent, but no more updates are replicated.

Retrying

If the server is configured to use a single connection, replication attempts to send the same update after waiting for 60 seconds and keeps trying until replication succeeds or the administrator skips the update.

- **ibm-replicationLastResult** The results of the last attempted update to this consumer, in the form:

<time stamp> <change id> <result code> <operation> <entry DN>

Note: This information is available for single threaded replication only.

- **ibm-replicationLastResultAdditional:** Any additional error information returned from the consumer for the last update.

Note: This information is available for single threaded replication only.

- **ibm-replicationPendingChangeCount:** The number of updates queued to be replicated to this consumer.
- **ibm-replicationPendingChanges:** Each value of this attribute gives information about one of the pending changes in the form:

<change id> <operation> <entry DN>

Requesting this attribute might return many values. Check the change count before requesting this attribute.

- **ibm-replicationChangeLDIF:** Gives the full details of the last failing update in LDIF.

Note: This information is available for single threaded replication only.

- **ibm-replicationFailedChanges:** Similar to **ibm-replicationPendingChanges** in that it lists the IDs, DNs, update types, result codes, timestamps, numbers of attempts for failures logged for a specified replication agreement. The number of failures displayed are less than or equal to **ibm-slapdMaxPendingChangesDisplayed**.
- **ibm-replicationFailedChangeCount:** Similar to **ibm-replicationPendingChangeCount** in that it returns a count of the failures logged for a specified replication agreement.
- **ibm-replicationPerformance:** Information about multi-threaded replication.

Note: Only the following are allowed to view **ibm-replicationPendingChanges**, **ibm-replicationPendingChangesCount**, **ibm-replicationFailedChanges** and **ibm-replicationChangeLDIF**:

- The administrator
- Members of the administrative group
- Members of the global administrative group
- Any user explicitly given update access to the replication topology entries through ACLs

Creating gateway servers

Creating a new Gateway server

Note: After creating a Gateway server, you must create new replication agreements to reflect the new topology. See the “Replication agreements” on page 169 for more information.

Create a new replica context, replica group and replica subentry in the DIT. The replica subentry must contain the `ibm-replicaSubentry` object class and `ibm-replicaGateway` auxiliary object class. The `ibm-replicaSubentry` object class and `ibm-replicaGateway` auxiliary object class are **bold** in the following example:

```
dn: o=ibm,c=us
objectclass: top
objectclass: organization
objectclass: ibm-replicationContext

dn: ibm-replicagroup=default,o=sandbox
objectclass: top
objectclass: ibm-replicaGroup
ibm-replicagroup: default

dn: ibm-replicaServerId=<serverid>,ibm-replicagroup=default,o=sandbox
objectclass: top
objectclass: ibm-replicaSubentry
objectclass: ibm-replicaGateway
ibm-replicaServerId:<serverid>
ibm-replicationServerIsMaster: TRUE
cn: <servername>
```

Where `<servername>` is the name of the server, and where `<serverid>` is a 37 character string assigned the first time a server is started. The server ID can be found by typing the following at a command prompt:

```
idsldapsearch -p <port> -b "" -s base objectclass=*
```

Converting an existing peer server to a Gateway server

Before converting a peer server to a Gateway server, make sure the subtree is quiesced and there are no pending changes. The following example shows a replica subentry that is NOT configured as a Gateway server.

```
dn: ibm-replicaServerId=<serverid>,ibm-replicagroup=default,o=sandbox
objectclass: top
objectclass: ibm-replicaSubentry
ibm-replicaServerId: <serverid>
ibm-replicationServerIsMaster: TRUE
cn: <servername>
```

To convert this peer to a gateway, add the `ibm-replicaGateway` auxiliary object class to the desired replica subentry in the DIT. The `ibm-replicaGateway` auxiliary object class is **bold** in the following example.

```
dn: ibm-replicaServerId=<serverid>,ibm-replicagroup=default,o=sandbox
changetype: modify
add: objectclass
objectclass: ibm-replicaGateway
```

Where `<servername>` is the name of the server, and where `<serverid>` is a 37 character string assigned the first time a server is started. The server ID can be found by typing the following at a command prompt:

```
idsldapsearch -p <port> -b "" -s base objectclass=*
```

For information about removing an auxiliary object class, see “Deleting an auxiliary object class” on page 304.

Chapter 14. Distributed directories

A distributed directory is a directory environment in which data is partitioned across multiple directory servers. A distributed directory must have a collection of machines including relational database management (RDBM) servers holding data, and proxy servers managing the topology.

The proxy server

Proxy server is a special type of IBM Tivoli Directory Server that provides request routing, load balancing, fail over, distributed authentication and support for distributed/membership groups and partitioning of containers. Most of these functions are provided in a new backend, the proxy backend. The proxy server does not have an RDBM backend and cannot take part in replication.

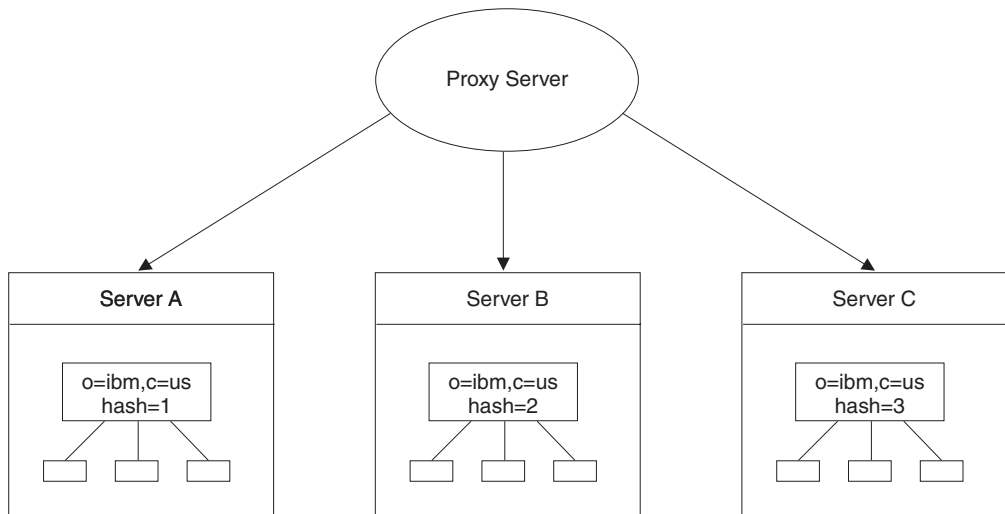
A directory proxy server sits at the front-end of a distributed directory and provides efficient routing of user requests thereby improving performance in certain situations, and providing a unified directory view to the client. It can also be used at the front-end of a server cluster for providing fail over and load balancing. The proxy server also provides data support for groups and ACLs which are not affected by partitioning, and support for partitioning of flat namespaces.

The proxy server is configured with connection information to connect to each of the backend servers for which it is proxying. The connection information comprises of host address, port number, bind DN, credentials and a connection pool size. Each of the back-end servers is configured with the DN and credentials that the proxy server uses to connect to it. The DN must be a member of the back-end server's (local) administration group or local administrator. Finally, the proxy server is configured with its own schema. You need to ensure that the proxy server is configured with the same schema as the back-end servers for which it is proxying. The proxy server must also be configured with partition information.

Note: If you use the `ddsetup` tool, the configuration information must be in sync with the `ddsetup` information. See “`ddsetup`” on page 420 for more information about `ddsetup`.

Splitting data within a subtree based on a hash of the RDN using a proxy server

In this setup, three servers have their data split within a “container” (under some entry in the directory tree). Because the proxy server handles the routing of requests to the appropriate servers, no referrals are used. Client applications need only be aware of the proxy server. The client applications never have to authenticate with servers A, B, or C.



Data is split evenly across the directories by hashing on the RDN just below the base of the split. In this example the data within the subtree is split based on the hash value of the RDN. Hashing is only supported on the RDN at one level in the tree under a container. Nested partitions are allowed. In the case of a compound RDN the entire normalized compound RDN is hashed. The hash algorithm assigns an index value to the DN of each entry. This value is then used to distribute the entries across the available servers evenly.

Notes:

1. The parent entries across multiple servers must remain synchronized. It is the administrator's responsibility to maintain the parent entries.
2. ACLs must be defined at the partition base level on each server.

Note: The number of partitions and the partition level are determined when the proxy server is configured, and when the data is split. There is no way to expand or reduce the topology without repartitioning.

The hash value enables the proxy server to locate and retrieve entries

For example: Data under `o=ibm,c=us` is split across three servers. This means that the proxy server is configured to hash RDN values immediately after `o=ibm,c=us` among 3 servers, or "buckets". This also means that RDN values more than 1 away from `o=ibm,c=us` will map to the same server as values immediately after `o=ibm,c=us`. For example, `cn=test,o=ibm,c=us` and `cn=user1,cn=test,o=ibm,c=us` will always map to the same server. Server A holds all the entries with a hash value of 1, server B holds all the entries with a hash value of 2, and server C holds all the entries with a hash value of 3. The proxy server receives an add request for an entry with DN `cn=Test,o=ibm,c=us`. The proxy server then uses the configuration information (specifically that there are 3 partitions with a base at `o=ibm,c=us`) and the `cn=Test` RDN as inputs to the internal hashing function. If the function returns 1, the entry resides on Server A and the add request is forwarded there.

Entry hashing is based on the RDN of the entry. Only the portion of the DN immediately to the left of the split point is used by the hash algorithm. Also, the whole string is used for the hash, not just the value. For example, if our split point is `o=ibm,c=us` and this is split into three partitions, then the following occurs:

- `cn=example,o=ibm,c=us` hashes to a single server, let's say serverA. This is determined by hashing `cn=example` into one of three partitions.

- `dc=example, o=ibm, c=us` hashes to a different server, let's say serverB. This is determined by hashing `dc=example`.
- `cn=foo, cn=example, o=ibm, c=us` hashes to serverA. This is because only `cn=example` is used for the hash algorithm. All entries beneath `cn=example, o=ibm, c=us` resolve to the same server as `cn=example, o=ibm, c=us`.

The distributed directory setup tool

The Distributed Directory Setup (`ddsetup`) tool splits an LDIF file into separate LDIF files that can be loaded onto individual directory servers. The `ddsetup` tool can be used in a non-distributed environment to merely split up an LDIF file into separate pieces. The user has the option of splitting the DIT at one or more subtrees, specifying the split points by DN.

The distributed directory setup (`ddsetup`) tool provides an option to split an LDIF file into files that can be loaded onto individual partitioned servers. In this release, the only supported way to split the data is to use the RDN Hash method. To split data with the RDN Hash method, the user specifies the base on which to split the LDIF file, and how many servers he wants the data split into. Given this information, the tool splits the LDIF file into separate LDIF files by putting the DN of each child entry through a hash function. This provides a uniform distribution of child entries over a small number of files. See Appendix M, "Distributed directory setup tool options," on page 587 for more information.

Adding and partitioning the data

Entries are added using either the Web Administration Tool (see "Adding an entry" on page 294 for additional information) or the command line (see "`idsldapmodify`, `ldapmodify`, `idsldapadd`, `ldapadd`" on page 396 commands).

In this scenario you are going to add the data contained in the `sample.ldif` file that is included with the IBM Tivoli Directory Server. Issue the following command:

```
idsldapadd -D cn=user1,cn=ibmpolicies -w mysecret -h <proxyhostname> -p 389
-i <IDS_LDAP_HOME>/examples/sample.ldif
```

Where `IDS_LDAP_HOME` varies by operating system platform:

- AIX operating systems - `/opt/IBM/ldap/V6.0`
- HP-UX operating systems - `/opt/IBM/ldap/V6.0`
- Linux operating systems - `/opt/ibm/ldap/V6.0`
- Solaris operating systems - `/opt/IBM/ldap/V6.0`
- Windows operating systems - `<local_drive>:\Program Files\IBM\LDAP\V6.0`
(This is the default install location. The actual `IDS_LDAP_HOME` is determined during installation.)

If you have an existing database with a large number of entries, you need to export the entries to an LDIF file. See "`idsdb2ldif`, `db2ldif`" on page 436 for information on how to do this:

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

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

2. Create a configuration file, for example `newhash.conf`, where `newhash.conf` contains:

```
BaseDirectory: <BaseDirectory>
Actiontype : SplitOnly
inputFile : <pathname>/mydata.ldif
defaultOutputFile : default.ldif
SplitBY: rdhash
```

```
BaseDN: o=ibm,c=US
URL: ldap://ServerA:389
FILE: ServerA.ldif
URL: ldap://ServerB:169
File: ServerB.ldif
URL: ldap://ServerC:389
File: ServerC.ldif:
URL: ldap://ServerD:389
File: ServerD.ldif
URL: ldap://ServerE:389
File: ServerE.ldif
```

Notes:

- a. If you specify the BaseDirectory parameter, it must come before any files in the configuration file. This includes the default ldif output file, the logfile, and any other output LDIF files.
 - b. Specify URL and File for each server under the BaseDN.
 - c. The URL and File stanzas specified for each server are order dependent.
3. Issue the command:
- ```
ddsetup -f newhash.conf
```

The ddsetup command divides the mydata.ldif file into multiple LDIF output files. 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.

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

When partitioning an existing directory containing AES-formatted data into a distributed directory, the partition servers must be synchronized with the original unpartitioned server. If not, LDIF export files produced by the ddsetup tool will fail to import.

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 I, “Synchronizing two-way cryptography between server instances,” on page 537 for information about synchronizing directory server instances.

4. Use idsldif2db or idsbulkload to load the data to the appropriate backend server.
  - ServerA (partition index 1) - ServerA.ldif
  - ServerB (partition index 2) - ServerB.ldif
  - ServerC (partition index 3) - ServerC.ldif
  - ServerD (partition index 4) - ServerD.ldif
  - ServerE (partition index 5) - ServerE.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.

For more information about the `ddsetup` utility, see “`ddsetup`” on page 420.

---

## Synchronizing information

There are two main kinds of configuration information that must be kept synchronized among the servers in a distributed directory.

### Subtree policies

ACLs are currently the only type of subtree policy. ACLs are honored locally within a server only. When data is split across a flat container each server contains the parent entry. If ACLs are defined on the parent entry, they must be defined on each of the parent entries. ACLs defined at the parent level or below must not have any dependencies on entries above the parent entry in the tree. The server does not enforce ACLs defined on another server.

At setup time, exact copies of the entire parent entry are added to each server if `ddsetup` is used; otherwise, it is the user’s responsibility to add copies of the entire parent entry to the server. If the parent entry has ACLs defined on it, each server has the same ACLs for the entries below the parent after initial configuration. Any changes made to the parent entries after initial configuration have to be sent to each server containing the parent entry without using the proxy server. It is the administrator’s responsibility to keep the parent entries (including the ACLs on the parent) synchronized among the servers.

### Global policies including schema and password policy

The `cn=pwdpolicy` subtree, `cn=ibmpolicies` subtree and `cn=schema` subtree store global configuration and must be replicated among the servers in a distributed directory. Set gateway replication agreements under the `cn=ibmpolicies` subtree, so that if any of the servers have a replica, the change is passed on to their individual replica. With the `cn=ibmpolicies` replication agreement, the `cn=schema` and `cn=pwdpolicy` subtrees are automatically replicated. Global policies include the global administration group entry stored under `cn=ibmpolicies`. See “Global administration group” on page 252 for more information.

#### Notes:

1. The global policies are not replicated to the proxy server.
2. Changes to `cn=schema` are not replicated to all the servers.

**Attention:** When you create a new directory server instance, be aware of the information that follows. If you want to use a distributed directory, 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 I, “Synchronizing two-way cryptography between server instances,” on page 537 for information about synchronizing directory server instances.

---

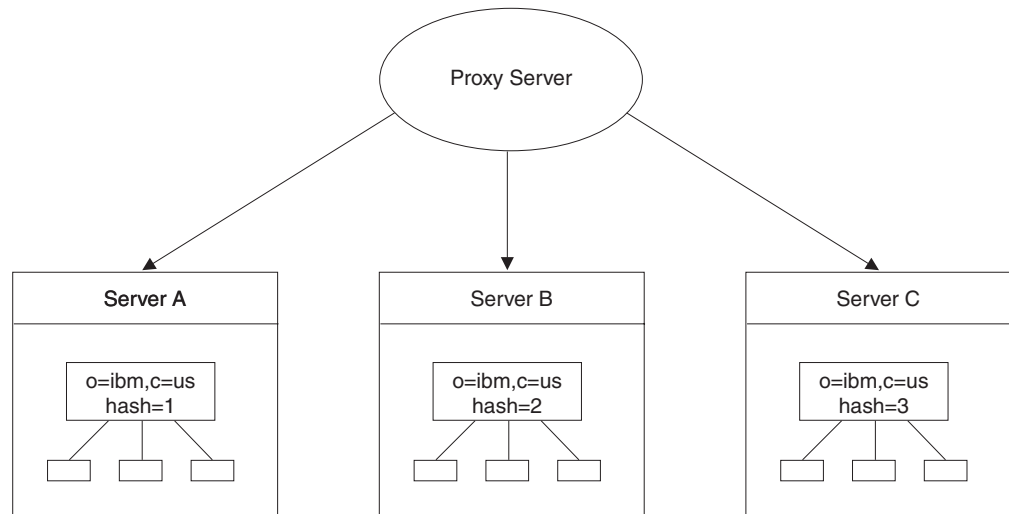
## Partition entries

Entries that exist as the base of a partition, for example, `o=ibm,c=us`, cannot be accessed through the proxy server. The proxy server can return one of these entries during a search (the proxy searches for duplicates, and any entry returned is a random entry), but these entries cannot be modified using the proxy server.

---

## Setting up a distributed directory with a proxy server

The following scenario shows how to set up the a proxy server and a distributed directory with three partitions for the subtree `o=ibm,c=us`.



## Setting up the back-end servers

Use one of the following methods to set up the back-end servers:

### Using Web Administration

**Adding the suffix to the backend servers:** To add the suffix, use one of the following methods.

1. Log on to ServerA, expand the **Manage server properties** category in the navigation area of the Web Administration Tool, select the **Suffixes** tab.
2. Enter the Suffix DN, `o=ibm,c=us`.
3. Click **Add**.
4. Repeat this process for as many suffixes as you want to add.
5. When you are finished, click **Apply** to save your changes without exiting, or click **OK** to apply your changes and exit.
6. Repeat this procedure for ServerB and ServerC.

For more information see “Adding and removing suffixes” on page 108.

**Global administration group:** The global administration group is a way for the directory administrator to delegate administrative rights in a distributed environment to the database backend. Global administrative group members are users that have been assigned the same set of privileges as the administrative group with regard to access entries in the database backend and have complete access to the directory server backend. All global administrative group members have the same set of privileges.

Global administrative group members have no privileges or access rights to any data or operations that are related to the configuration settings of the directory server. This is commonly called the configuration backend.

Global administrative group members cannot access schema data.

Global administrative group members also do not have access to the audit log. Local administrators, therefore, can use the audit log to monitor global administrative group member activity for security purposes.

**Note:** The global administration group should be used by applications or administrators to communicate with the proxy server using administrative credentials. For example, the member that was set up using these instructions (cn=manager,cn=ibmpolicies) should be used in place of the local administrator (cn=root) when directory entries are to be modified through the proxy server. Binding to the proxy server as cn=root gives an administrator full access to the proxy server's configuration, but only anonymous access to the directory entries.

#### **Creating a user entry for membership in the global administrators group:**

1. Log onto ServerA. This is the server that you specified as the partition for cn=ibmpolicies.
2. Start the server.
3. From the navigation area, expand the **Directory management** topic.
4. Click **Add an entry**. See "Adding an entry" on page 294 for additional information.
5. From the **Structural object class** drop-down menu, select **person**.
6. Click **Next**.
7. Click **Next** to skip the **Select auxiliary object classes** panel.
8. Type **cn=manager** in the **Relative DN** field.
9. Type **cn=ibmpolicies** in the **Parent DN** field.
10. Type **manager** in the **cn** field.
11. Type **manager** in the **sn** field.
12. Click the **Optional attributes** tab.
13. Type a password in the **userPassword** field. For example **mysecret**.
14. Click **Finish**.

**Adding the user entry to the global administration group:** The following steps add cn=manager to the global administration group.

1. In the navigation area, click **Manage entries**.
2. Select the radio button for cn=ibmpolicies and click **Expand**.
3. Select the radio button for globalGroupName=GlobalAdminGroup and from the **Select action** drop-down menu select **Manage members** and click **Go**.
4. Type **cn=manager,cn=ibmpolicies** in the member field and click **Add**.
5. A message is displayed: You have not loaded entries from the server. Only your changes will be displayed in the table. Do you want to continue?, click **OK**.
6. cn=manager is displayed in the table. Click **Ok**. cn=manager is now a member of the global administration group.

## Using the command line

**Creating a user entry for membership in the global administrators group:** Issue the commands:

```
idsldapadd -h <ServerA> -D <admin_dn> -w <admin_pw> -f <LDIF1>
idsldapadd -h <ServerA> -D <admin_dn> -w <admin_pw> -f <LDIF2>
```

where <LDIF1> contains:

```
dn: cn=manager,cn=ibmpolicies
objectclass: person
sn: manager
cn: manager
userpassword: secret
```

and where <LDIF2> contains:

```
dn: globalGroupName=GlobalAdminGroup,cn=ibmpolicies
changetype: modify
add: member
member: cn=manager,cn=ibmpolicies
```

**Adding the user entry to the global administration group:** Issue the commands:

```
idsldapadd -h <ServerA> -D <admin_dn> -w <admin_pw> -f <LDIF1>
idsldapadd -h <ServerA> -D <admin_dn> -w <admin_pw> -f <LDIF2>
```

where <LDIF1> contains:

```
dn: cn=manager,cn=ibmpolicies
objectclass: person
sn: manager
cn: manager
userpassword: secret
```

and where <LDIF2> contains:

```
dn: globalGroupName=GlobalAdminGroup,cn=ibmpolicies
changetype: modify
add: member
member: cn=manager,cn=ibmpolicies
```

## Setting up the proxy server

Use one of the following methods to set up the proxy server:

### Using Web Administration

#### Configuring the proxy server:

**Note:** If the server you are configuring as a proxy server contains the entry data that you want to distribute across the directory, you must extract the entry data into an LDIF file before you configure the server. After the server is configured as a proxy server you cannot access the data that is contained in its RDBM. If you need to access the data in its RDBM, you can either reconfigure the server so that it is not a proxy or create a new directory server instance that points to the RDBM as its database.

1. Log onto the server that you are going to use as the proxy server.
2. Start the server in configuration only mode.
3. From the navigation area expand the **Proxy administration** topic.
4. Click **Manage proxy properties**.
5. Click the **Configure as proxy server** check box.

6. In the **Suffix DN** field enter **cn=ibmpolicies** and click **Add**.
7. In the **Suffix DN** field enter **o=ibm,c=us** and click **Add**.
8. In the **Suffix DN** field enter **cn=pwdpolicy** and click **Add**.
9. Click **OK** to save your changes and return to the **Introduction** panel.

**Note:** You must log off the Web Administration, and log in again. Doing so will update the navigation area. If you do not log off and then log on again, the navigation area is not updated for a proxy server.

#### Identifying the distributed directory servers to the proxy server:

1. From the navigation area, click **Manage back-end directory servers**.
2. Click **Add**.
3. Enter the host name for ServerA in the **Hostname** field.
4. Enter the port number for ServerA (for this example all servers use 389).
5. Enter the number of connections that the proxy server can have with the back-end server in the **Connection pool size** field. The minimum value is 1 and the maximum value is 100. For this example, set the value to 5.

**Note:** For this release, do not set the value in the **Connection pool size** field to be less than 5.

6. Specify **Simple** in the **Authentication method** field.
7. Click **Next**.
8. Specify the administration DN or the DN of a member of the local administrator in the **Bind DN** field. For example, **cn=root**.
9. Specify and confirm the administration password, in the **Bind password** fields. For example, **secret**.
10. Click **Finish**.
11. Repeat steps 2 through 10 for ServerB and ServerC.
12. When you are finished, click **Close** to save your changes and return to the **Introduction** panel.
13. Ensure that all the back-end servers are started.

**Note:** If the proxy server cannot connect with one or more of the back-end servers at start up, the proxy starts in configuration mode only. This is true unless you set up server groups. See “Server groups” on page 261.

**Synchronizing global policies:** These steps set up **cn=ibmpolicies** as a single partition. This is necessary to enable you to synchronize the global policies on all of the servers.

**Note:** Schema modifications are not replicated by or to the proxy server. Any schema updates need to be entered on each proxy server manually.

1. From the navigation area, click **Manage partition bases**.
2. On the **Partition bases** table, click **Add**.
3. Enter **cn=ibmpolicies** in the **Partition DN** field.
4. Enter **1** in the **Number of partitions** field.

**Note:** A value greater than 1 for **cn=ibmpolicies** and **cn=pwdpolicy** is not supported.

5. Click **OK**.
6. Select the radio button for **cn=ibmpolicies** and click **View servers**.

7. Verify that `cn=ibmpolicies` is displayed in the **Partition base DN** field.
8. In the **Back-end directory servers for partition base** table, click **Add**.
9. From the **Back-end directory server** menu, select `ServerA`.
10. Enter `1` in the **Partition index** field.
11. Click **OK**. Doing this enables you to have the global administration group member entry on a single back-end server instead of having to create it on each of the back-end servers.
12. Repeat steps 1 through 11 for `cn=pwdpolicy`.

**Dividing the data into partitions:** These steps divide the data in the subtree `o=ibm,c=us` into three partitions.

1. On the **Partition bases** table, click **Add**.
2. Enter `o=ibm,c=us` in the **Partition DN** field.
3. Enter `3` in the **Number of partitions** field.
4. Click **OK**.

**Assigning partition index values to the servers:** These steps assign a partition value to each of the servers.

1. Select the radio button for `o=ibm,c=us` and click **View servers**.
2. Verify that `o=ibm,c=us` is displayed in the **Partition base DN** field.
3. In the **Back-end directory servers for partition base** table, click **Add**.
4. From the **Back-end directory server** drop-down menu, select `ServerA`.
5. Ensure that `1` is displayed in the **Partition index** field.
6. Click **OK**.
7. In the **Back-end directory servers for partition base** table, click **Add**.
8. From the **Back-end directory server** drop-down menu, select `ServerB`.
9. Ensure that `2` is displayed in the **Partition index** field.

**Note:** This number is automatically incremented for you. You can manually change the partition index number, however, it cannot exceed the actual number of partitions for the base. For example, you cannot use `4` as a partition index, if the partition base has only three partitions. Duplicate partition indexes are only allowed on servers participating in replication on that subtree.

10. Click **OK**.
11. In the **Back-end directory servers for partition base** table, click **Add**.
12. From the **Back-end directory server** drop-down menu, select `ServerC`.
13. Ensure that `3` is displayed in the **Partition index** field.
14. Click **OK**.
15. When you are finished, click **Close**.
16. Restart the proxy server for the changes to take effect.

## Using the command line

**Configuring the proxy server:** Issue the commands:

```
idsldapmodify -h <Proxy Server> -D <admin_dn> -w <admin_pw> -i <LDIF1>
idsldapmodify -h <Proxy Server> -D <admin_dn> -w <admin_pw> -i <LDIF2>
```

where `<LDIF1>` contains:

```
dn: cn=Configuration
changetype: modify
replace: ibm-slapdServerBackend
ibm-slapdServerBackend: PROXY
```

and where <LDIF2> contains:

```
dn: cn=ProxyDB, cn=Proxy Backends, cn=IBM Directory, cn=Schemas, cn=Configuration
changetype: modify
add: ibm-slapdSuffix
ibm-slapdSuffix: cn=ibmpolicies
ibm-slapdSuffix: cn=pwdpolicy
ibm-slapdSuffix: o=ibm,c=us
```

**Identifying the distributed directory servers to the proxy server:** Issue the commands:

```
idsldapadd -h <Proxy Server> -D <admin_dn> -w <admin_pw> -f <LDIF1>
```

where <LDIF1> contains:

```
dn: cn=Server1, cn=ProxyDB, cn=Proxy Backends, cn=IBM Directory, cn=Schemas,
 cn=Configuration
cn: Server1
ibm-slapdProxyBindMethod: Simple
ibm-slapdProxyConnectionPoolSize: 5
ibm-slapdProxyDN: cn=root
ibm-slapdProxyPW: secret
ibm-slapdProxyTargetURL: ldap://ServerA:389
objectClass: top
objectClass: ibm-slapdProxyBackendServer
objectClass: ibm-slapdConfigEntry
```

```
dn: cn=Server2, cn=ProxyDB, cn=Proxy Backends, cn=IBM Directory, cn=Schemas,
 cn=Configuration
cn: Server2
ibm-slapdProxyBindMethod: Simple
ibm-slapdProxyConnectionPoolSize: 5
ibm-slapdProxyDN: cn=root
ibm-slapdProxyPW: secret
ibm-slapdProxyTargetURL: ldap://ServerB:389
objectClass: top
objectClass: ibm-slapdProxyBackendServer
objectClass: ibm-slapdConfigEntry
```

```
dn: cn=Server3, cn=ProxyDB, cn=Proxy Backends, cn=IBM Directory, cn=Schemas,
 cn=Configuration
cn: Server3
ibm-slapdProxyBindMethod: Simple
ibm-slapdProxyConnectionPoolSize: 5
ibm-slapdProxyDN: cn=root
ibm-slapdProxyPW: secret
ibm-slapdProxyTargetURL: ldap://ServerC:389
objectClass: top
objectClass: ibm-slapdProxyBackendServer
objectClass: ibm-slapdConfigEntry
```

**Dividing the data into partitions and assigning partition index values to the servers:** Issue the commands:

```
idsldapadd -h <Proxy Server> -D <admin_dn> -w <admin_pw> -f <LDIF2>
```

where <LDIF2> contains:

```
dn: cn=cn\=ibmpolicies split, cn=ProxyDB, cn=Proxy Backends, cn=IBM Directory,
 cn=Schemas, cn=Configuration
cn: cn=ibmpolicies split
ibm-slapdProxyNumPartitions: 1
```

```

ibm-slapdProxyPartitionBase: cn=ibmpolicies
objectclass: top
objectclass: ibm-slapdConfigEntry
objectclass: ibm-slapdProxyBackendSplitContainer

dn: cn=split1, cn=cn\=ibmpolicies split, cn=ProxyDB, cn=Proxy Backends,
 cn=IBM Directory, cn=Schemas, cn=Configuration
cn: split1
ibm-slapdProxyBackendServerDN: cn=Server1,cn=ProxyDB,cn=Proxy Backends,
 cn=IBM Directory,cn=Schemas,cn=Configuration
ibm-slapdProxyPartitionIndex: 1
objectclass: top
objectclass: ibm-slapdConfigEntry
objectclass: ibm-slapdProxyBackendSplit

dn: cn=cn\=pwdpolicy split, cn=ProxyDB, cn=Proxy Backends, cn=IBM Directory,
 cn=Schemas, cn=Configuration
cn: cn\=pwdpolicy split
ibm-slapdProxyNumPartitions: 1
ibm-slapdProxyPartitionBase: cn=pwdpolicy
objectclass: top
objectclass: ibm-slapdConfigEntry
objectclass: ibm-slapdProxyBackendSplitContainer

dn: cn=split1,cn=cn\=pwdpolicy split, cn=ProxyDB, cn=Proxy Backends,
 cn=IBM Directory, cn=Schemas, cn=Configuration
cn: split1
ibm-slapdProxyBackendServerDN: cn=Server1,cn=ProxyDB,cn=Proxy Backends,
 cn=IBM Directory,cn=Schemas,cn=Configuration
ibm-slapdProxyPartitionIndex: 1
objectclass: top
objectclass: ibm-slapdConfigEntry
objectclass: ibm-slapdProxyBackendSplit

dn: cn=o\=ibm\,c\=us split, cn=ProxyDB, cn=Proxy Backends, cn=IBM Directory,
 cn=Schemas, cn=Configuration
cn: o=ibm\c=us split
ibm-slapdProxyNumPartitions: 3
ibm-slapdProxyPartitionBase: o=ibm,c=us
objectclass: top
objectclass: ibm-slapdConfigEntry
objectclass: ibm-slapdProxyBackendSplitContainer

dn: cn=split1, cn=o\=ibm\,c\=us split, cn=ProxyDB, cn=Proxy Backends,
 cn=IBM Directory, cn=Schemas, cn=Configuration
cn: split1
ibm-slapdProxyBackendServerDN: cn=Server1,cn=ProxyDB,cn=Proxy Backends,
 cn=IBM Directory,cn=Schemas,cn=Configuration
ibm-slapdProxyPartitionIndex: 1
objectclass: top
objectclass: ibm-slapdConfigEntry
objectclass: ibm-slapdProxyBackendSplit

dn: cn=split2, cn=o\=ibm\,c\=us split, cn=ProxyDB, cn=Proxy Backends,
 cn=IBM Directory, cn=Schemas, cn=Configuration
cn: split2
ibm-slapdProxyBackendServerDN: cn=Server2,cn=ProxyDB,cn=Proxy Backends,
 cn=IBM Directory,cn=Schemas,cn=Configuration
ibm-slapdProxyPartitionIndex: 2
objectclass: top
objectclass: ibm-slapdConfigEntry
objectclass: ibm-slapdProxyBackendSplit

dn: cn=split3, cn=o\=ibm\,c\=us split, cn=ProxyDB, cn=Proxy Backends,
 cn=IBM Directory, cn=Schemas, cn=Configuration
cn: split3
ibm-slapdProxyBackendServerDN: cn=Server3,cn=ProxyDB,cn=Proxy Backends,

```



```
cn=IBM Directory,cn=Schemas,cn=Configuration
ibm-slapdProxyPartitionIndex: 3
objectclass: top
objectclass: ibm-slapdConfigEntry
objectclass: ibm-slapdProxyBackendSplit
```

---

## Failover and load balancing

The proxy server is aware of all of the replicas of a given partition, and load balances read requests between the online replicas. The proxy server is aware of all of the masters for a given partition, and must use one of these as the primary master. The first master found in the partition is the primary master. If the primary master server is down, the proxy server is capable of failing over to a backup server (one of the other master servers). If the requested operation cannot be performed by the currently online servers, the proxy server returns an operations error.

**Note:** For better performance, all backend servers and the proxy server should share the same stash files.

The proxy server performs load balancing for read requests, and fail over for update requests.

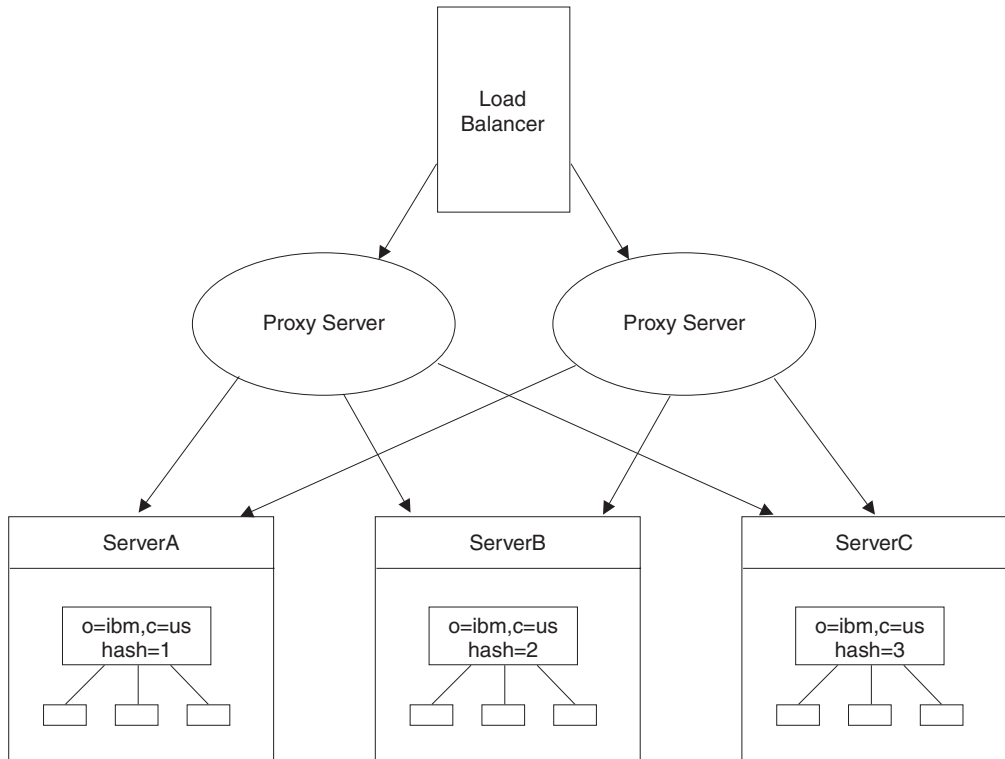
If a backend server is unavailable, the operation will error out. All subsequent operations will fail over to the next available server.

---

## Failover between proxy servers

In a proxied directory, failover support between proxies is provided by creating an additional proxy server that is identical to the first proxy server. These are not the same as peer masters, the proxy servers have no knowledge of each other and must be managed through a load balancer.

A load balancer, such as the IBM WebSphere Edge Server, has a virtual host name that applications use when sending updates to the directory. The load balancer is configured to send those updates to only one server. If that server is down, or unavailable because of a network failure, the load balancer sends the updates to the next available proxy server until the first server is back on line and available. Refer to your load balancer product documentation for information on how to install and configure the load balancing server.

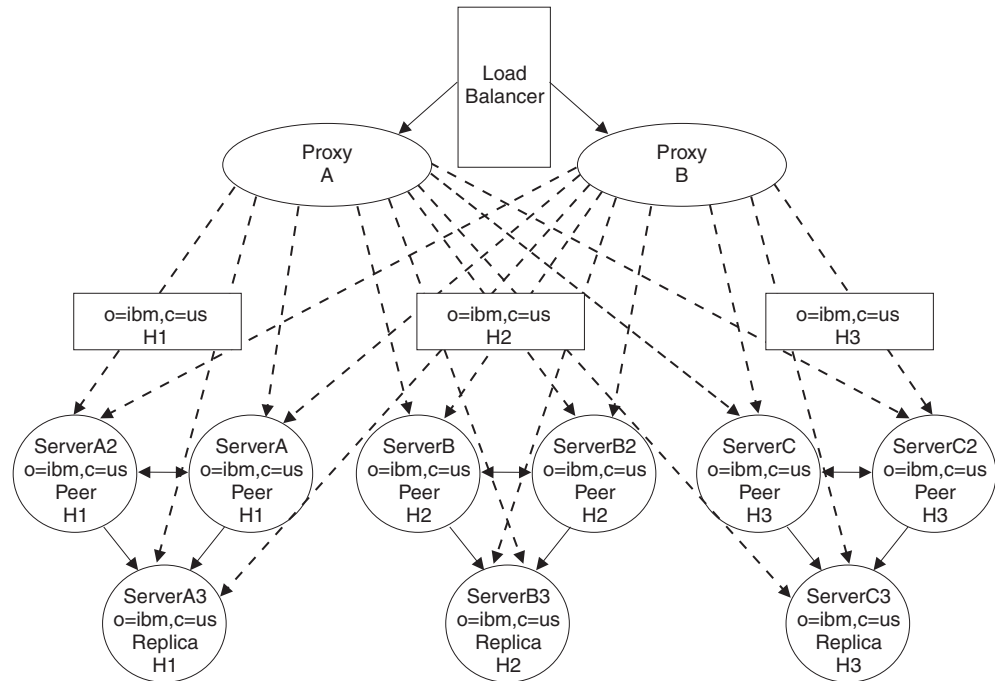


**Note:** In a load-balanced proxy environment, if a proxy server fails, the first operation sent to it fails and returns an error. All subsequent operations are sent to the failover proxy server. The first operation that failed can be retried. It is not automatically sent to the failover server.

---

## Setting up backup replication for a distributed directory with proxy servers

In this example you are going to set up a distributed directory and use replication to provide read and write backup capabilities. The three partitions for the suffix `o=ibm,c=us` has a corresponding hash value (H1, H2, or H3). Each partition has its own replication site consisting of two peer servers and a replica to provide the read write backup capabilities. Each proxy server has knowledge of all the servers in the topology (indicated by the dashed connections). The relationships among the servers in each replication site is represented by the solid lines.



To create this scenario you must

1. Create an LDIF file for the data you are going to partition. See “Creating an LDIF file for your data entries” on page 262
2. Create a replication topology for the data subtree. See “Setting up the replication topology” on page 263.
3. Create a second replication topology for the cn=ibmpolicies subtree. See “Setting up a topology for global policies” on page 264.
4. Set up the proxy servers. See “Setting up the proxy servers” on page 264
5. Partition existing data. See “Partitioning the data” on page 265.
6. Load the data. See “Loading the partitioned data” on page 265.
7. Start replication. See “Starting replication” on page 266

For more information about setting up replication, see Chapter 13, “Replication,” on page 161.

## Server groups

If the proxy server is unable to contact a backend server, or if authentication fails, then proxy server startup fails and the proxy server starts in configuration only mode by default, unless server groupings have been defined in the configuration file.

Server groupings enable the user to state that several backend servers are mirrors of each other, and proxy server processing can continue even if one or more backend servers in the group is down, assuming that at least one backend server is online. Connections are restarted periodically if the connections are closed for some reason, such as the remote server is stopped or restarted.

The proxy configuration file supports a special set of entries that enable a directory administrator to define server groups in the configuration file. Each group contains a list of backend servers. As long as at least one backend server in each group can be contacted, the proxy server will start successfully and service client requests,

though performance might be degraded. Each backend server in the entry is defined to have an OR relationship, and all the entries have an AND relationship.

In this release, the Web Administration Tool does not support the management of these server groups. The directory administrator must define these server groups using `idsldapadd` and `idsldapmodify` to add and modify the required entries. The directory administrator must ensure that each of the backend servers is placed in a server group and that the backend servers in each server group contain the same partition of the directory database. For example, suppose that `server1` and `server2` are peers of each other, with `server3` and `server4` being separate peers, that is, `server1` and `server2` hold a disjoint data set from `server3` and `server4`. In this case, a user would add `server1` and `server2` in a server group entry under the `cn=configuration` suffix, and `server3` and `server4` in a separate server group entry. If either `server1` or `server2` is up, then the proxy server can proceed to check if either `server3` or `server4` is online. If neither `server3` or `server4` is up, then the proxy server starts in configuration only mode.

In addition to the server grouping, the administrator must add the `serverID` of each backend server in the server group entry. If the server is down, no root DSE information can be gained, and the `serverID` is needed for determining the supplier/consumer relationships throughout the topology.

Any backend servers not in a server group that are offline at proxy server startup cause the proxy server to start in configuration only mode.

The following is an example of user-defined server groupings:

```
dn: cn=serverGroup, cn=ProxyDB, cn=Proxy Backends, cn=IBM Directory, cn=Schemas,
 cn=Configuration
cn: serverGroup
ibm-slapdProxyBackendServerDN: cn=Server1,cn=ProxyDB,cn=Proxy Backends,
 cn=IBM Directory, cn=Schemas,cn=Configuration
ibm-slapdProxyBackendServerDN: cn=Server2,cn=ProxyDB,cn=Proxy Backends,
 cn=IBM Directory, cn=Schemas,cn=Configuration
objectclass: top
objectclass: ibm-slapdConfigEntry
objectclass: ibm-slapdProxyBackendServerGroup
```

**Notes:**

1. In each entry pointed to by `ibm-slapdProxyBackendServerDn`, the attribute `ibm-slapdServerId` must be added, with its value identical to the value on the corresponding backend server.
2. Web Administration Tool support for server groupings is not available. It is the administrator's responsibility to keep these entries in sync and correct with the distributed configuration. LDAP protocol must be used to maintain the entries.

## Creating an LDIF file for your data entries

To create an LDIF file (`mydata.ldif`) for the data entries in the subtree `o=ibm,c=us` if they currently reside on a server:

- Issue the command:

```
idsdb2ldif -o mydata.ldif -s o=ibm,c=us -I <instance_name>
-k <key seed> -t <key salt>
```

**Note:** You must use the `-I` option if there is more than one instance. You must use the `-k` and `-t` options if keys on the server are not in sync.

**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*, see 177.

See “idsdb2ldif, db2ldif” on page 436 for more information on how to do this.

## Setting up the replication topology

Ensure that you understand replication concepts and terms before attempting to create this scenario. See Chapter 13, “Replication,” on page 161, if you do not understand the concept of replication.

In this topology created using the Web Administration Tool, each partition is treated as a separate replication site. However, there are no gateway servers in this topology because you do not want the partitioned data to be replicated to the other partitions.

**Note:** At this point you are creating the topology. Do not load any entry data.

1. Log onto ServerA and, if you have not already done so, add the subtree o=ibm,c=us. Doing this makes ServerA a master server for o=ibm,c=us. See “Adding a subtree” on page 222.
2. Create a set of credentials for the topology. See “Adding credentials” on page 224.
3. Add ServerA2 as a peer-master server. See “Adding a peer-master or gateway server” on page 228.
4. Add ServerA3 as a replica. Ensure that the supplier agreement with ServerA2 is selected. See “Adding a replica server” on page 231.

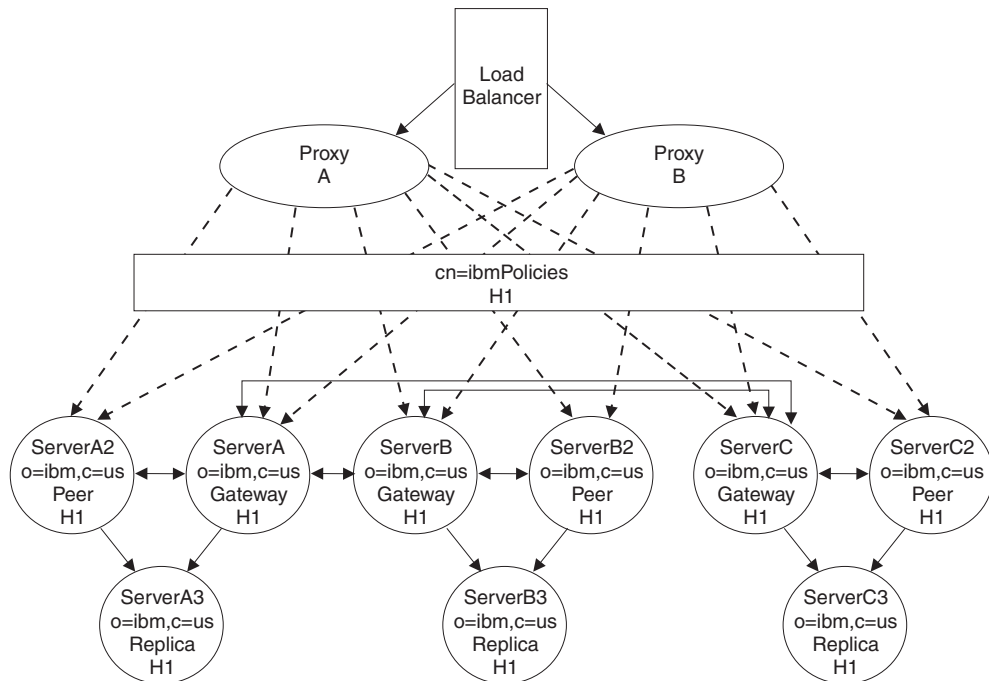
**Note:** You can either log on to ServerB and ServerC to create similar topologies as you did with ServerA or continue to create the topology from ServerA. Remember that if you continue to add the topology from ServerA, you must deselect any agreements that the Web Administration Tool tries to create that are not appropriate for the topology. For example, no agreement can exist between any of the “A” servers and any of the “B” or “C” servers. Conversely, none of the “B” servers can have any agreements with any of the “A” or “C” servers.

5. Add ServerB as a master server for the subtree o=ibm,c=us. See “Adding a peer-master or gateway server” on page 228. Remember to deselect any agreements with ServerA, Server A2, and ServerA3.
6. Add ServerB2 as a peer-master server of Server B. See “Adding a peer-master or gateway server” on page 228. Remember to deselect any agreements with ServerA, Server A2, and ServerA3.
7. Add ServerB3 as a replica. Deselect any supplier agreements from ServerA and ServerA2 are selected. See “Adding a replica server” on page 231.
8. Add ServerC as a master server for the subtree o=ibm,c=us. See “Adding a peer-master or gateway server” on page 228. Remember to deselect any agreements with ServerA, Server A2, ServerA3, ServerB, ServerB2, and ServerB3.
9. Add ServerC2 as a peer-master server of Server B. See “Adding a peer-master or gateway server” on page 228. Remember to deselect any agreements with ServerA, Server A2, ServerA3, ServerB, ServerB2, and ServerB3.
10. Add ServerC3 as a replica. Deselect any supplier agreements from ServerA, ServerA2, ServerB, and ServerB2. See “Adding a replica server” on page 231.

For more information about setting up replication, see Chapter 13, “Replication,” on page 161.

## Setting up a topology for global policies

You need to set up a second topology for the `cn=ibmPolicies` subtree to replicate global policy updates. For example you could use the same topology setup that you created for `o=ibm,c=us` and make ServerA, ServerB, and ServerC gateway servers.



In this topology any updates made to any one of the servers is updated to all the servers.

Ensure that you create the appropriate agreements between the replication sites. See “Setting up a gateway topology” on page 207 and “Managing gateway servers” on page 234 for information on how to set up this kind of a topology.

You do not have to use the same topology model that you set up for the data subtree. You could create a topology in which servers A, A2, B, B2, C, and C2 are all peer servers with agreements amongst themselves and the replica servers A3, B3, and C3. The only requirement is that all the servers in your data subtree topology are included in the `cn=ibmpolicies` subtree topology.

**Note:** Remember that schema changes are not replicated by the proxy servers. Entries that update the schema must be made on each of the proxy servers and on one of the peer-master servers in the `cn=ibmpolicies` topology.

## Setting up the proxy servers

1. Set up proxy server Proxy A:

Follow the directions in “Setting up the proxy server” on page 254 to set up your proxy server. Remember that when the instructions tell you to repeat steps for ServerB and ServerC, you need to perform those steps for ServerA2, ServerA3, ServerB2, ServerB3, ServerC2, and ServerC3 as well.

**Note:** Remember to assign the correct partition values, when assigning partition values to the backend servers.

| Server name | Partition index value |
|-------------|-----------------------|
| ServerA     | 1                     |
| ServerA2    | 1                     |
| ServerA3    | 1                     |
| ServerB     | 2                     |
| ServerB2    | 2                     |
| ServerB3    | 2                     |
| ServerC     | 3                     |
| ServerC2    | 3                     |
| ServerC3    | 3                     |

2. Set up the second proxy server, Proxy B, the same way you set up Proxy A.
3. Add a load balancer such as the IBM WebSphere Edge Server.

## Partitioning the data

To partition the data contained in the mydata.ldif file you created for the subtree o=ibm,c=us, you must:

1. Create a configuration file, for example **newhash.conf**. Where newhash.conf contains:

```
BaseDirectory: <BaseDirectory>
Action type : splitonly
inputFile : <pathname>/mydata.ldif
defaultOutputFile : default.ldif
SplitBY: rdnhash
```

```
BaseDN: o=ibm,c=US
URL: ldap://ServerA:389
FILE: out1.ldif
URL: ldap://ServerB:389
File: out2.ldif
URL: ldap://ServerC:389
File: out3.ldif
```

2. Issue the command:  
ddsetup -f newhash.conf

## Loading the partitioned data

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.

Depending upon the amount of your data, use idsldif2db or idsbulkload to load the data to the appropriate backend servers. Again, depending on the amount of data, loading the appropriate LDIF file to each server might be more efficient than having the data replicated.

- ServerA (partition index 1) - ServerA.ldif
- ServerA2 (partition index 1) - ServerA.ldif
- ServerA3 (partition index 1) - ServerA.ldif
- ServerB (partition index 2) - ServerB.ldif
- ServerB2 (partition index 2) - ServerB.ldif

- ServerB3 (partition index 2) - ServerB.ldif
- ServerC (partition index 3) - ServerC.ldif
- ServerC2 (partition index 3) - ServerC.ldif
- ServerC3 (partition index 3) - ServerC.ldif

## Starting replication

If replication has not automatically started, you will need to unquiesce the subtree and restart the queues for each of the servers. See “Quiescing the subtree” on page 223 and “Managing queues” on page 240 for information on how to do those tasks.



---

## Chapter 15. Logging Utilities

The IBM Tivoli Directory Server Version 6.0 provides several logging utilities that can be viewed either through the Web Administration Tool or the system command line.

- “Modifying default log settings” on page 269
- “Modifying administration daemon error log settings” on page 270
- “Enabling the administration daemon audit log and modifying administration audit log settings” on page 271
- “Enabling the audit log and modifying audit log settings” on page 274
- “Modifying bulkload error log settings” on page 279
- “Modifying configuration tools log settings” on page 280
- “Modifying DB2 error log settings” on page 281
- “Modifying lost and found log settings” on page 283
- “Modifying the server error log” on page 284

### Notes:

1. In the Web Administration Tool the **Logfiles** link in each task title bar accesses the Web Administration console log files. The IBM Tivoli Directory Server log files are accessible by using the procedures specified in the following sections.
2. On Windows-based systems, if a path begins with the drive letter and a colon, it is assumed to be the full path. A path without the drive letter, starts in the installation tree. As examples: `c:\tmp\mylog` is a full path, while `\tmp\mylog` is interpreted as `c:\idsslapd-<instancename>\tmp\mylog`.

Only the administrator or members of the administrative group can view or access log information.

---

## Default log paths

The default log path for all logs is:

UNIX path:

```
<instance base directory>/idsslapd-<instance name>/logs
```

Where:

- *instance base directory* is the home directory of the directory server instance owner, or the directory you specified when you created the directory server instance.
- *instance name* is the name of the directory server instance.

Windows path:

```
<drive>\idsslapd-<instance name>\logs
```

Where:

- *drive* is the drive you specified when you created a directory server instance.
- *instance name* is the name of the directory server instance.

---

## Log management tool

**Note:** The log management tool requires that you have the IBM Tivoli Directory Integrator installed.

The log management tool enables the LDAP administrator to limit the size of log files. The tool process, `idslogmgmt`, wakes up 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. Except for the administrative tools' and the `idslogmgmt`'s log, the configuration settings for the logs are located in the `ibmslapd` configuration file. See the *IBM Tivoli Directory Server version 6.0 Installation and Configuration Guide* for more information. Also see "idslogmgmt" on page 449.

---

## Default log management

A new configuration entry is created for the default log file management. This entry contains the default log settings for the all logs with the exception of the `ibm-slapdLog` attribute. These settings can be overridden in the specific log management entries described in the following section. By default the entry will not have attributes; hence, there will be no log limits enforced. Here is the description of the entry:

```
dn: cn=default, cn=Log Management, cn=Configuration
ibm-slapdLogSizeThreshold:
ibm-slapdLogMaxArchives:
ibm-slapdLogArchivePath:
objectclass: top
objectclass: ibm-slapdLogConfig
objectclass: ibm-slapdConfigEntry
objectclass: container
```

The following attributes are defined:

### **ibm-slapdLogSizeThreshold**

When this size threshold, in MB, is exceeded the file will be archived.

### **ibm-slapdLogMaxArchives**

The maximum number of archived logs.

### **ibm-slapdLogArchivePath**

The path where the archived logs will be placed.

By default, the `idslogmgmt` application logs data to the following file on UNIX:

```
/var/idsldap/V6.0/idslogmgmt.log
```

and to the following file on Windows:

```
<install_directory>\var\idslogmgmt.log
```

The following are the default values for the log management of `idslogmgmt.log`:

- The default threshold is 10 MB.
- The maximum number of archive files is 3.
- The archive location will be the same as the original log location.

---

## Modifying default log settings

If you have the Log Management Tool and IBM Tivoli Directory Integrator installed, or you can set the default maximum log size threshold, the maximum number of log archives, and Log archive path values. For example, if you want all the log to keep only three archived logs, you can set the maximum log archives value to three for all the logs using the default settings.

Individual log settings override default log settings. The default log settings have no values by default.

Use the following procedures to modify log settings. The default log settings apply to all logs.

### Using the Web Administration Tool

To modify default log settings:

1. Click **Server administration** in the Web Administration navigation area and then click **Logs** in the expanded list.
2. Click **Modify log settings**.
3. Click **Default log settings**.
4. Under **Log size threshold (MB)** select the first radio button and enter the maximum log size in Megabytes. If you do not want to limit log size, select the **Unlimited** radio button instead.
5. Under **Maximum log archives**, select one of the following:
  - If you want to specify a maximum number of archived logs, select the radio button with an edit window next to it. Enter the maximum number of archives you want to save. One archived log is an earlier log that reached its size threshold.
  - If you do not want to archive logs, select **No archives**.
  - If you do not want to limit the number of archived logs, select **Unlimited**.
6. Under **Log archive path**, do one of the following:
  - If you want to specify the path where archives are kept, select the radio button with an edit window next to it and enter the desired path.
  - If you want to keep the archives in the directory where the log file is located, select the **Same directory as log file** radio button.
7. Click **Apply** to apply your changes and continue working with logs, or click **OK** to save your changes and to return to the IBM Tivoli Directory Server Web Administration Introduction panel. Click **Cancel** to return to the IBM Tivoli Directory Server Web Administration Introduction panel without saving any changes.

### Using the command line

Issue the command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where <filename> contains:

```
dn: cn=Default, cn=Log Management, cn=Configuration
changetype: modify
replace: ibm-slapdLogSizeThreshold
ibm-slapdLogSizeThreshold: <size threshold in MB>
-
replace: ibm-slapdLogMaxArchives
ibm-slapdLogMaxArchives: <number of log archives to save>
```

```
-
replace: ibm-slapdLogArchivePath
ibm-slapdLogArchivePath: <archived logs path>
```

---

## Modifying administration daemon error log settings

An administration daemon is a limited LDAP server that accepts extended operations to stop, start, and restart the LDAP server. The administration daemon error log (idsdiradm.log is the default file name) enables you to view status and errors encountered by the administration daemon.

To modify the administration daemon error log settings, use one of the following methods. Remember that individual log settings override the Default log settings.

### Using Web Administration Tool

1. Expand **Logs** in the navigation area, click **Modify log settings**.
2. Click **Admin daemon log**.
3. Enter the path and file name for the administration daemon error log. Ensure that the file exists on the LDAP server and that the path is valid. See “Default log paths” on page 267 for default log paths.

**Note:** If you specify a file that is not an acceptable file name (for example, invalid syntax or if the server does not have the rights to create and/or modify the file), the attempt fails with the following error: LDAP Server is unwilling to perform the operation.

4. Under **Log size threshold (MB)** select the first radio button and enter the maximum log size in Megabytes. If you do not want to limit log size, select the **Unlimited** radio button instead.
5. Under **Maximum log archives**, select one of the following:
  - If you want to specify a maximum number of archived logs, select the radio button with an edit window next to it. Enter the maximum number of archives you want to save. One archived log is an earlier log that reached its size threshold.
  - If you do not want to archive logs, select **No archives**.
  - If you do not want to limit the number of archived logs, select **Unlimited**.
6. Under **Log archive path**, do one of the following:
  - If you want to specify the path where archives are kept, select the radio button with an edit window next to it and enter the desired path.
  - If you want to keep the archives in the directory where the log file is located, select the **Same directory as log file** radio button.
7. Click **Apply** to apply your changes and continue working with logs, or click **OK** to save your changes and to return to the IBM Tivoli Directory Server Web Administration Introduction panel. Click **Cancel** to return to the IBM Tivoli Directory Server Web Administration Introduction panel without saving any changes.
8. You must stop the server for changes to take effect. See “Starting and stopping the server” on page 68. After stopping the server you must also stop and start the administration daemon locally to resynchronize the ports.
  - Issue the commands:

```
ibmdirctl -D <AdminDN> -w <AdminPW> -p <portnumber> stop

ibmdirctl -D <AdminDN> -w <AdminPW> admstop
```

```
idsdiradm
```

```
ibmdirctl -D <AdminDN> -w <AdminPW> -p <portnumber> start
```

- For Windows systems, you can also:
  - a. Go to **Control Panel->Administrative Tools->Services**.
  - b. Select **IBM Tivoli Directory Admin Daemon V6.0 – <InstanceName>**.
  - c. Do one of the following:
    - Click **Action -> Stop**.
    - Click **Stop the service**.
  - d. Select **IBM Tivoli Directory Admin Daemon V6.0 – <InstanceName>**.
  - e. Do one of the following:
    - Click **Action -> Start**.
    - Click **Start the service**.

Restart the server.

## Using the command line:

Issue the command:

```
idsldapmodify -D <adminDN> -w >adminPW> -i <filename>
```

where *<filename>* contains:

```
dn: cn=Admin, cn=Log Management, cn=Configuration
changetype: modify
replace: ibm-slapdLog
ibm-slapdLog: <newpathname>
-
replace: ibm-slapdLogSizeThreshold
ibm-slapdLogSizeThreshold: <size threshold in MB>
-
replace: ibm-slapdLogMaxArchives
ibm-slapdLogMaxArchives: <number of log archives to save>
-
replace: ibm-slapdLogArchivePath
ibm-slapdLogArchivePath: <archived logs path>
```

You must stop the server for changes to take effect. After stopping the server you must also stop and start the administration daemon locally to resynchronize the ports. Start the server.

```
ibmdirctl -D <AdminDN> -w <AdminPW> -p <portnumber> stop
```

```
ibmdirctl -D <AdminDN> -w <AdminPW> admstop
```

```
idsdiradm
```

```
ibmdirctl -D <AdminDN> -w <AdminPW> -p <portnumber> start
```

---

## Enabling the administration daemon audit log and modifying administration audit log settings

Audit logging is used to improve the security of the directory server. The directory administrator and administrative group members can use the records stored in the audit log to check for suspicious patterns of activity in an attempt to detect security violations. If security is violated, the administration daemon audit log (adminaudit.log is the default file name) can be used to determine how and when the problem occurred and perhaps the amount of damage done.

**Note:** Failed connection attempts are audited only if they fail after reaching the LDAP server. Connections that fail in the SSL layer, network, or operating system layer are not audited.

**Note:** Members of the administrative group can view the administration daemon audit log and settings but not modify them. Only the administrator is enabled to access, change or clear the administration daemon audit log files.

To modify the administration audit log settings, use one of the following methods. Remember that individual log settings override the Default log settings.

**Note:** The administration daemon audit log audits binds, unbinds, searches and extended operations.

## Using Web Administration Tool

1. Expand **Logs** in the navigation area, click **Modify log settings**.
2. Click **Admin daemon audit log**.
3. Select **Enable admin daemon audit logging** to use the audit log utility with the administration daemon.

**Note:** The default setting is enabled. You only need to select the check box, if you have previously disabled the administration daemon audit log.

4. Enter the path and file name for the administration daemon audit log. Ensure that the file exists on the ldap server and that the path is valid. See "Default log paths" on page 267 for default log paths.

**Note:** If you specify a file that is not an acceptable file name (for example, invalid syntax or if the server does not have the rights to create and/or modify the file), the attempt fails with the following error: LDAP Server is unwilling to perform the operation.

5. Under **Log size threshold (MB)** select the first radio button and enter the maximum log size in Megabytes. If you do not want to limit log size, select the **Unlimited** radio button instead.
6. Under **Maximum log archives**, select one of the following:
  - If you want to specify a maximum number of archived logs, select the radio button with an edit window next to it. Enter the maximum number of archives you want to save. One archived log is an earlier log that reached its size threshold.
  - If you do not want to archive logs, select **No archives**.
  - If you do not want to limit the number of archived logs, select **Unlimited**.
7. Under **Log archive path**, do one of the following:
  - If you want to specify the path where archives are kept, select the radio button with an edit window next to it and enter the desired path.
  - If you want to keep the archives in the directory where the log file is located, select the **Same directory as log file** radio button.
8. Click **Apply** to apply your changes and continue working with logs, or click **OK** to save your changes and to return to the IBM Tivoli Directory Server Web Administration Introduction panel. Click **Cancel** to return to the IBM Tivoli Directory Server Web Administration Introduction panel without saving any changes.

9. You must stop the server for changes to take effect. See “Starting and stopping the server” on page 68. After stopping the server you must also stop and start the administration daemon locally to resynchronize the ports.

- Issue the commands:

```
ibmdirctl -D <AdminDN> -w <Adminpw> admstop
```

```
idsdiradm
```

- For Windows systems, you can also:
  - a. Through the Control Panel, open the Services window.
  - b. Click **Directory Admin Daemon**.
  - c. Click **Action -> Stop**.
  - d. Click **Directory Admin Daemon**.
  - e. Click **Action -> Start**.

Restart the server.

## Using the command line:

Issue the command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where <filename> contains:

```
dn: cn=Admin Audit, cn=Log Management, cn=Configuration
```

```
changetype: modify
```

```
replace: ibm-audit
```

```
ibm-audit: true
```

```
-
```

```
replace: ibm-slapdLog
```

```
ibm-slapdLog: <newpathname>
```

```
-
```

```
replace: ibm-slapdLogSizeThreshold
```

```
ibm-slapdLogSizeThreshold: <size threshold in MB>
```

```
-
```

```
replace: ibm-slapdLogMaxArchives
```

```
ibm-slapdLogMaxArchives: <number of log archives to save>
```

```
-
```

```
replace: ibm-slapdLogArchivePath
```

```
ibm-slapdLogArchivePath: <archived logs path>
```

You must stop and restart the server for changes to take effect. After stopping the Admin Daemon you must issue a dynamic update request to the LDAP server.

```
ibmdirctl -D <AdminDN> -w <adminPW> admstop
```

```
idsdiradm
```

```
idsldapexop -D <AdminDN> -w <adminPW> -op readconfig -scope
entry "cn=Admin Audit,cn=Log Management,c=Configuration"
```

---

## Disabling the administration daemon audit log

To disable audit logging:

### Using Web Administration:

1. Expand **Logs** in the navigation area, click **Modify log settings**.
2. Click **Admin daemon audit log**.
3. Deselect **Enable admin daemon audit logging**.

- Click **Apply** to apply your changes and continue working with logs, or click **OK** to save your changes and to return to the IBM Tivoli Directory Server Web Administration Introduction panel. Click **Cancel** to return to the IBM Tivoli Directory Server Web Administration Introduction panel without saving any changes.

## Using the command line:

Issue the command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where <filename> contains:

```
dn: cn=Admin Audit, cn=Log Management, cn=Configuration
changetype: modify
replace: ibm-audit
ibm-audit: false
```

**Note:** You need to restart the Admin Daemon only.

---

## Enabling the audit log and modifying audit log settings

Audit logging is used to improve the security of the directory server. A default audit plug-in is provided with the server. Depending on the audit configuration parameters, this plug-in might log an audit entry in the default or specified audit log for each LDAP operation the server processed. The administrator can use the activities stored in the audit log to check for suspicious patterns of activity in an attempt to detect security violations. If security is violated, the audit log can be used to determine how and when the problem occurred and perhaps the amount of damage done. This information is very useful, both for recovery from the violation and, possibly, in the development of better security measures to prevent future problems. You can also write your own audit plug-ins to either replace, or add more processing to, the default audit plug-in. For more information about plug-ins, see the *IBM Tivoli Directory Server Plug-ins Reference Version 6.0*.

**Note:** Failed connection attempts are audited only if they fail after reaching the LDAP server. Connections that fail in the SSL layer, network, or operating system layer are not audited.

The audit log displays log entries chronologically. Each non-message entry contains a general information header followed by operation-specific data. For example,

```
2000-03-23-16:01:01.345-06:00--V3 Bind--bindDN:cn=root
--client:9.1.2.3:12345--
ConnectionID:12--received:2000-03-23-16:01:01.330-06:00
--success
name: cn=root
authenticationChoice: simple
```

If the audit version is version 2 the header contains "AuditV2--".

```
AuditV2--2003-07-22-09:39:54.421-06:00DST--V3 Bind--bindDN: cn=root--client: 127
.0.0.1:8196--connectionID: 3--received: 2003-07-22-09:39:54.421-06:00DST--Success
```

If the audit version is version 3 the header contains "AuditV3--".

```
AuditV3--2003-07-22-09:39:54.421-06:00DST--V3 Bind--bindDN: cn=root--client: 127
.0.0.1:8196--connectionID: 3--received: 2003-07-22-09:39:54.421-06:00DST--Success
UniqueID:
```

**Note:** For an operation, one of the following is printed:



- Unknown
- Bind
- Unbind
- Search
- Add
- Modify
- Delete
- ModifyDN
- event notification: registration
- event notification: unregister
- extended operation
- Compare

The header is in the following format:

**Timestamp 1 "--"**

The local time the entry is logged, that is, the time the request was processed. The timestamp is in the format YYYY-MM-DD-HH:MM:SS.mmm=(or-)HH:MM. The =(or-)HH:MM is UTC offset. mmm is milliseconds.

**Version number+[SSL|TLS]+[unauthenticated or anonymous] Operation "--"**

Shows the LDAP request that was received and processed. Version number is either V2 or V3. **SSL** displays only when SSL was used for the connection. **TLS** displays only when TLS is used for the connection. **unauthenticated** or **anonymous** displays to indicate whether the request was from an unauthenticated or anonymous client. Neither unauthenticated or anonymous display if the request is from an authenticated client.

**bindDN:**

Shows the bind DN. For V3 unauthenticated or anonymous requests, this field is <\*CN=NULLDN\*>.

**client:Client IP address:Port number "--"**

Shows the client IP address and port number.

**ConnectionID: xxxx "--"**

Is used to group all the entries received in the same connection, meaning between the bind and unbind, together.

**received: Timestamp 2 "--"**

Is the local time when the request was received, or to be more specific, the beginning time when the request was processed. Its format is the same as Timestamp 1.

**Result or Status string**

Shows the result or status of the LDAP operation. For the result string, the textual form of the LDAP resultCode is logged, for example, success or operationsError, instead of 0 or 1.

**UniqueID**

The uniqueID is the unique request ID to store in the control. The clientIP is the client's original IP to store in the control. If critical is true the criticality of the control will be set to true; if false the criticality will be set to false.

Operation-specific data follows the header and displays operation-specific data, for example,

- Bind operations  
name: Y249bWFuYWdlcg0K  
authenticationChoice: simple
- Add operations  
entry: cn=Jim Brown, ou=sales,o=ibm\_us,c=us  
attributes: objectclass, cn, sn, telephonenumber
- Delete operations  
entry: cn=Jim Brown, ou=sales,o=ibm\_us,c=us
- Modify operations  
object: cn=Jim Brown, ou=sales,o=ibm\_us,c=us  
add: mail  
delete: telephonenumber

By default the audit log is disabled.

**Note:** Members of the administrative group can view the audit log and settings but not modify them. Only the administrator is enabled to access, change or clear the audit log files.

To enable audit logging and modify logging settings, use one of the following methods. Remember that individual log settings override the Default log settings.

## Using Web Administration

1. Expand **Logs** in the navigation area, click **Modify log settings**.
2. Click **Audit log**.
3. Select **Enable audit logging** to use the audit log utility.
4. Enter the **Path and file name** for the audit log. The audit log can also be directed to something other than a file, for example, a line printer. Ensure that the file exists on the ldap server and that the path is valid. See “Default log paths” on page 267 for default log paths.

**Note:** If you specify a file that is not an acceptable file name (for example, invalid syntax or if the server does not have the rights to create and/or modify the file), the attempt fails with the following error: LDAP Server is unwilling to perform the operation.

5. Under **Log size threshold (MB)** select the first radio button and enter the maximum log size in Megabytes. If you do not want to limit log size, select the **Unlimited** radio button instead.
6. Under **Maximum log archives**, select one of the following:
  - If you want to specify a maximum number of archived logs, select the radio button with an edit window next to it. Enter the maximum number of archives you want to save. One archived log is an earlier log that reached its size threshold.
  - If you do not want to archive logs, select **No archives**.
  - If you do not want to limit the number of archived logs, select **Unlimited**.
7. Under **Audit version**, select the audit version you want to use. Version 1 maintains previous audit logging capabilities for any applications that parse the audit log. Version 2 enables you to log extended operations, however, you might need to modify existing applications that parse the audit log. Version 3, the default value, also writes out a unique ID, if the server generates one for

the request. The unique ID only appears on the proxy server and is printed between the header information and any control data.

8. Under **Audit log level**, do one of the following:
  - If you want to log only failed attempts, select the **Only failed attempts** radio button.
  - If you want to log all attempts, select the **All attempts** radio button.
9. Under **Log archive path**, do one of the following:
  - If you want to specify the path where archives are kept, select the radio button with an edit window next to it and enter the desired path.
  - If you want to keep the archives in the directory where the log file is located, select the **Same directory as log file** radio button.
10. Select the operations you wish to log. Consult the field help for additional information about the various operations you can log.
  - **Bind** - records connections to the server
  - **Unbind** - records disconnections from the server
  - **Search** - records LDAP search operations performed by any client
  - **Add** - records additions to LDAP
  - **Modify** - records modifications to LDAP
  - **Delete** - records deletions from LDAP
  - **Compare** - records compare operations
  - **Modify RDN** - records modifications made to RDNs
  - **Event notification** - records event notifications
  - **Extended operations**- records extended operations performed against the server
  - **Group values sent on group control** - records the groups defined in the group control.
  - **Attributes sent on group evaluation extended operation** - records attributes sent with the group evaluation extended operation.
11. Click **Apply** to apply your changes and continue working with logs, or click **OK** to save your changes and to return to the IBM Tivoli Directory Server Web Administration Introduction panel. Click **Cancel** to return to the IBM Tivoli Directory Server Web Administration Introduction panel without saving any changes.

## Using the command line:

Issue the command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where <filename> contains:

```
dn: cn=Audit, cn=Log Management, cn=Configuration
changetype: modify
replace: ibm-audit
ibm-audit: true
-
replace: ibm-slapdLog
ibm-slapdLog: <newpathname>
-
replace: ibm-slapdLogSizeThreshold
ibm-slapdLogSizeThreshold: <size threshold in MB>
-
replace: ibm-slapdLogMaxArchives
ibm-slapdLogMaxArchives: <number of log archives to save>
```

```

-
replace: ibm-slapdLogArchivePath
ibm-slapdLogArchivePath: <archived logs path>
-
replace: ibm-auditadd
ibm-auditadd: {TRUE|FALSE}
#select TRUE to enable, FALSE to disable
-
replace: ibm-auditbind
ibm-auditbind: {TRUE|FALSE}
#select TRUE to enable, FALSE to disable
-
replace: ibm-auditdelete
ibm-auditdelete: {TRUE|FALSE}
#select TRUE to enable, FALSE to disable
-
replace: ibm-auditextopevent
ibm-auditextopevent: {TRUE|FALSE}
#select TRUE to enable, FALSE to disable
-
replace: ibm-auditfailedoonly
ibm-auditfailedoonly: {TRUE|FALSE}
#select TRUE to enable, FALSE to disable
-
replace: ibm-auditmodify
ibm-auditmodify: {TRUE|FALSE}
#select TRUE to enable, FALSE to disable
-
replace: ibm-auditmodifydn
ibm-auditmodifydn: {TRUE|FALSE}
#select TRUE to enable, FALSE to disable
-
replace: ibm-auditsearch
ibm-auditsearch: {TRUE|FALSE}
#select TRUE to enable, FALSE to disable
-
replace: ibm-auditunbind
ibm-auditunbind: {TRUE|FALSE}
#select TRUE to enable, FALSE to disable
-
replace: ibm-auditversion
ibm-auditversion: {1|2|3}
#select 2 or 3, if you are enabling audit of additional information on controls
-
replace: ibm-auditExtOp
ibm-auditExtOp: {TRUE|FALSE}
#select TRUE to enable, FALSE to disable
-
-
replace: ibm-auditCompare
ibm-auditCompare: {TRUE|FALSE}
#select TRUE to enable, FALSE to disable
-
replace: ibm-auditGroupsOnGroupControl
ibm-auditGroupsOnGroupControl: {TRUE|FALSE}
#select TRUE to enable, FALSE to disable
-
replace: ibm-auditAttributesOnGroupEvalOp
ibm-auditAttributesOnGroupEvalOp: {TRUE|FALSE}
#select TRUE to enable, FALSE to disable

```

---

## Disabling the audit log

To disable audit logging use one of the following methods:

## Using Web Administration:

1. Expand **Logs** in the navigation area, click **Modify log settings**, click **Audit log**.
2. Deselect **Enable audit logging**.
3. Click **Apply** to apply your changes and continue working with logs, or click **OK** to save your changes and to return to the IBM Tivoli Directory Server Web Administration Introduction panel. Click **Cancel** to return to the IBM Tivoli Directory Server Web Administration Introduction panel without saving any changes.

## Using the command line:

Issue the command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where <filename> contains:

```
dn: cn=Audit, cn=Log Management, cn=Configuration
changetype: modify
replace: ibm-audit
ibm-audit: false
```

---

## Modifying bulkload error log settings

Bulkload is used for loading entries. The bulkload log allows you to view status and errors related to bulkload. See “idsbulkload, bulkload” on page 423.

To modify the bulkload log settings, use one of the following methods. Remember that individual log settings override the Default log settings.

## Using Web Administration

1. Expand **Logs** in the navigation area, click **Modify log settings**.
2. Click **Bulkload log**.
3. Enter the path and file name for the error log. Ensure that the file exists on the ldap server and that the path is valid. See “Default log paths” on page 267 for default log paths.

**Note:** If you specify a file that is not an acceptable file name (for example, invalid syntax or if the server does not have the rights to create and/or modify the file), the attempt fails with the following error: LDAP Server is unwilling to perform the operation.

4. Under **Log size threshold (MB)** select the first radio button and enter the maximum log size in Megabytes. If you do not want to limit log size, select the **Unlimited** radio button instead.
5. Under **Maximum log archives**, select one of the following:
  - If you want to specify a maximum number of archived logs, select the radio button with an edit window next to it. Enter the maximum number of archives you want to save. One archived log is an earlier log that reached its size threshold.
  - If you do not want to archive logs, select **No archives**.
  - If you do not want to limit the number of archived logs, select **Unlimited**.
6. Under **Log archive path**, do one of the following:
  - If you want to specify the path where archives are kept, select the radio button with an edit window next to it and enter the desired path.

- If you want to keep the archives in the directory where the log file is located, select the **Same directory as log file** radio button.
7. Click **Apply** to apply your changes and continue working with logs, or click **OK** to save your changes and to return to the IBM Tivoli Directory Server Web Administration Introduction panel. Click **Cancel** to return to the IBM Tivoli Directory Server Web Administration Introduction panel without saving any changes.

## Using the command line:

Issue the command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where <filename> contains:

```
cn=Bulkload, cn=Log Management, cn=Configuration
changetype: modify
replace: ibm-slapdLog
ibm-slapdLog: <newpathname>
-
replace: ibm-slapdLogSizeThreshold
ibm-slapdLogSizeThreshold: <size threshold in MB>
-
replace: ibm-slapdLogMaxArchives
ibm-slapdLogMaxArchives: <number of log archives to save>
-
replace: ibm-slapdLogArchivePath
ibm-slapdLogArchivePath: <archived logs path>
-
```

To update the settings dynamically, issue the following **idsldapexop** command:

```
idsldapexop -D <adminDN> -w <adminPW> -op readconfig -scope single
"cn=Bulkload, cn=Log Management, cn=Configuration" ibm-slapdLog
```

The **idsldapexop** command updates only those attributes that are dynamic. For other changes to take effect you must stop and restart the server. See “Dynamically-changed attributes” on page 578 for a list of the attributes that can be updated dynamically.

---

## Modifying configuration tools log settings

The configuration tools log enables you to view status and error messages related to the configuration tools, such as **idscfgdb**, **idsucfgdb**, **idscfgchlog**, **idsucfgchlog**, **idscfgsuf**, **idsucfgsuf**, **idsdnpw**, **idsxcfg** .

To modify the configuration tools log settings, use one of the following methods. Remember that individual log settings override the Default log settings.

## Using Web Administration

1. Expand **Logs** in the navigation area, click **Modify log settings**.
2. Click **Configuration tools log**.
3. Enter the path and file name for the error log. Ensure that the file exists on the ldap server and that the path is valid. See “Default log paths” on page 267 for default log paths.

**Note:** If you specify a file that is not an acceptable file name (for example, invalid syntax or if the server does not have the rights to create and/or

modify the file), the attempt fails with the following error: LDAP Server is unwilling to perform the operation.

4. Under **Log size threshold (MB)** select the first radio button and enter the maximum log size in Megabytes. If you do not want to limit log size, select the **Unlimited** radio button instead.
5. Under **Maximum log archives**, select one of the following:
  - If you want to specify a maximum number of archived logs, select the radio button with an edit window next to it. Enter the maximum number of archives you want to save. One archived log is an earlier log that reached its size threshold.
  - If you do not want to archive logs, select **No archives**.
  - If you do not want to limit the number of archived logs, select **Unlimited**.
6. Under **Log archive path**, do one of the following:
  - If you want to specify the path where archives are kept, select the radio button with an edit window next to it and enter the desired path.
  - If you want to keep the archives in the directory where the log file is located, select the **Same directory as log file** radio button.
7. Click **Apply** to apply your changes and continue working with logs, or click **OK** to save your changes and to return to the IBM Tivoli Directory Server Web Administration Introduction panel. Click **Cancel** to return to the IBM Tivoli Directory Server Web Administration Introduction panel without saving any changes.

## Using the command line

Issue the command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where <filename> contains:

```
dn: cn=Tools, cn=Log Management, cn=Configuration
changetype: modify
replace: ibm-slapdLog
ibm-slapdLog: <newpathname>
-
replace: ibm-slapdLogSizeThreshold
ibm-slapdLogSizeThreshold: <size threshold in MB>
-
replace: ibm-slapdLogMaxArchives
ibm-slapdLogMaxArchives: <number of log archives to save>
-
replace: ibm-slapdLogArchivePath
ibm-slapdLogArchivePath: <archived logs path>
```

---

## Modifying DB2 error log settings

The DB2 error log (db2cli.log is the default file name) records database errors that occur as a result of LDAP operations.

To modify the DB2 log settings, use one of the following methods. Remember that individual log settings override the Default log settings.

## Using Web Administration

1. Expand **Server administration** in the navigation area, click **Logs**, click **Modify log settings**, click **DB2 log**.

2. Enter the path and file name for the DB2 log. Ensure that the path is valid. If the file does not exist, it is created. The error log can also be directed to something other than a file, for example, a line printer. See “Default log paths” on page 267 for default log paths.

**Note:** If you specify a file that is not an acceptable file name (for example, invalid syntax or if the server does not have the rights to create and/or modify the file), the attempt fails with the following error: LDAP Server is unwilling to perform the operation.

3. Under **Log size threshold (MB)** select the first radio button and enter the maximum log size in Megabytes. If you do not want to limit log size, select the **Unlimited** radio button instead.
4. Under **Maximum log archives**, select one of the following:
  - If you want to specify a maximum number of archived logs, select the radio button with an edit window next to it. Enter the maximum number of archives you want to save. One archived log is an earlier log that reached its size threshold.
  - If you do not want to archive logs, select **No archives**.
  - If you do not want to limit the number of archived logs, select **Unlimited**.
5. Under **Log archive path**, do one of the following:
  - If you want to specify the path where archives are kept, select the radio button with an edit window next to it and enter the desired path.
  - If you want to keep the archives in the directory where the log file is located, select the **Same directory as log file** radio button.
6. Click **Apply** to apply your changes and continue working with logs, or click **OK** to save your changes and to return to the IBM Tivoli Directory Server Web Administration Introduction panel. Click **Cancel** to return to the IBM Tivoli Directory Server Web Administration Introduction panel without saving any changes.

## Using the command line:

Issue the command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where <filename> contains:

```
dn: cn=DB2CLI, cn=Log Management, cn=Configuration
changetype: modify
replace: ibm-slapdLog
ibm-slapdLog: <newpathname>
-
replace: ibm-slapdLogSizeThreshold
ibm-slapdLogSizeThreshold: <size threshold in MB>
-
replace: ibm-slapdLogMaxArchives
ibm-slapdLogMaxArchives: <number of log archives to save>
-
replace: ibm-slapdLogArchivePath
ibm-slapdLogArchivePath: <archived logs path>
-
```

To update the settings dynamically, issue the following **idsldapexop** command:

```
idsldapexop -D <adminDN> -w <adminPW> -op readconfig -scope single
"cn=DB2CLI, cn=Log Management, cn=Configuration" ibm-slapdLog
```



The `idsldapexop` command updates only those attributes that are dynamic. For other changes to take effect you must stop and restart the server. See “Dynamically-changed attributes” on page 578 for a list of the attributes that can be updated dynamically.

---

## Modifying lost and found log settings

The lost and found log (`LostAndFound.log` is the default file name) records errors that occur as a result of a replication conflict.

To modify the lost and found log settings, use one of the following methods. Remember that individual log settings override the Default log settings.

### Using Web Administration

1. Expand **Logs** in the navigation area, click **Modify log settings**.
2. Click **Lost and found log**.
3. Enter the path and file name for the error log. Ensure that the file exists on the ldap server and that the path is valid. See “Default log paths” on page 267 for default log paths.

**Note:** If you specify a file that is not an acceptable file name (for example, invalid syntax or if the server does not have the rights to create and/or modify the file), the attempt fails with the following error: LDAP Server is unwilling to perform the operation.

4. Under **Log size threshold (MB)** select the first radio button and enter the maximum log size in Megabytes. If you do not want to limit log size, select the **Unlimited** radio button instead.
5. Under **Maximum log archives**, select one of the following:
  - If you want to specify a maximum number of archived logs, select the radio button with an edit window next to it. Enter the maximum number of archives you want to save. One archived log is an earlier log that reached its size threshold.
  - If you do not want to archive logs, select **No archives**.
  - If you do not want to limit the number of archived logs, select **Unlimited**.
6. Under **Log archive path**, do one of the following:
  - If you want to specify the path where archives are kept, select the radio button with an edit window next to it and enter the desired path.
  - If you want to keep the archives in the directory where the log file is located, select the **Same directory as log file** radio button.
7. Click **Apply** to apply your changes and continue working with logs, or click **OK** to save your changes and to return to the IBM Tivoli Directory Server Web Administration Introduction panel. Click **Cancel** to return to the IBM Tivoli Directory Server Web Administration Introduction panel without saving any changes.

### Using the command line

Issue the command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where <filename> contains:

```

dn: cn=Replication, cn=Log Management, cn=Configuration
changetype: modify
replace: ibm-slapdLog
ibm-slapdLog: <newpathname>
-
replace: ibm-slapdLogSizeThreshold
ibm-slapdLogSizeThreshold: <size threshold in MB>
-
replace: ibm-slapdLogMaxArchives
ibm-slapdLogMaxArchives: <number of log archives to save>
-
replace: ibm-slapdLogArchivePath
ibm-slapdLogArchivePath: <archived logs path>

```

---

## Modifying the server error log

The error log, **ibmslapd.log** (this is the default file name), is enabled by default. The error log enables you to view status and error messages related to the server.

To modify the error log settings, use one of the following methods. Remember that individual log settings override the Default log settings.

### Using Web Administration

1. Expand **Server administration** in the navigation area, click **Logs**, click **Modify log settings**.
2. Click **Server error log**.
3. Enter the path and file name for the error log. Ensure that the path is valid. If the file does not exist, it is created. The error log can also be directed to something other than a file, for example, a line printer. See “Default log paths” on page 267 for default log paths.

**Note:** If you specify a file that is not an acceptable file name (for example, invalid syntax or if the server does not have the rights to create and/or modify the file), the attempt fails with the following error: LDAP Server is unwilling to perform the operation.

4. Under **Log size threshold (MB)** select the first radio button and enter the maximum log size in Megabytes. If you do not want to limit log size, select the **Unlimited** radio button instead.
5. Under **Maximum log archives**, select one of the following:
  - If you want to specify a maximum number of archived logs, select the radio button with an edit window next to it. Enter the maximum number of archives you want to save. One archived log is an earlier log that reached its size threshold.
  - If you do not want to archive logs, select **No archives**.
  - If you do not want to limit the number of archived logs, select **Unlimited**.
6. Under **Log archive path**, do one of the following:
  - If you want to specify the path where archives are kept, select the radio button with an edit window next to it and enter the desired path.
  - If you want to keep the archives in the directory where the log file is located, select the **Same directory as log file** radio button.
7. Select either Low, Medium, or High for the level of error logging.
  - Low logs the least amount of error information, for example,

```

Oct 06 10:33:02 2004 GLPSRV009I IBM Tivoli Directory (SSL), Version 6.0
Server started.

```

- Medium logs a medium amount of error information, for example,
 

```
Oct 06 10:35:41 2004 GLPCOM024I The extended Operation plugin is successfully
loaded from libloga.dll.
Oct 06 10:35:41 2004 GLPCOM003I Non-SSL port initialized to 389.
Oct 06 10:35:44 2004 GLPSRV009I IBM Tivoli Directory (SSL), Version 6.0
Server started.
```
  - High logs the most amount of error information, for example
 

```
Oct 06 10:37:48 2004 GLPSRV047W Anonymous binds will be allowed.
Oct 06 10:37:48 2004 GLPCOM024I The extended Operation plugin is successfully
loaded from libloga.dll.
Oct 06 10:37:48 2004 GLPSRV003I Configuration file successfully read.
Oct 06 10:37:48 2004 GLPCOM003I Non-SSL port initialized to 389.
Oct 06 10:37:51 2004 GLPSRV009I IBM Tivoli Directory (SSL), Version 6.0
Server started.
```
8. Click **Apply** to apply your changes and continue working with logs, or click **OK** to save your changes and to return to the IBM Tivoli Directory Server Web Administration Introduction panel. Click **Cancel** to return to the IBM Tivoli Directory Server Web Administration Introduction panel without saving any changes.

## Using the command line

Issue the command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where *<filename>* contains:

```
dn: cn=ibmslapd, cn=Log Management, cn=Configuration
changetype: modify
replace: ibm-slapdLog
ibm-slapdLog: <newpathname>
-
replace: ibm-slapdLogSizeThreshold
ibm-slapdLogSizeThreshold: <size threshold in MB>
-
replace: ibm-slapdLogMaxArchives
ibm-slapdLogMaxArchives: <number of log archives to save>
-
replace: ibm-slapdLogArchivePath
ibm-slapdLogArchivePath: <archived logs path>
-
replace: ibm-slapdLogOptions
ibm-slapdLogOptions: {l | m | h}
```

To update the settings dynamically, issue the following **idsldapexop** command:

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

The **idsldapexop** command updates only those attributes that are dynamic. For other changes to take effect you must stop and restart the server. See “Dynamically-changed attributes” on page 578 for a list of the attributes that can be updated dynamically.

---

## Viewing logs

The following sections show you how to view the IBM Tivoli Directory Server logs.

### View logs using Web Administration

To view a log using the Web Administration Tool, do the following:

1. Click **Server administration** in the Web Administration navigation area and then click **Logs** in the expanded list. Click **View log**.
2. Select the log you want to view from the **Select log** drop-down menu; for example, **Lost and Found log**
3. You can:
  - Use the navigation arrows at the bottom of the panel allow you to go to the **Next** page or to the **Previous** page.
  - Select a specific page from the edit menu, for example **Page 6 of 16**, and click **Go** to display that page of the error log.
  - Click **Refresh** to update the entries in the log.
  - Click **Clear log** to delete all entries in the log.

**Note:** Admin Group members cannot clear the Audit logs.

4. Click **Close** to return to the IBM Tivoli Directory Server Web Administration Introduction panel.

## View logs using the command line

Use the following procedures to view logs using the command line.

### Viewing the Admin daemon error log

To view the administration daemon error log in the default location, issue the following command:

On a UNIX operating system:

```
more <instance base directory>/idsslapd-<instance name>/logs/idsdiradm.log
```

Where:

- *instance base directory* is the home directory of the directory server instance owner, or the directory you specified when you created the directory server instance.
- *instance name* is the name of the directory server instance.

On a Windows operating system:

```
more <drive>\idsslapd-<instance name>\logs
```

Where *drive* is the drive you specified when you created a directory server instance, and *instance name* is the name of the directory server instance.

Do the following to view the Web Administration error log from a system with the IBM Tivoli Directory Server client:

```
idsldapexop -D <adminDN> -w <adminPW> -op readlog -log idsdiradm -lines all
```

Do the following to clear the Web Administration error log:

```
ldapexop -D <adminDN> -w <adminPW> -op clearlog -log idsdiradm
```

### Viewing the Admin daemon audit log settings

To view the administration daemon audit log in the default location, issue the following command:

On a UNIX operating system:

```
more <instance base directory>/idsslapd-<instance name>/logs/adminaudit.log
```

Where:

- *instance base directory* is the home directory of the directory server instance owner, or the directory you specified when you created the directory server instance.
- *instance name* is the name of the directory server instance.

On a Windows operating system:

```
more <drive>\idsslapd-<instance name>\logs\adminaudit.log
```

Where *drive* is the drive you specified when you created a directory server instance, and *instance name* is the name of the directory server instance.

Do the following to view the administration daemon log from a system with the IBM Tivoli Directory Server client:

```
idsldapexop -D <adminDN> -w <adminPW> -op readlog -log adminAudit -lines all
```

Do the following to clear the administration daemon log:

```
idsldapexop -D <adminDN> -w <adminPW> -op clearlog -log adminAudit
```

## Viewing the audit log

To view the audit log in the default location, issue the following command:

On a UNIX operating system:

```
more <instance base directory>/idsslapd-<instance name>/logs/audit.log
```

Where:

- *instance base directory* is the home directory of the directory server instance owner, or the directory you specified when you created the directory server instance.
- *instance name* is the name of the directory server instance.

On a Windows operating system:

```
more <drive>\idsslapd-<instance name>\logs\audit.log
```

Where *drive* is the drive you specified when you created a directory server instance, and *instance name* is the name of the directory server instance.

Do the following to view the audit log from a system with the IBM Tivoli Directory Server client:

```
idsldapexop -D <adminDN> -w <adminPW> -op readlog -log audit -lines all
```

Do the following to clear the audit log:

```
idsldapexop -D <adminDN> -w <adminPW> -op clearlog -log audit
```

## Viewing the Bulkload log

To view the bulkload log in the default location, issue the following command:

On a UNIX operating system:

```
more <instance base directory>/idsslapd-<instance name>/logs/bulkload.log
```

Where:

- *instance base directory* is the home directory of the directory server instance owner, or the directory you specified when you created the directory server instance.
- *instance name* is the name of the directory server instance.

On a Windows operating system:

```
more <drive>\idsslapd-<instance name>\logs\bulkload.log
```

Where *drive* is the drive you specified when you created a directory server instance, and *instance name* is the name of the directory server instance.

Do the following to view the bulkload error log from a system with the IBM Directory Server client:

```
idsldapexop -D <adminDN> -w <adminPW> -op readlog -log bulkload -lines all
```

Do the following to clear the bulkload error log:

```
idsldapexop -D <adminDN> -w <adminPW> -op clearlog -log bulkload
```

## Viewing the Configuration tools log

To view the Configuration tools log in the default location, issue the following command:

On a UNIX operating system:

```
more <instance base directory>/idsslapd-<instance name>/logs/idstools.log
```

Where:

- *instance base directory* is the home directory of the directory server instance owner, or the directory you specified when you created the directory server instance.
- *instance name* is the name of the directory server instance.

On a Windows operating system:

```
more <drive>\idsslapd-<instance name>\logs\idstools.log
```

Where *drive* is the drive you specified when you created a directory server instance, and *instance name* is the name of the directory server instance.

Do the following to view the Configuration tools log from a system with the IBM Directory Server client:

```
idsldapexop -D <adminDN> -w <adminPW> -op readlog -log config -lines all
```

Do the following to clear the Configuration tools log:

```
idsldapexop -D <adminDN> -w <adminPW> -op clearlog -log config
```

## Viewing the DB2 log

To view the DB2 log in the default location, issue the following command:

On a UNIX operating system:

```
more <instance base directory>/idsslapd-<instance name>/logs/db2cli.log
```

Where:

- *instance base directory* is the home directory of the directory server instance owner, or the directory you specified when you created the directory server instance.
- *instance name* is the name of the directory server instance.

On a Windows operating system:

```
more <drive>\idsslapd-<instance name>\logs\db2cli.log
```

Where *drive* is the drive you specified when you created a directory server instance, and *instance name* is the name of the directory server instance.

Do the following to view the DB2 error log from a system with the IBM Directory Server client:

```
idsldapexop -D <adminDN> -w <adminPW> -op readlog -log cli -lines all
```

Do the following to clear the DB2 error log:

```
idsldapexop -D <adminDN> -w <adminPW> -op clearlog -log cli
```

## Viewing the Lost and found error log

To view the Lost and Found log in the default location, issue the following command:

On a UNIX operating system:

```
more <instance base directory>/idsslapd-<instance name>/logs
/LostAndFound.log
```

Where:

- *instance base directory* is the home directory of the directory server instance owner, or the directory you specified when you created the directory server instance.
- *instance name* is the name of the directory server instance.

On a Windows operating system:

```
more <drive>\idsslapd-<instance name>\logs\LostAndFound.log
```

Where *drive* is the drive you specified when you created a directory server instance, and *instance name* is the name of the directory server instance.

Do the following to view the Lost and found error log from a system with the IBM Directory Server client:

```
idsldapexop -D <adminDN> -w <adminPW> -op readlog -log LostAndFound -lines all
```

Do the following to clear the Lost and found error log:

```
idsldapexop -D <adminDN> -w <adminPW> -op clearlog -log LostAndFound
```

## Viewing the Server error log

To view the Configuration tools log in the default location, issue the following command

On a UNIX operating system:

```
more <instance base directory>/idsslapd-<instance name>/logs/ibmslapd.log
```

Where:

- *instance base directory* is the home directory of the directory server instance owner, or the directory you specified when you created the directory server instance.
- *instance name* is the name of the directory server instance.

On a Windows operating system:

```
more <drive>\idsslapd-<instance name>\logs\ibmslapd.log
```

Where *drive* is the drive you specified when you created a directory server instance, and *instance name* is the name of the directory server instance.

Do the following to view the error log from a system with the IBM Directory Server client:

```
idsldapexop -D <adminDN> -w <adminPW> -op readlog -log slapd -lines all
```

Do the following to clear the error log:

```
idsldapexop -D <adminDN> -w <adminPW> -op clearlog -log slapd
```



---

## Part 3. Directory Management



---

## Chapter 16. Working with directory entries

Expand the **Directory management** category in the navigation area of the Web Administration Tool. All the directory entry tasks that you want to perform can be accessed by selecting **Manage entries**. Two short cuts have been added to the navigation area for the specific tasks of adding an entry and finding (searching for) entries

You can perform the following operations with directory entries:

- Browse the directory tree
- Add or remove an entry
- Add or remove an auxiliary object class to an entry
- Edit the attributes of an entry
- Copy an entry
- Manage members
- Manage memberships
- Edit ACLs
- Search for entries

---

### Browsing the tree

If you have not done so already, expand the **Directory management** category in the navigation area, then click **Manage entries**. The Manage entries table displays the following column information:

**Select** Select the radio button next to the name of an attribute you want to view, edit, copy or delete.

**Expand**

Indicates an expandable entry. An expandable entry is an entry that has child entries.

**Note:** It is possible that even though the + sign is present, you still might not see any child entries, as ACLs do not permit a user to see child entries.

**RDN** Displays the relative distinguished name (RDN) of the entry.

**Object class**

Displays the object classes of the entry.

**Created**

Lists the date the entry was created.

**Modified**

Lists the date the entry was last modified.

**Modified by**

Lists the identity of the person who last modified the entry.

Select a subtree and click **Expand** to view the next lower level in the subtree. You can click **Collapse/Go to** to move one level back up the subtree hierarchy. You can also click **Find** to locate the entry you want to work on (see “Searching the directory entries” on page 305). After you have located the level for the entry that

you want to work on, select it and choose the operation you want to perform from tool bar or the **Select action** drop-down menu.

---

## Adding an entry

### Using Web Administration

If you have not done so already, expand the **Directory management** category in the navigation area.

1. Click **Add an entry**.
2. Select a filter object class from the drop-down menu and click **Refresh**.
3. Select one **Structural object class** from the list box.
4. Click **Next**.
5. Select a filter object class from the drop-down menu and click **Refresh**.
6. Select any **Auxiliary object classes** you wish to use from the Available box and click **Add**. Repeat this process for each auxiliary object class you want to add. You can also delete an auxiliary object class from the Selected box by selecting it and clicking **Remove**.
7. Click **Next**.
8. In the **Relative DN** field, enter the relative distinguished name (RDN) of the entry that you are adding, for example, cn=John Doe.
9. In the **Parent DN** field, enter the distinguished name of the tree entry you selected, for example, ou=Austin, o=IBM. You can also click **Browse** to select the Parent DN from the list. You can also expand the selection to view other choices lower in the subtree. Specify your choice and click **Select** to specify the Parent DN that you want. The **Parent DN** defaults to the entry selected in the tree.

**Note:** If you started this task from the **Manage entries** panel, this field is prefilled for you. You selected the **Parent DN** before clicking **Add** to start the add entry process. However, if your server supports the modifyDN operation (starting with IBM Tivoli Directory Server version 6.0), the field is still modifiable if the entry is a leaf node. That is, if it has no entries below it, you can move the entry to another parent DN entry.

10. At the **Required attributes** tab enter the values for the required attributes.

**Notes:**

- a. If the attribute is multi-valued and you want to add more than one value for a particular attribute, click **Multiple values**. See "Multiple values for attributes" on page 295.
  - b. If an attribute requires binary data, click **Binary data**. See "Binary data for attributes" on page 295.
  - c. If your server has language tags enabled, click **Language tag value** to add or remove language tag descriptors. See "Language tags" on page 297 and "Language tag values for attributes" on page 298 for more information.
  - d. If an attribute contains referrals, click **Manage referral**. See Chapter 12, "Referrals," on page 153 and "Creating default referrals" on page 157 for more information.
11. Click **Optional attributes**.
  12. At the **Optional attributes** tab enter the values as appropriate for the other attributes.

13. Click **Finish** to create the entry.

## Using the command line

Issue the command:

```
idsldapadd -D <adminDN> -w <adminPW> -i <filename>
```

where <filename> contains the following:

```
dn: cn=John Doe, ou=Austin, o=IBM
cn: John Doe
objectclass: inetOrgPerson
objectclass: organizationalPerson
objectclass: person
objectclass: top
sn: Doe
```

---

## Multiple values for attributes

If the attribute supports multiple values and you want to add more than one value for a particular attribute:

1. Click **Multiple values**.
2. Supply the additional value for the attribute.
3. Click **Add**.
4. Repeat this for each additional value.
5. When you are finished click **OK**.

The values are added to a drop-down menu displayed below the attribute.

If the attribute supports multiple values and you want to remove one or more values for a particular attribute:

1. Click **Multiple values**.
2. Select the value you want to remove.
3. Click **Remove**.
4. Repeat this for each additional value that you want to remove.
5. When you are finished click **OK**.

The values are removed from the drop-down menu displayed below the attribute. The drop-down menu displays the remaining values. If only one value or no values are assigned to the value, the drop-down menu is no longer displayed.

**Note:** If you select a language value tag in the **Display with language tags** menu, then any attribute you add or remove is associated with that language tag. See “Language tag values for attributes” on page 298 for more information about adding language tag values.

---

## Binary data for attributes

### Using Web Administration

If an attribute requires binary data, a **Binary data** button is displayed next to the attribute field. If the attribute has no data the field is blank. Because binary attributes cannot be displayed, if an attribute contains binary data, the field displays **Binary data 1**. If the attribute contains multiple values, the field displays as a drop-down list.

Click the **Binary data** button to work with binary attributes.

You can import, export or remove binary data.

To add binary data to the attribute:

1. Click the **Binary data** button.
2. Click **Import**.
3. You can either enter the path name for the file you want or click **Browse** to locate and select the binary file.
4. Click **Submit file**. A File uploaded message is displayed.
5. Click **Close**. **Binary data 1** is now displayed in the table under **Binary data entries**.
6. Repeat the import process (steps 2 through 5) for as many binary files as you want to add. The subsequent entries are listed as **Binary data 2**, **Binary data 3**, and so on.
7. When you are finished adding binary data, click **OK**.

After the first binary data file has been imported, you can perform two additional operations to export or remove the binary data.

To export binary data:

1. If you have not already done so, click the **Binary data** button.
2. Select the binary file you want to export.
3. Click **Export**.
4. Click on the link **Binary data to download**.
5. Follow the directions of your wizard to either display the binary file or to save it to a new location.
6. Click **Close**.
7. Repeat the import process for as many binary files as you want to export.
8. When you are finished exporting data, click **OK**.

To remove binary data:

1. If you have not already done so, click the **Binary data** button.
2. Check the binary data file you want to remove. For this task multiple files can be selected.
3. Click **Delete**.
4. When you are prompted to confirm the deletion, click **OK**. The binary data marked for deletion are removed from the list.
5. When you are finished deleting data, click **OK**.

**Note:** Binary attributes are not searchable.

## Using the command line

Issue the command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where <filename> contains the following:

```
dn: cn=Bob Garcia, ou=deptABC, ou=Austin, o=IBM
changetype: modify
add: jpegphoto
jpegphoto:< file:///usr/local/directory/photos/Bob.jpg
```

---

## Language tags

### Notes:

1. For language tags to work correctly, your database must be configured as a UTF-8 database.
2. After enabling the language tag feature, if you associate language tags with the attributes of an entry, the server returns the entry with the language tags. This occurs even if you later disable the language tag feature. Because the behavior of the server might not be what the application is expecting, to avoid potential problems, do not disable the language tag feature after it has been enabled.

The term, language tags, defines a mechanism that enables the directory to associate natural language codes with values held in a directory and enables clients to query the directory for values that meet certain natural language requirements. The language tag is a component of an attribute description. The language tag is a string with the prefix **lang-**, a primary subtag of alphabetic characters and, optionally, subsequent subtags connected by a hyphen (-). The subsequent subtags can be any combination of alphanumeric characters, only the primary subtag needs to be alphabetic. The subtags can be any length, the only limitation is that the total length of the tag cannot exceed 240 characters. Language tags are case insensitive; en-us and en-US and EN-US are identical. Language tags are not allowed in components of DN or RDN. Only one language tag per attribute description is allowed.

**Note:** On a per attribute basis, language tags are mutually exclusive with unique attributes. If you have designated a particular attribute as being a unique attribute, it cannot have language tags associated with it.

If language tags are included when data is added to a directory, they can be used with search operations to selectively retrieve attribute values in specific languages. If a language tag is provided in an attribute description within the requested attribute list of a search, then only attribute values in a directory entry which have the same language tag as that provided are to be returned. Thus for a search like:

```
idsldapsearch -b "o=ibm,c=us" (objectclass=organization) description;lang-en
```

the server returns values of an attribute "description;lang-en", but does not return values of an attribute "description" or "description;lang-fr".

If a request is made specifying an attribute without providing a language tag, then all attribute values regardless of their language tag are returned.

The attribute type and the language tag are separated with a semicolon (;) character.

**Note:** RFC2252 allows the semicolon character to be used in the "NAME" part of an AttributeType. However, because this character is being used to separate the AttributeType from the language tag, its usage in the "NAME" part of an AttributeType is no longer permitted as specified in draft-ietf-ldapbis-models-07.txt.

For example, if the client requests a "description" attribute, and a matching entry contains:

```
objectclass: top
objectclass: organization
o: Software GmbH
description: software
```

```
description;lang-en: software products
description;lang-de: Softwareprodukte
postalAddress: Berlin 8001 Germany
postalAddress;lang-de: Berlin 8001 Deutschland
```

the server returns:

```
description: software
description;lang-en: software products
description;lang-de: Softwareprodukte
```

If the search requests a "description;lang-de" attribute, then the server returns:  
description;lang-de: Softwareprodukte

This type of server processing enables directories that contain multi-lingual data to support clients that operate in various languages. If an application is implemented correctly, the German client sees data entered for the lang-de attribute only, and the French client sees data entered for the lang-fr attribute only.

To determine whether the language tag feature is enabled, issue a root DSE search specifying the attribute "ibm-enabledCapabilities".

```
idsldapsearch -b "" -s base objectclass=* ibm-enabledCapabilities
```

If the OID "1.3.6.1.4.1.4203.1.5.4" is returned, the feature is enabled.

If the language tag support is not enabled, any LDAP operation that associates a language tag with an attribute is rejected with the error message:

```
LDAP_NO_SUCH_ATTRIBUTE
```

## Attributes that cannot have associated language tags

The following attributes cannot have language tags associated with them:

- objectclass
- member
- uniquemember
- memberURL
- ibm-memberGroup
- userpassword
- secretkey
- ref
- operational attributes
- configuration attributes
- binary attributes

Or, to generate a list of attributes that cannot have language tags associated with them, use the following command:

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

## Language tag values for attributes

If the attribute supports language tag values and you want to add language tag values for a particular attribute:

1. Click **Language tag values**.
2. In the **Language tag** field, enter the name of the tag you are creating. Remember the tag must begin with the prefix **lang-**.



3. Enter the value for the tag in the **Value** field.
4. Click **Add**.
5. Repeat adding values as necessary, if this attribute has the **Multiple values** feature enabled. If the **Multiple values** button is not enabled, you can enter one language tag value only. See “Multiple values for attributes” on page 295.
6. Click **OK** for your values to be accepted.

**Note:** If you do not click **OK**, your attribute values are not saved.

The values are added to the **Display with language tags** menu.

You can expand the **Display with language tags** menu and select a language tag. Click **Change view** and the attribute values that you have entered for that language tag are displayed. Any values that you add or remove in this view apply to the selected language tag only.

If the attribute supports language tag values and you want to remove one or more values for a particular attribute, see “Removing a language tag descriptor from an entry.”

## Searching for entries containing attributes with language tags

Issuing the command,

```
idsldapsearch -b "o=ibm,c=us" "cn=Mark Anthony" sn
```

return the following results:

```
cn=Mark Anthony,o=IBM,c=US
sn=Anthony
sn;lang-spanish=Antonio
```

**Note:** All versions of “sn” are displayed in the output.

Issuing the command,

```
idsldapsearch -b "o=ibm,c=us" "cn=Mark Anthony" sn;lang-spanish
```

returns the following results.

```
cn=Mark Anthony,o=IBM,c=US
sn;lang-spanish=Antonio
```

**Note:** Only “sn;lang-spanish” is displayed in the output.

Issuing the command,

```
idsldapsearch -b "o=ibm,c=us" "sn;lang-spanish=Antonio"
```

returns the entire entry:

```
cn=Mark Anthony,o=IBM,c=US
objectclass=person
objectclass=top
cn=Mark Anthony
sn=Anthony
sn;lang-spanish=Antonio
```

## Removing a language tag descriptor from an entry

Use either of the following methods to remove a language tag descriptor from an entry:

## Using Web Administration:

From either the **Manage entries** -> **Edit attributes** path or the **Add an entry** -> **Select structural object class** -> **Select auxiliary object class** -> **Enter the attributes** path:

1. Select the attribute from which you want to remove the language tag.
2. Click the **Language tag value** button to access the Language tag values panel .
3. In the **Language tag** field, click the language tag you want to remove.
4. Click **Remove**. The language tag and its values is removed from the menu list.
5. Repeat steps 3 and 4 for each language tag you want to remove.
6. When you have finished, click **OK**.

## Using the command line:

Issue the command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where <filename> contains:

```
dn: cn=Mark Anthony, o=IBM, c=US
changetype: modify
delete:sn;lang-spanish
sn;lang-spanish: Antonio
```

This removes the attribute `sn;lang-spanish` that has the value "Antonio" from the entry.

If you want to delete the entire entry see "Deleting an entry."

---

## Deleting an entry

**Note:** When you are logged into the console, the Web Administration Tool does not permit you to delete the entry that you are logged on as. For example, if you logged on as user `cn=John Doe,ou=mylocale,o=mycompany,c=mycountry`, and you try to delete the entry, `cn=John Doe` from that tree, you receive an error message. You must log on as some other user to delete the John Doe entry.

## Using Web Administration

If you have not done so already, expand the **Directory management** category in the navigation area, then click **Manage entries**. You can expand the various subtrees and select the subtree, the suffix, or the entry that you want to work on. Click **Delete**.

- You are requested to confirm the deletion. Click **OK**.
- The entry is deleted from the entry and you are returned to the list of entries.

## Using the command line

Issue the command:

```
idsldapdelete -D <adminDN> -w <adminPW> "cn=John Doe, ou=Austin, o=IBM"
```

---

## Modifying an entry

### Using Web Administration

If you have not done so already, expand the **Directory management** category in the navigation area, then click **Manage entries**. You can expand the various subtrees and select the entry that you want to work on. Click **Edit attributes**.

1. View the object class inheritance for the entry in the Object class drop-down menu. Object classes are sorted by inheritance.
2. In the **Relative DN** field, you can change the relative distinguished name (RDN) of the entry that you are editing; for example, change `cn=Bob Garcia` to `cn=Robert Garcia`.
3. In the **Parent DN** field, the distinguished name of the tree entry you selected is displayed. If your server supports the modifyDN operation (starting with IBM Tivoli Directory Server version 6.0), you can modify the Parent DN with a new superior attribute on a leaf node. You can either edit this field or you can click **Browse**, select a Parent DN from the list, and click **Select** to change the Parent DN of the entry.
4. At the **Required attributes** tab enter the values for the required attributes.

#### Notes:

- a. If the attribute is multi-valued and you want to add more than one value for a particular attribute, click **Multiple values**. See “Multiple values for attributes” on page 295.
  - b. If an attribute requires binary data, click **Binary data**. See “Binary data for attributes” on page 295
  - c. If your server has language tags enabled, click **Language tag value** to add or remove language tag descriptors. See “Language tags” on page 297 and “Language tag values for attributes” on page 298 for more information.
  - d. If an attribute contains referrals, click **Manage referral**. See Chapter 12, “Referrals,” on page 153 and “Creating default referrals” on page 157 for more information.
5. Click **Optional attributes**.
  6. At the **Optional attributes** tab enter the values as appropriate for the other attributes.
  7. Click **OK** to modify the entry.

### Using the command line

#### Renaming an entry

Issue the following command to rename an entry, changing RDN from `cn=Bob Garcia` to `cn=Robert Garcia`:

```
idsldapmodrdn -D <adminDN> -w <adminPW>
-r "cn=Bob Garcia, ou=deptABC, ou=Austin, o=IBM" "cn=Robert Garcia"
```

#### Moving an entry

Issue the following command to move an entry, for example, moving Bob to a new department:

```
idsldapmodrdn -D <adminDN> -w <adminPW> -s "ou=deptXYZ, ou=Austin,
o=IBM" "cn=Bob Garcia, ou=deptABC, ou=Austin, o=IBM" "cn=Bob Garcia"
```

You can also issue the following command to move an entry:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where *<filename>* contains the following:

```
dn: cn=Bob Garcia, ou=deptABC, ou=Austin, o=IBM
changetype: modrdn
newrdn: cn=Bob Garcia
deleteoldrdn: 0
newsuperior: ou=deptXYZ, ou=Austin, o=IBM
```

### Modifying attributes of an entry

Issue the following command to modify the attributes of an entry, for example, replacing the roomNumber attribute value:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where *<filename>* contains the following:

```
dn: cn=Bob Garcia, ou=deptABC, ou=Austin, o=IBM
changetype: modify
replace: roomNumber
roomNumber: 4B-014
```

---

## Copying an entry

This function is useful if you are creating similar entries. The copy inherits all the attributes of the original. You need to make some modifications to name the new entry.

### Using Web Administration

If you have not done so already, expand the **Directory management** category in the navigation area, then click **Manage entries**. You can expand the various subtrees and select the entry, such as John Doe, that you want to work on. Expand the **Select Action** drop-down menu, select **Copy**, and click **Go**.

- Change the RDN entry in the DN field. For example change cn=John Doe to cn=Jim Smith.
- If your server supports the modifyDN operation (starting with IBM Tivoli Directory Server version 6.0), you can modify the Parent DN with a new superior attribute on a leaf node. You can either edit this field or you can click **Browse**, select a Parent DN from the list, and click **Select** to change the Parent DN of the entry.
- On the required attributes tab, change the cn entry to the new RDN. In this example Jim Smith.
- Change the other required attributes as appropriate. In this example change the sn attribute from Doe to Smith.
- Change the optional attributes as appropriate.
- When you have finished making the necessary changes click **OK** to create the new entry.
- The new entry Jim Smith is added to the bottom of the entry list.

**Note:** This procedure copies only the attributes of the entry. The group memberships of the original entry are not copied to the new entry. See “Managing memberships for an entry” on page 340 to add memberships to the entry.

### Using the command line

Do the following to copy an entry using the command line:

1. Search to get the current entry back in LDIF form. Issue the following command:

```
idsldapsearch -L -s base -b "cn=Bob Garcia,
ou=deptABC, ou=Austin, o=IBM" (objectclass=*)
```

which returns something like the following (save this information to a file):

```
dn: cn=Bob Garcia, ou=deptABC, ou=Austin, o=IBM
cn: Bob Garcia
cn: Robert Garcia
objectclass: inetOrgPerson
objectclass: organizationalPerson
objectclass: person
objectclass: top
sn: Garcia
roomNumber: 4B-014
```

2. Edit the entry to change the name and room number in the new entry:

```
DN dn: Matt Morris, ou=deptABC, ou=Austin, o=IBM
cn: Matt Morris
cn: Matthew Morris
objectclass: inetOrgPerson
objectclass: organizationalPerson
objectclass: person
objectclass: top
sn: Morris
roomNumber: 2B-001
```

3. Add the new entry. Issue the following command:

```
idsldapadd -D <adminDN> -w <adminPW> -i <filename>
```

---

## Editing access control lists for an entry

To edit the access control lists (ACLs) for an entry:

1. If you have not done so already, expand the **Directory management** category in the navigation area .
2. Click **Manage entries**.
3. Expand the various subtrees and select the entry, such as `cn=Robert Garcia,ou=Austin,o=ibm,c=us`, that you want to work on.
4. Expand the **Select Action** drop-down menu.
5. Select **Edit ACL**.
6. Click **Go**.

To view ACL properties using the Web Administration Tool utility and to work with ACLs, see “Working with ACLs” on page 318.

See Chapter 17, “Access control lists,” on page 309 for additional information.

---

## Adding an auxiliary object class

### Using Web Administration

If you have not done so already, expand the **Directory management** category in the navigation area, then click **Manage entries**. You can expand the various subtrees and select the entry, such as John Doe, that you want to work on. From the **Select Action** drop-down menu scroll down and select **Add auxiliary class** and click **Go**.

1. Select a filter object class from the drop-down menu and click **Refresh**.

2. Select any **Auxiliary object classes** you wish to use from the Available box and click **Add**. Repeat this process for each auxiliary object class you want to add. You can also delete an auxiliary object class from the Selected box by selecting it and clicking **Remove**.
3. Click **Next**.
4. At the **Required attributes** tab enter the values for the required attributes.

**Notes:**

- a. If the attribute is multi-valued and you want to add more than one value for a particular attribute, click **Multiple values**. See “Multiple values for attributes” on page 295.
  - b. If an attribute requires binary data, click **Binary data**. See “Binary data for attributes” on page 295
  - c. If your server has language tags enabled, click **Language tag value** to add or remove language tag descriptors. See “Language tags” on page 297 and “Language tag values for attributes” on page 298 for more information.
  - d. If an attribute contains referrals, click **Manage referral**. See Chapter 12, “Referrals,” on page 153 and “Creating default referrals” on page 157 for more information.
5. Click **Optional attributes**.
  6. At the **Optional attributes** tab enter the values as appropriate for the other attributes.
  7. Click **Finish** to modify the entry.

## Using the command line

Issue the following command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where <filename> contains the following:

**Note:** The hyphen ( - ) on the 5th line is important.

```
dn: cn=Bob Garcia, ou=deptABC, ou=Austin, o=IBM
changetype: modify
add: objectclass
objectclass: uniquelyIdentifiedUser
-
add: serialNumber
serialNumber: 738393
```

Any attributes that are required by the auxiliary object class must be added to the entry as part of the same modify operation.

---

## Deleting an auxiliary object class

Although you can delete an auxiliary class during the add auxiliary class procedure, it is easier to use the delete auxiliary object class function if you are going to delete a single auxiliary class from an entry. However, it might be more convenient to use the add auxiliary class procedure if you are going to delete multiple auxiliary classes from an entry.

## Using Web Administration

1. If you have not done so already, expand the **Directory management** category in the navigation area, then click **Manage entries**. You can expand the various

subtrees and select the entry, such as John Doe, that you want to work on. From the **Select Action** drop-down menu scroll down and select **Delete auxiliary class** and click **Go**.

2. From the list of auxiliary object classes select the auxiliary classes you want to delete and press **OK**.
3. You are asked to confirm the deletion, click **OK**.
4. The auxiliary object classes are deleted from the entry and you are returned to the list of entries.

## Using the command line

Issue the following command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where <filename> contains the following:

**Note:** The hyphen ( - ) on the 5th line is important.

```
dn: cn=Bob Garcia, ou=deptABC, ou=Austin, o=IBM
changetype: modify
delete: objectclass
objectclass: uniquelyIdentifiedUser
-
delete: serialNumber
serialNumber: 738393
```

Any attributes that were allowed in the entry only because of the auxiliary object class must be deleted from the entry as part of the same modify operation.

---

## Searching the directory entries

There are three options for searching the directory tree:

- A Simple search using a predefined set of search criteria
- An Advanced search using a user-defined set of search criteria
- A Manual search

The search options are accessible by expanding the **Directory management** category in the navigation area, click **Find entries**. Select one of the following tabs:

**Note:** Binary attributes such as userpassword are only searchable to find if they actually exist.

### Search filters

Select on of the following types of searches:

#### Simple search

A simple search uses a default search criteria:

- Base DN is **All suffixes**
- Search scope is **Subtree**
- Search size is **500**
- Time limit is **900**
- Alias dereferencing is **never**
- Chase referrals is deselected (off)

To perform a simple search:

1. On the **Search filter** tab, click **Simple**.
2. Select an object classes from the drop-down list.
3. If your server has language tags enabled, you can specify a language tag. See "Language tags" on page 297 for more information.
4. Select a specific attribute for the selected entry type. If you select to search on a specific attribute, select an attribute from the drop-down list and enter the attribute value in the **Is equal to** box. If you do not specify an attribute, the search returns all the directory entries of the selected entry type.
5. Click **OK**.

### Advanced search

An advanced search enables you to specify search constraints and enable search filters. The default search criteria are the same as those for a simple search.

- To perform an advanced search:
  1. On the **Search filter** tab, click **Advanced** .
  2. Click **Add**.
  3. Select an **Attribute** from the drop-down list.
  4. If your server has language tags enabled, you can specify a language tag. See "Language tags" on page 297 for more information.
  5. Select a **Comparison** operator
    - **Is equal to** - The attribute is equal to the value.
    - **Is not equal to** - The attribute is not equal to the value.
    - **Is less than or equal to** - The attribute is less than or equal to the value.
    - **Is greater than or equal to** - The attribute is greater than or equal to the value.
    - **Is approximately equal to** - The attribute is approximately equal to the value.
  6. Enter the **Value** for comparison.
  7. If you already added at least one search filter, specify the additional criteria and select an operator from the **Operator** drop-down menu. The **AND** command returns entries that match both sets of search filter criteria. The **OR** command returns entries that match either set of search filter criteria. The default operator is **AND**.
  8. Click **OK** to add the search filter criteria to the advanced search.
 

The Search results table contains the following columns:

    - **Select** - Select the radio button next to the name of the filter you want to add, edit or delete.
    - **Attribute** - The attribute on which the filter is performed, for example, objectclass.
    - **Comparison** - The filter's comparison criteria, for example, Is equal to.
    - **Value** - The value used for comparison; for example, the wildcard value (\*).
    - **Operator** - The search operator that was specified, for example, AND.
  9. Click the check box to select each filter that you want to use in the search.
  10. Change any of the default settings on the **Options** tab. See "Options" on page 307.
  11. Click **OK** to begin the search.
 

The Search results table contains the following columns:



- **Select** - Select the radio button next to the name of the entry you want to perform an action on.
  - **RDN** - The RDN of the entry.
  - **Object class** - The object class to which the entry belongs.
  - **Created** - The date the entry was created.
  - **Last modified** - The date the entry was last modified.
  - **Last modified by** - The ID of the user who last modified the entry.
12. After viewing the search results click, you can modify the entry attributes (see Chapter 16, "Working with directory entries," on page 293) or click **Close** to return to the Find entries panel.

To modify a search filter:

1. Select the filter you want to modify.
2. Click edit.
3. Change any of the fields that were set when you added the search filter.
4. Click **OK**.

To remove the search filters:

- Click the check box to select each filter that you want to remove.
- Click **Remove** to remove the search filter criteria from the advanced search.

**Note:** If you want to clear all search filters, click **Remove all**.

## Manual search

**Notes:**

1. Avoid using wildcard searches where the wildcard is in any position other than the leading character in a term, or a trailing character. Use wildcard searches that are similar to the following (leading character):

sn=\*term

or the following (trailing character):

sn=term\*

2. Do not use both wildcard searches simultaneously.

Use this method to create a search filter. The default search criteria are the same as those for a simple search. For example to search on surnames enter `sn=*` in the field. If you are searching on multiple attributes, you must use search filter syntax. For example to search for the surnames of a particular department you enter:

`(&(sn=*)(dept=<departmentname>))`

## Options

At the **Options** tab:

- **Search base DN** - Choose one of the radio buttons to select a search base:
  - **DN** - Select the DN radio button if you want to specify the search base explicitly. Enter the search base in the DN field; for example, `o=ibm,c=us`.
  - **Suffix** - Select a suffix from the Suffix drop-down menu to search only within that suffix. If you started this task from the "Manage entries" panel, this field is prefilled for you.
  - **All suffixes** - Select All suffixes to search the entire tree
- **Search scope**

- Select **Object** to search only within the selected object.
- Select **Single level** to search only within the immediate children of the selected object.
- Select **Subtree** to search the selected object all descendants of the selected object.
- **Search size limit** - Enter the maximum number of entries to search or select **Unlimited**.
- **Search time limit** - Enter the maximum number of seconds for the search or select **Unlimited**.
- If the server supports alias dereferencing, select a type of **Alias dereferencing** from the drop-down list.
  - **Never** - If the selected entry is an alias, it is not dereferenced for the search, that is, the search ignores the reference to the alias. Also, entries found in the search are not dereferenced.
  - **Find** - If the selected entry is an alias, the search dereferences the alias and search from the location of the alias.
  - **Search** - The selected entry is not dereferenced, but any entries found in the search are dereferenced.
  - **Always** - All aliases encountered in the search are dereferenced.
- Select the **Chase referrals** check box to follow referrals to another server if a referral is returned in the search. When a referral directs the search to another server, the connection to the server uses the current credentials. If you are logged in as Anonymous you might need to log in to the server using an authenticated DN.

If an entry is found on the referred server, the **Search results** panel shows only the DN of the entry. Other columns such as object class, modified timestamp and so forth are not shown. You are not able to perform such operations as **Edit Acls**, **Delete**, **Add auxiliary** or **Delete auxiliary** on the referral entry.

See Chapter 12, "Referrals," on page 153 and Chapter 17, "Access control lists," on page 309 for more information.

See "Setting Searches" on page 99 for additional information about searches.

---

## Chapter 17. Access control lists

The following sections describe access control lists (ACLs) and how to manage them.

---

### Overview

Access control lists (ACLs) provide a means to protect information stored in a LDAP directory. Administrators use ACLs to restrict access to different portions of the directory, or specific directory entries. LDAP directory entries are related to each other by a hierarchical tree structure. Each directory entry (or object) contains the distinguished name of the object as well as a set of attributes and their corresponding values.

The access control model defines two sets of attributes:

- The entryOwner information
- The Access Control Information (ACI)

In conformance with the LDAP model, the ACI information and the entryOwner information is represented as attribute-value pairs. The LDIF syntax can be used to administer these values.

### EntryOwner information

The entryOwner information controls which subjects can define the ACIs. An entry Owner also acquires full access rights to the target object. The attributes that define entry ownership are:

- entryOwner - Explicitly defines an entry owner.
- ownerPropagate - Specifies whether the permission set is propagated to the subtree descendant entries.

The entry owners have complete permissions to perform any operation on the object regardless of the aclEntry. Additionally, the entry owners are the only ones who are permitted to administer the aclEntries for that object. EntryOwner is an access control subject, it can be defined as individuals, groups or roles.

**Note:** The directory administrator and administration group members are the entryOwners for all objects in the directory by default, and this entryOwnership cannot be removed from any object.

### Access control information

The ACI specifically defines a subject's permission to perform a given operation against certain LDAP objects.

#### Non-filtered ACLs

This type of ACL applies explicitly to the directory entry that contains them, but may be propagated to none or all of its descendant entries. The default behavior of the non-filtered ACL is to propagate. The attributes that define non-filtered ACLs are:

- aclEntry - Defines a permission set.
- aclPropagate - Specifies whether the permission set is propagated to the subtree descendant entries.

## Filtered ACLs

Filter-based ACLs differ in that they employ a filter-based comparison, using a specified object filter, to match target objects with the effective access that applies to them.

Although they perform the same function, the behavior of the two types of ACLs is significantly different. Filter-based ACLs do not propagate in the same way that non-filter-based ACLs currently do. By nature, they inherently propagate to any comparison matched objects in the associated subtree. For this reason, the `aclPropagate` attribute, which is used to stop propagation of non-filter ACLs, does not apply to the new filter-based ACLs.

The default behavior of filter-based ACLs to accumulate from the lowest containing entry, upward along the ancestor entry chain, to the highest containing entry in the DIT. The effective access is calculated as the union of the access rights granted, or denied, by the constituent ancestor entries. There is an exception to this behavior. For compatibility with the subtree replication feature, and to allow greater administrative control, a ceiling attribute is used as a means to stop accumulation at the entry in which it is contained.

A separate set of access control attributes are used specifically for filter-based ACL support, rather than merging filter-based characteristics into the existing non-filter based ACLs. The attributes are:

- `ibm-filterAclEntry`
- `ibm-filterAclInherit`

The `ibm-filterAclEntry` attribute has the same format as `aclEntry`, with the addition of an object filter component. The associated ceiling attribute is `ibm-filterAclInherit`. By default it is set to true. When set to false, it terminates the accumulation.

---

## The access control attribute syntax

Each of these attributes can be managed using LDIF notation. The syntax for the new filter-based ACL attributes are modified versions of the current non-filter-based ACL attributes. The following defines the syntax for the `ACL` and `entryOwner` attributes using baccus naur form (BNF).

```
<aclEntry> ::= <subject> [":" <rights>]
<aclPropagate> ::= "true" | "false"
<ibm-filterAclEntry> ::= <subject> ":" <object filter> [":" <rights>]
<ibm-filterAclInherit> ::= "true" | "false"
<entryOwner> ::= <subject>
<ownerPropagate> ::= "true" | "false"
<subject> ::= <subjectDnType> ':' <subjectDn> |
 <pseudoDn>
<subjectDnType> ::= "role" | "group" | "access-id"
<subjectDn> ::= <DN>
<DN> ::= distinguished name as described in RFC 2251, section 4.1.3.
<pseudoDn> ::= "group:cn=anybody" | "group:cn=authenticated" |
 "access-id:cn=this"
```

```

<object filter> ::= string search filter as defined in RFC 2254, section 4
 (extensible matching is not supported).
<rights> ::= <accessList> [":" <rights>]
<accessList> ::= <objectAccess> | <attributeAccess> |
 <attributeClassAccess>
<objectAccess> ::= "object:" [<action> ":"] <objectPermissions>
<action> ::= "grant" | "deny"
<objectPermissions> ::= <objectPermission> [<objectPermissions>]
<objectPermission> ::= "a" | "d" | ""
<attributeAccess> ::= "at." <attributeName> ":" [<action> ":"]
 <attributePermissions>
<attributeName> ::= attributeType name as described in RFC 2251, section 4.1.4.
 (OID or alpha-numeric string with leading
 alphabet, "-" and ";" allowed)
<attributePermissions> ::= <attributePermission>
 [<attributePermissions>]
<attributePermission> ::= "r" | "w" | "s" | "c" | ""
<attributeClassAccess> ::= <class> ":" [<action> ":"]
 <attributePermissions>
<class> ::= "normal" | "sensitive" | "critical" | "system" | "restricted"

```

## Subject

A subject (the entity requesting access to operate on an object) consists of the combination of a DN (Distinguished Name) type and a DN. The valid DN types are: access Id, Group and Role.

The DN identifies a particular access-id, role or group. For example, a subject might be "access-id: cn=personA, o=IBM or group: cn=deptXYZ, o=IBM".

Because the field delimiter is the colon (:), a DN containing colons must be surrounded by double-quotation marks ("""). If a DN already contains characters with double-quotation marks, these characters must be escaped with a backslash (\).

All directory groups can be used in access control.

**Note:** Any group of **AccessGroup**, **GroupOfNames**, **GroupofUniqueNames**, or **groupOfURLs** structural objectclasses or the **ibm-dynamicGroup**, **ibm-staticGroup** auxiliary objectclasses can be used for access control.

Another DN type used within the access control model is role. While roles and groups are similar in implementation, conceptually they are different. When a user is assigned to a role, there is an implicit expectation that the necessary authority has already been set up to perform the job associated with that role. With group membership, there is no built in assumption about what permissions are gained (or denied) by being a member of that group.

Roles are similar to groups in that they are represented in the directory by an object. Additionally, roles contain a group of DNs. Roles that are used in access control must have an objectclass of **AccessRole**.

## Pseudo DNs

Pseudo DNs are used in access control definition and evaluation. The directory contains several pseudo DNs (for example, "group:cn=Anybody" and "access-id:cn=this"), which are used to refer to large numbers of DNs that share a common characteristic, in relation to either the operation being performed or the object on which the operation is being performed.

Three pseudo DNs are supported by LDAP version 3:

### **access-id: cn=this**

When specified as part of an ACL, this DN refers to the bindDN, which matches the DN on which the operation is performed. For example, if an operation is performed on the object "cn=personA, ou=IBM, c=US" and the bindDn is "cn=personA, ou=IBM, c=US", the permissions granted are a combination of those given to "cn=this" and those given to "cn=personA, ou=IBM, c=US".

### **group: cn=anybody**

When specified as part of an ACL, this DN refers to all users, even those that are unauthenticated. Users cannot be removed from this group, and this group cannot be removed from the database.

### **group: cn=Authenticated**

This DN refers to any DN that has been authenticated by the directory. The method of authentication is not considered.

**Note:** "cn=Authenticated" refers to a DN that has been authenticated anywhere on the server, regardless of where the object representing the DN is located. It should be used with caution, however. For example, under one suffix, "cn=Secret" could be a node called "cn=Confidential Material" which has an aclentry of "group:cn=Authenticated:normal:rsc". Under another suffix, "cn=Common" could be the node "cn=Public Material". If these two trees are located on the same server, a bind to "cn=Public Material" would be considered authenticated, and would get permission to the normal class on the "cn= Confidential Material" object.

## Examples of pseudo DNs

Some examples of pseudo DNs:

### **Example 1**

Consider the following ACL for object: cn=personA, c=US AclEntry:

```
access-id: cn = this:critical:rWSC
AclEntry: group: cn=Anybody: normal:rsc
AclEntry: group: cn=Authenticated: sensitive:rcs
```

| User Binding as  | Would receive                          |
|------------------|----------------------------------------|
| cn=personA, c=US | normal:rsc:sensitive:rcs:critical:rWSC |
| cn=personB, c=US | normal:rsc:sensitive:rsc               |
| NULL (unauth.)   | normal:rsc                             |

In this example, personA receives permissions granted to the "cn=this" ID, and permissions given to both the "cn=Anybody" and "cn=Authenticated" pseudo DN groups.

### Example 2

Consider the following ACL for object: cn=personA, c=US AclEntry:

```
access-id:cn=personA, c=US: object:ad
AclEntry: access-id: cn = this:critical:rwsc
AclEntry: group: cn=Anybody: normal:rsc
AclEntry: group: cn=Authenticated: sensitive:rsc
```

For an operation performed on cn=personA, c=US:

| User Binding as  | Would receive            |
|------------------|--------------------------|
| cn=personA, c=US | object:ad:critical:rwsc  |
| cn=personB, c=US | normal:rsc:sensitive:rsc |
| NULL (unauth.)   | normal:rsc               |

In this example, personA receives permissions granted to the "cn=this" ID, and those given to the DN itself "cn=personA, c=US". Note that the group permissions are not given because there is a more specific aclentry ("access-id:cn=personA, c=US") for the bind DN ("cn=personA, c=US").

### Example 3

Consider the following ACL for object: cn=personA, c=US AclEntry, where you want to give that user the ability to change his or her own password:

```
access-id:cn=this:at.userpassword:rwsc
```

| User Binding as  | Would receive        |
|------------------|----------------------|
| cn=personA, c=US | at.userpassword:rwsc |

## Object filter

This parameter applies to filtered ACLs only. The string search filter as defined in RFC 2254, is used as the object filter format. Because the target object is already known, the string is not used to perform an actual search. Instead, a filter-based compare on the target object in question is performed to determine if a given set of `ibm-filterAclEntry` values apply to it.

## Rights

Access rights can apply to an entire object or to attributes of the object. The LDAP access rights are discreet. One right does not imply another right. The rights may be combined together to provide the desired rights list following a set of rules discussed later. Rights can be of an unspecified value, which indicates that no access rights are granted to the subject on the target object. The rights consist of three parts:

#### Action:

Defined values are **grant** or **deny**. If this field is not present, the default is set to **grant**.

#### Permission:

There are six basic operations that may be performed on a directory object. From these operations, the base set of ACI permissions are taken. These are: add an entry, delete an entry, read an attribute value, write an attribute value, search for an attribute, and compare an attribute value.

The possible attribute permissions are: read ( r ), write ( w ), search ( s ), and compare ( c ). Additionally, object permissions apply to the entry as a whole. These permissions are add child entries ( a ) and delete this entry ( d ).

The following table summarizes the permissions needed to perform each of the LDAP operations.

| Operation     | Permission Needed                                                                                                                                                                                                                                                                |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| idsldapadd    | add (on parent)                                                                                                                                                                                                                                                                  |
| idsldapdelete | delete (on object)                                                                                                                                                                                                                                                               |
| idsldapmodify | write (on attributes being modified)                                                                                                                                                                                                                                             |
| idsldapsearch | <ul style="list-style-type: none"> <li>• search, read (on attributes in RDN)</li> <li>• search (on attributes specified in the search filter)</li> <li>• search (on attributes returned with just names)</li> <li>• search, read (on attributes returned with values)</li> </ul> |
| idsldapmodrdn | write (on RDN attributes)                                                                                                                                                                                                                                                        |

For search operations, the subject is required to have search (s) access to all the attributes in the search filter or no entries are returned. For returned entries from a search, the subject is required to have search (s) and read (r) access to all the attributes in the RDN of the returned entries or these entries are not returned.

In the following example, the **at.telephoneNumber:rsc** permission set grants members of the **cn=Bowling Team, ou=Groups, o=IBM, C=US** read only access to only the **telephoneNumber** attribute contained in this entry. The **at.cn:rsc** permission set ensures that the RDN search criteria is met. For this example the only the **cn** or **telephoneNumber** attributes can be used in a search filter. If the title attribute was to be used in a search filter then an additional **at.title:rsc** permission set would have to be added for the search to be successful.

```
dn: cn=Bonnie Daniel, ou=Widget Division, ou=Austin, o=IBM, c=US
objectclass: person
objectclass: organizationalPerson
cn: Bonnie Daniel
sn: Daniel
telephonenumber: 1-812-855-7453
internationalISDNNumber: 755-7453
title: RISC Manufacturing
seealso: cn=Mary Burnnet, ou=Widget Division, ou=Austin, o=IBM, c=US
postalcode: 1515
aclentry: group: cn=Bowling Team, ou=Groups, o=IBM, C=US: at.cn:rsc:
at.telephoneNumber:r
```

#### Access Target:

These permissions can be applied to the entire object (add child entry, delete entry), to an individual attribute within the entry, or can be applied to groups of attributes (Attribute Access Classes) as described in the following.

Attributes requiring similar permissions for access are grouped together in classes. Attributes are mapped to their attribute classes in the directory schema file. These classes are discrete; access to one class does not imply access to another class. Permissions are set with regard to the attribute



access class as a whole. The permissions set on a particular attribute class apply to all attributes within that access class unless individual attribute access permissions are specified.

IBM defines five attribute classes that are used in evaluation of access to user attributes: **normal**, **sensitive**, **critical**, **system**, and **restricted**. As examples, the attribute **commonName** belongs to the normal class, and the attribute **userPassword** belongs to the critical class. User defined attributes belong to the normal access class unless otherwise specified.

The system class attributes that apply to access control are:

- **aclSource**
- **ibm-effectiveAcl**
- **ownerSource**

These are attributes maintained by the LDAP server and are read-only to the directory users and administrators. **OwnerSource** and **aclSource** are described in the Propagation section.

The restricted class attributes that define access control are:

- **aclEntry**
- **aclPropagate**
- **entryOwner**
- **ibm-filterAclEntry**
- **ibm-filterAclInherit**
- **ownerPropagate**

By default all users have read access to the restricted attributes but only **entryOwners** can create, modify, and delete these attributes.

---

## Propagation

Entries on which an **aclEntry** has been placed are considered to have an explicit **aclEntry**. Similarly, if the **entryOwner** has been set on a particular entry, that entry has an explicit owner. The two are not intertwined, an entry with an explicit owner may or may not have an explicit **aclEntry**, and an entry with an explicit **aclEntry** might have an explicit owner. If either of these values is not explicitly present on an entry, the missing value is inherited from an ancestor node in the directory tree.

Each explicit **aclEntry** or **entryOwner** applies to the entry on which it is set. Additionally, the value might apply to all descendants that do not have an explicitly set value. These values are considered propagated; their values propagate through the directory tree. Propagation of a particular value continues until another propagating value is reached.

**Note:** Filter-based ACLs do not propagate in the same way that non-filter-based ACLs do. They propagate to any comparison matched objects in the associated subtree. See “Filtered ACLs” on page 310 for more information on the differences.

**AclEntry** and **entryOwner** can be set to apply to just a particular entry with the propagation value set to “false”, or an entry and its subtree with the propagation value set to “true”. Although both **aclEntry** and **entryOwner** can propagate, their propagation is not linked in anyway.

The **aclEntry** and **entryOwner** attributes allow multiple values within the same entry, however, the propagation attributes, **aclPropagate** and **ownerPropagate**, can only have a single value within the same entry.

The system attributes **aclSource** and **ownerSource** contain the DN of the effective node from which the **aclEntry** or **entryOwner** are evaluated, respectively. If no such node exists, the value **default** is assigned.

An object's effective access control definitions can be derived by the following logic:

- If there is a set of explicit access control attributes at the object, then that is the object's access control definition.
- If there is no explicitly defined access control attributes, then traverse the directory tree upwards until an ancestor node is reached with a set of propagating access control attributes.
- If no such ancestor node is found, the default access described in "Access evaluation" is granted to the subject.

---

## Access evaluation

Access for a particular operation is granted or denied based on the subject's bind DN for that operation on the target object. Processing stops as soon as access can be determined.

The checks for access are done by first finding the effective **entryOwnership** and **ACI** definition, checking for entry ownership, and then by evaluating the object's ACI values.

Filter-based ACLs accumulate from the lowest containing entry, upward along the ancestor entry chain, to the highest containing entry in the DIT. The effective access is calculated as the union of the access rights granted, or denied, by the constituent ancestor entries. The existing set of specificity and combinatory rules are used to evaluate effective access for filter based ACLs.

Filter-based and non-filter-based attributes are mutually exclusive within a single containing directory entry. Placing both types of attributes into the same entry is not allowed, and is a constraint violation. Operations associated with the creation of, or updates to, a directory entry fail if this condition is detected.

When calculating effective access, the first ACL type to be detected in the ancestor chain of the target object entry sets the mode of calculation. In filter-based mode, non-filter-based ACLs are ignored in effective access calculation. Likewise, in non-filter-based mode, filter-based ACLs are ignored in effective access calculation.

To limit the accumulation of filter-based ACLs in the calculation of effective access, an **ibm-filterAclInherit** attribute set to a value of "false" may be placed in any entry between the highest and lowest occurrence of **ibm-filterAclEntry** in a given subtree. This causes the subset of **ibm-filterAclEntry** attributes above it in the target object's ancestor chain to be ignored.

To exclude the accumulation of filter-based ACLs in the calculation of effective access, an **ibm-filterAclInherit** attribute set to a value of "false" may be placed in any entry below the lowest occurrence of **ibm-filterAclEntry** in a given subtree. This causes all **ibm-filterAclEntry** attributes above it in the target object's ancestor chain to be ignored. The resulting access resolves to the default filter ACL value.

By default, the directory administrator, administration group members, and the master server (or peer server for replication) get full access rights to all objects in the directory except write access to system attributes. Other **entryOwners** get full access rights to the objects under their ownership except write access to system attributes. By default all users have read access rights to normal, system, and restricted attributes. If the requesting subject has **entryOwnership**, access is determined by the above default settings and access processing stops.

**Note:** If explicit ACLs are set on an entry, but no explicit ACLs are set for system attributes, then the requester is automatically granted rsc (read, search, and compare) permissions. To deny access, you must deny it explicitly. Access is not denied by default.

If the requesting subject is not an entryOwner, then the ACI values for the object entries are checked. The access rights as defined in the ACIs for the target object are calculated by the specificity and combinatory rules.

#### **Specificity rule**

The most specific acEntry definitions are the ones used in the evaluation of permissions granted/denied to a user. The levels of specificity are:

- Access-id is more specific than group or role. Groups and roles are on the same level.
- Within the same **dnType** level, individual attribute level permissions are more specific than attribute class level permissions.
- Within the same attribute or attribute class level, **deny** is more specific than **grant**.

#### **Combinatory rule**

Permissions granted to subjects of equal specificity are combined. If the access cannot be determined within the same specificity level, the access definitions of lesser specific level are used. If the access is not determined after all defined ACIs are applied, the access is denied.

**Note:** After a matching access-id level **acEntry** is found in access evaluation, the group level acEntries are not included in access calculation. The exception is that if the matching access-id level **acEntries** are all defined under cn=this, then all matching group level **acEntries** are also combined in the evaluation.

In other words, within the object entry, if a defined ACI entry contains an access-id subject DN that matches the bind DN, then the permissions are first evaluated based on that acEntry. Under the same subject DN, if matching attribute level permissions are defined, they supersede any permissions defined under the attribute classes. Under the same attribute or attribute class level definition, if conflicting permissions are present, denied permissions override granted permissions.

**Note:** A defined null value permission prevents the inclusion of less specific permission definitions.

If access still can not be determined and all found matching acEntries are defined under "cn=this", then group membership is evaluated. If a user belongs to more than one groups, the user receives the combined permissions from these groups. Additionally, the user automatically belongs to the cn=Anybody group and

possibly the `cn=Authenticated` group if the user did an authenticated bind. If permissions are defined for those groups, the user receives the specified permissions.

**Note:** Group and Role membership is determined at bind time and last until either another bind takes place, or until an unbind request is received. Nested groups and roles, that is a group or role defined as a member of another group or role, are not resolved in membership determination nor in access evaluation.

For example, assume `attribute1` is in the sensitive attribute class, and user `cn=Person A, o=IBM` belongs to both `group1` and `group2` with the following `aclEntries` defined:

1. `aclEntry: access-id: cn=Person A, o=IBM: at.attribute1:grant:rsc:sensitive:deny:rsc`
2. `aclEntry: group: cn=group1,o=IBM:critical:deny:rWSC`
3. `aclEntry: group: cn=group2,o=IBM:critical:grant:r:normal:grant:rsc`

This user gets:

- Access of `'rsc'` to `attribute1`, (from 1. Attribute level definition supersedes attribute class level definition).
- No access to other sensitive class attributes in the target object, (from 1).
- No other rights are granted (2 and 3 are NOT included in access evaluation).

For another example, with the following `aclEntries`:

1. `aclEntry: access-id: cn=this: sensitive`
2. `aclEntry: group: cn=group1,o=IBM:sensitive:grant:rsc:normal:grant:rsc`

The user has:

- no access to sensitive class attributes, (from 1. Null value defined under `access-id` prevents the inclusion of permissions to sensitive class attributes from `group1`).
- and access of `'rsc'` to normal class attributes (from 2).

---

## Working with ACLs

The following sections describe various task that you can perform to manage ACLs.

### Using the Web Administration Tool utility to manage ACLs

To view ACL properties using the Web Administration Tool utility and to work with ACLs.

1. Click **Directory management**.
2. Click **Manage entries**.
3. Select a directory entry. For example, `ou=Widget Division,ou=Austin,o=ibm,c=us`.
4. Expand the **Select Action** drop-down menu.
5. Select **Edit ACL**.
6. Click **Go**.

**Note:** The Edit ACL panel is displayed with the **Effective ACLs** tab preselected. This panel has five tabs:

- Effective ACLs

- Effective owners
- Non-filtered ACLs
- Filtered ACLs
- Owners

The **Effective ACLs** and **Effective owners** tabs contain read-only information about the ACLs.

### Effective ACLs

Effective Access Control Lists (ACLs) are the explicit and inherited ACLs of the selected entry. To view the effective ACLs for the selected entry, click the Load button at the top of the table. The Effective ACLs table contains read-only information in the following columns:

- **Select** - Select the radio button next to the name of an ACL you want to view.
- **Subject DN** - The distinguished name of the entry to which access is being granted or denied.
- **Subject type** - The type of ACL. There are three subject types:
  - **access-id** - Associates access with a user.
  - **group** - Associates access with users who are members of the selected group.
  - **role** - Associates access with users that have been assigned the selected role.

Click **Load** to load the ACLs. After you have loaded the ACLs, you can refresh the table at any time by clicking **Refresh**. The timestamp below the table records when the table was last refreshed.

**Viewing access rights:** You can view the access rights for a specific effective ACL by selecting it and clicking **View**. The **View access rights** panel opens.

- The **Subject DN** section displays the distinguished name of the entry that you are viewing.
- The **Subject type** section displays the type of ACL that the entry is associated with.
- The **Rights** section displays the addition and deletion rights of the subject.
  - **Add child** grants or denies the subject the right to add a directory entry beneath the selected entry.
  - **Delete entry** grants or denies the subject the right to delete the selected entry.
- The **Security class access rights** section defines permissions for security classes. Attributes are grouped into security classes:
  - **Normal** - Normal attributes require the least security, for example, the attribute `commonName`.
  - **Sensitive** - Sensitive attributes require a moderate amount of security, for example `homePhone`.
  - **Critical** - Critical attributes require the most security, for example, the attribute `userpassword`.
  - **System** - System attributes are read only attributes that are maintained by the server.
  - **Restricted** - Restricted attributes are used to define access control.

You can view the attribute to determine its security class. See “Viewing attributes” on page 43 if you need information about how to do this.

**Note:** The system and restricted security class options are displayed only if your server supports system and restricted ACLs. The system security class cannot be set to writable.

- The Attribute access rights section lists attributes that have had their permissions individually set, instead of using those set for security class to which the attribute belongs.
  - **Read** - The subject can read attributes.
  - **Write** - The subject can modify the attributes.

**Note:** System class is not writable.

- **Search** - The subject can search attributes.
- **Compare** - The subject can compare attributes.
- Click **Close** to return to the Effective ACL panel.

### Effective owners

Effective owners are the explicit and inherited owners of the selected entry. The Effective owner table contains read-only information about Subject DN and the Subject type of the effective owners.

### Non-filtered ACLs

You can add new non-filtered ACLs to an entry, or edit existing non-filtered ACLs.

Non-filtered ACLs can be propagated. This means that access control information defined for one entry can be applied to all of its subordinate entries. The ACL source is the source of current ACL for the selected entry. If the entry does not have an ACL, it inherits an ACL from parent objects based on the ACL settings of the parent objects.

If no ACL applies to a directory object either directly or through inheritance, the following default access is applied:

```
aclentry:group:CN=ANYBODY:normal:rsc:system:rsc:restricted:rsc.
```

### Adding or editing non-filtered ACLs:

1. Select the **Non-filtered ACLs** tab.

**Note:** If no non-filtered ACLs exist for the entry, the Propagate ACLs check box is preselected and cannot be modified.

2. Select the **Propagate** check box to allow descendants without an explicitly defined ACL to inherit from this entry. If the check box is selected, the descendent inherits ACLs from this entry and if the ACL is explicitly defined for the child entry, then the ACL which was inherited from parent is replaced with the new ACL that was added. If the check box is not selected, descendant entries without an explicitly defined ACL will inherit ACLs from a parent of this entry that has this option enabled.
3. Click **Add** to create new access rights for the entry or select an existing Subject DN and click **Edit** to modify existing ACLs.
  - Specify **Subject DN** - Type the DN of the entity requesting access to perform operations on the selected entry, for example, cn=Ricardo Garcia,ou=austin,o=ibm,c=us. You cannot modify this field if you are editing the ACL.
  - Specify the **Subject type** - Select the type of ACL. For example, select access-id if the DN is a user. You cannot modify this field if you are editing the ACL.

- From the **Add child** menu, select whether to grant or deny the subject the right to add a directory entry beneath the selected entry. In this example, if you select grant, Ricardo Garcia is able to add child entries under ou=Widget Division.
- From the **Delete entry** menu, select whether to grant or deny the subject the right to delete the selected entry. In this example, it grants or denies cn=Ricardo Garcia the ability to delete ou=Widget Division and any of its child entries.
- Set the permissions for the **Security class access rights** for each of the security classes. You can grant the permissions individually or click **Grant all** or **Deny all** to grant or deny permissions globally. Ricardo Garcia is given the permissions you set here to all of the attributes of each security class. See “Viewing access rights” on page 319 for more information.

**Note:** If you select **Grant all**, it gives Ricardo Garcia access to the restricted attributes including the ACLs themselves. This means that Ricardo Garcia can grant himself additional permissions on the entry. For example, if the administrator denied **Delete entry** permission to Ricardo Garcia on the entry ou=Widget Division,ou=austin,o=ibm,c=us, Ricardo Garcia could not delete the entry or any of its child entries. If the administrator also clicked **Grant All** for the security class permissions, Ricardo Garcia is able to change the ACL and can give himself permission to delete the child entries of ou=Widget Division,ou=austin,o=ibm,c=us and the parent entry itself. If you do select **Grant All** when creating ACLs, you might want to explicitly deny write permission to the restricted class for security purposes.

- Additionally, you may specify permissions based on the attribute instead of the security class to which the attribute belongs.
  - Select an attribute from the **Define an attribute** drop-down list.
  - Click **Define**. The attribute is displayed with a permissions table.
  - Specify whether to grant or deny each of the four security class permissions associated with the attribute or click **Grant all** or **Deny all** to grant or deny permissions globally .
  - You can repeat this procedure for multiple attributes.
  - To remove an attribute, simply select the attribute and click **Delete**.
  - When you are finished click **OK** to return to the Edit ACL Panel.
- Click **OK** to save your changes and exit.

**Removing ACLs non-filtered ACLs:** To remove non-filtered ACLs:

- Select the **Non-filtered** ACLs tab
- Select the radio button next to the ACL you want to delete.
- Click **Remove** or click **Remove all** to delete all Subject DN's from the list.
- Click **OK** to save your changes.

### Filtered ACLs

You can add new filtered ACLs to an entry, or edit existing filtered ACLs.

Filter-based ACLs employ a filter-based comparison, using a specified object filter, to match target objects with the effective access that applies to them.

The default behavior of filter-based ACLs to accumulate from the lowest containing entry, upward along the ancestor entry chain, to the highest containing

entry in the DIT. The effective access is calculated as the union of the access rights granted, or denied, by the constituent ancestor entries. There is an exception to this behavior. For compatibility with the subtree replication feature, and to allow greater administrative control, a ceiling attribute is used as a means to stop accumulation at the entry in which it is contained.

If no ACL applies to a directory object either directly or through inheritance, the following default access is applied:

```
ibm-filteraclentry:group:CN=ANYBODY:(objectclass=*)normal:rsc:system:rsc
:restricted:rsc
```

### Adding or editing filtered ACLs:

1. Select the **Filtered ACLs** tab
2. Enter the following information on the Filtered ACLs tab:
  - Select the **Not specified** radio button to remove the `ibm-filterACLInherit` attribute from the selected entry.
  - Select the **True** radio button to allow the ACLs for the selected entry to accumulate from that entry, upward along the ancestor entry chain, to the highest filter ACL containing entry in the DIT.
  - Select the **False** radio button to stop the accumulation of filter ACLs at the selected entry.
3. Click **Add** to create new access rights for the entry or select an existing Subject DN and click **Edit** to modify existing filtered ACLs.
  - Specify **Subject DN** - Type the DN of the entity requesting access to perform operations on the selected entry, for example, `cn=Ricardo Garcia,ou=austin,o=ibm,c=us`. You cannot modify this field if you are editing the ACL.
  - Specify the **Subject type** - Select the type of ACL. For example, select `access-id` if the DN is a user. You cannot modify this field if you are editing the ACL.
  - From the **Add child** menu, select whether to grant or deny the subject the right to add a directory entry beneath the selected entry. In this example, if you select grant, Ricardo Garcia is able to add child entries under `ou=Widget Division`.
  - From the **Delete entry** menu, select whether to grant or deny the subject the right to delete the selected entry. In this example, it grants or denies `cn=Ricardo Garcia` the ability to delete `ou=Widget Division` and any of its child entries.
  - Specify the filter for the selected ACL in the **Object filter** field. The ACL propagates to any descendant object in the associated subtree that matches the filter that you specified in this field. For example, if you specify `sn=Campbell` as the filter, then Ricardo Garcia has access permissions under `ou=Widget Division,ou=austin,o=ibm,c=us` to the entries `cn=David Campbell`, `cn=James Campbell`, `cn=Michael Campbell+postalcode=4609` and `cn=Michael Campbell` because each of the entries contain the `sn` attribute with the value `Campbell`. Click **Edit filter** for assistance in composing the search filter string.
  - Set the permissions for the **Security class access rights** for each of the security classes. You can grant the permissions individually or click **Grant all** or **Deny all** to grant or deny permissions globally. Ricardo Garcia is given the permissions you set here to all of the attributes of each security class. See “Viewing access rights” on page 319 for more information.



**Note:** If you select **Grant all**, it gives Ricardo Garcia access to the restricted attributes including the ACLs themselves. This means that Ricardo Garcia can grant himself additional permissions on the entry. For example, if the administrator denied **Delete entry** permission to Ricardo Garcia on the entry `ou=Widget Division,ou=austin,o=ibm,c=us`, Ricardo Garcia could not delete the entry or any of its child entries. If the administrator also clicked **Grant All** for the security class permissions, Ricardo Garcia is able to change the ACL and can give himself permission to delete the child entries of `ou=Widget Division,ou=austin,o=ibm,c=us` and the parent entry itself. If you do select **Grant All** when creating ACLs, you might want to explicitly deny write permission to the restricted class for security purposes.

- Additionally, you may specify permissions based on the attribute instead of the security class to which the attribute belongs.
    - Select an attribute from the **Define an attribute** drop-down list.
    - Click **Define**. The attribute is displayed with a permissions table.
    - Specify whether to grant or deny each of the four security class permissions associated with the attribute or click **Grant all** or **Deny all** to grant or deny permissions globally .
    - You can repeat this procedure for multiple attributes.
    - To remove an attribute, simply select the attribute and click **Delete**.
    - When you are finished click **OK** to return to the Edit ACL Panel..
4. Click **OK** to save your changes and exit.

**Removing filtered ACLs:** To remove filtered ACLs:

- Select the **Filtered** ACLs tab
- Select the radio button next to the ACL you want to delete.
- Click **Remove** or click **Remove all** to delete all Subject DNs from the list.
- Click **OK** to save your changes.

## Owners

Entry owners have complete permissions to perform any operation on an object. Entry owners can be explicit or propagated (inherited). The owner is the source of the current owner for the selected entry. If the entry does not inherit an owner from a ancestor, this field displays a message stating that this entry inherits owners from default. Adding owners to this entry overrides all inherited owners. By default the directory administrator is the owner of all of the entries in the directory.

**Adding an owner:** To add an owner for the entry:

1. Select the **Owners** tab.
  - Select the **Propagate owners** check box to allow descendants without an explicitly defined owner to inherit from this entry. If the check box is not selected, descendant entries without an explicitly defined owner will inherit owner from a parent of this entry that has this option enabled.
  - Specify the **Subject DN**. Type the (DN) Distinguished name of the entity that you are granting owner access on the selected entry, for example, `cn=Ricardo Garcia,ou=austin,o=ibm,c=us`.
  - Select the **Subject type** of DN. For example, select `access-id` if the DN is a user.
2. Click **Add**.

3. Repeat the process for any additional owners that you want to create.
4. When you are finished, click **OK** to save your changes and exit to the **Manage entries** panel.

**Removing an owner:** To remove an owner from an entry:

1. Select the **Owners** tab.
2. Select the radio button next to the owner you want to delete.
3. Click **Remove** or click **Remove all** to delete all Subject DNs from the list.
4. Click **OK** to save your changes.

## Using the command line utilities to manage ACLs

The following sections describe how to use the LDIF utilities to manage ACLs

### Defining the ACLs and entry owners

The following two examples show an administrative subdomain being established. The first example shows a single user being assigned as the entryOwner for the entire domain. The second example shows a group assigned as the entryOwner.

```
entryOwner: access-id:cn=Person A,o=IBM
ownerPropagate: true
```

```
entryOwner: group:cn=System Owners, o=IBM
ownerPropagate: true
```

The next example shows how an access ID "cn=Person 1, o=IBM" is being given permissions to read, search, and compare attribute1. The permission applies to any node in the entire subtree, at or below the node containing this ACL, that matches the "(objectclass=groupOfNames)" comparison filter. The accumulation of matching ibm-filteraclentry attributes in any ancestor nodes has been terminated at this entry by setting the ibm-filterAclInherit attribute to "false".

```
ibm-filterAclEntry: access-id:cn=Person 1,o=IBM:(objectclass=groupOfNames):
 at.attribute1:grant:rsc
```

```
ibm-filterAclInherit: false
```

The next example shows how a group "cn=Dept XYZ, o=IBM" is being given permissions to read, search and compare attribute1. The permission applies to the entire subtree below the node containing this ACL.

```
aclEntry: group:cn=Dept XYZ,o=IBM:at.attribute1:grant:rsc
aclPropagate: true
```

The next example shows how a role "cn=System Admins,o=IBM" is being given permissions to add objects below this node, and read, search and compare attribute2 and the critical attribute class. The permission applies only to the node containing this ACL.

```
aclEntry: role:cn=System Admins,o=IBM:object:grant:a:at.
 attribute2:grant:rsc:critical:grant:rsc
aclPropagate: false
```

### Modifying the ACL and entry owner values

#### Modify-replace

Modify-replace works the same way as all other attributes. If the attribute value does not exist, create the value. If the attribute value exists, replace the value.

Given the following ACLs for an entry:

```
aclEntry: group:cn=Dept ABC,o=IBM:normal:grant:rsc
aclPropagate: true
```

perform the following change:

```
dn: cn=some entry
changetype: modify
replace: aclEntry
aclEntry: group:cn=Dept XYZ,o=IBM:normal:grant:rsc
```

The resulting ACI is:

```
aclEntry: group:cn=Dept XYZ,o=IBM:normal:grant:rsc
aclPropagate: true
```

ACI values for Dept ABC are lost through the replace.

Given the following ACIs for an entry:

```
ibm-filterAclEntry: group:cn=Dept ABC,o=IBM:(cn=Manager ABC):normal
 :grant:rsc
ibm-filterAclInherit: true
```

perform the following changes:

```
dn: cn=some entry
changetype: modify
replace: ibm-filterAclEntry
ibm-filterAclEntry: group:cn=Dept XYZ,o=IBM:(cn=Manager XYZ):normal
 :grant:rsc
```

```
dn: cn=some entry
changetype: modify
replace: ibm-filterAclInherit
ibm-filterAclInherit: false
```

The resulting ACI is:

```
ibm-filterAclEntry: group:cn=Dept XYZ,o=IBM:(cn=Manager XYZ):normal
 :grant:rsc
ibm-filterAclInherit: false
```

ACI values for Dept ABC are lost through the replace.

### **Modify-add**

During an `idsldapmodify-add`, if the ACI or `entryOwner` does not exist, the ACI or `entryOwner` with the specific values is created. If the ACI or `entryOwner` exists, then add the specified values to the given ACI or `entryOwner`. For example, given the ACI:

```
aclEntry: group:cn=Dept XYZ,o=IBM:normal:grant:rsc
```

with a modification:

```
dn: cn=some entry
changetype: modify
add: aclEntry
aclEntry: group:cn=Dept ABC,o=IBM:at.attribute1:grant:rsc
```

would yield an multi-valued `aclEntry` of:

```
aclEntry: group:cn=Dept XYZ,o=IBM:normal:grant:rsc
aclEntry: group:cn=Dept ABC,o=IBM:at.attribute1:grant:rsc
```

For example, given the ACI:

```
Ibm-filterAclEntry: group:cn=Dept XYZ,o=IBM:(cn=Manager XYZ):normal
 :grant:rsc
```

with a modification:

```
dn: cn=some entry
changetype: modify
add: ibm-filterAclEntry
ibm-filterAclEntry: group:cn=Dept ABC,o=IBM:(cn=Manager ABC)
:at.attribute1:grant:rsc
```

would yield an multi-valued aclEntry of:

```
Ibm-filterAclEntry: group:cn=Dept XYZ,o=IBM:(cn=Manager XYZ):normal
:grant:rsc
ibm-filterAclEntry: group:cn=Dept ABC,o=IBM:(cn=Manager ABC):at.attribute1
:grant:rsc
```

The permissions under the same attribute or attribute class are considered as the basic building blocks and the actions are considered as the qualifiers. If the same permission value is being added more than once, only one value is stored. If the same permission value is being added more than once with different action values, the last action value is used. If the resulting permission field is empty (""), this permission value is set to null and the action value is set to **grant**.

For example, given the following ACI:

```
aclEntry: group:cn=Dept XYZ,o=IBM:normal:grant:rsc
```

with a modification:

```
dn: cn=some entry
changetype: modify
add: aclEntry
aclEntry: group:cn=Dept XYZ,o=IBM:normal:deny:r:critical:deny::sensitive
:grant:r
```

yields an aclEntry of:

```
aclEntry: group:cn=Dept XYZ,o=IBM:normal:grant:sc:normal:deny:r:critical
:grant::sensitive:grant:r
```

For example, given the following ACI:

```
Ibm-filterAclEntry: group:cn=Dept XYZ,o=IBM:(cn=Manager XYZ):normal
:grant:rsc
```

with a modification:

```
dn: cn=some entry
changetype: modify
add: ibm-filterAclEntry
ibm-filterAclEntry: group:cn=Dept XYZ,o=IBM:(cn=Manager XYZ):normal
:deny:r:critical:deny::sensitive:grant:r
```

yields an aclEntry of:

```
ibm-filterAclEntry: group:cn=Dept XYZ,o=IBM:(cn=Manager XYZ):normal
:grant:sc:normal:deny:r:critical:grant::sensitive
:grant:r
```

### Modify-delete

To delete a particular ACI value, use the regular `idsldapmodify-delete` syntax.

Given an ACI of:

```
aclEntry: group:cn=Dept XYZ,o=IBM:object:grant:ad
aclEntry: group:cn=Dept XYZ,o=IBM:normal:grant:rswc
```

```
dn: cn = some entry
changetype: modify
delete: aclEntry
aclEntry: group:cn=Dept XYZ,o=IBM:object:grant:ad
```

yields a remaining ACI on the server of :

```
aclEntry: group:cn=Dept XYZ,o=IBM:normal:grant:rws
```

Given an ACI of:

```
ibm-filterAclEntry: group:cn=Dept XYZ,o=IBM:(cn=Manager XYZ):object
:grant:ad
ibm-filterAclEntry: group:cn=Dept XYZ,o=IBM:(cn=Manager XYZ):normal
:grant:rws
```

```
dn: cn = some entry
changetype: modify
delete: ibm-filterAclEntry
ibm-filterAclEntry: group:cn=Dept XYZ,o=IBM:(cn=Manager XYZ):object
:grant:ad
```

yields a remaining ACI on the server of:

```
ibm-filterAclEntry: group:cn=Dept XYZ,o=IBM:(cn=Manager XYZ):normal
:grant:rws
```

Deleting an ACI or entryOwner value that does not exist results in an unchanged ACI or entryOwner and a return code specifying that the attribute value does not exist.

## Deleting the ACI/entry owner values

With the `idsldapmodify-delete` operation, the entryOwner can be deleted by specifying

```
dn: cn = some entry
changetype: modify
delete: entryOwner
```

In this case, the entry would then have no explicit entryOwner. The ownerPropagate is also removed automatically. This entry would inherit its entryOwner from the ancestor node in the directory tree following the propagation rule.

The same can be done to delete aclEntry completely:

```
dn: cn = some entry
changetype: modify
delete: aclEntry
```

Deleting the last ACI or entryOwner value from an entry is not the same as deleting the ACI or entryOwner. It is possible for an entry to contain an ACI or entryOwner with no values. In this case, nothing is returned to the client when querying the ACI or entryOwner and the setting propagates to the descendent nodes until it is overridden. To prevent dangling entries that nobody can access, the directory administrator always has full access to an entry even if the entry has a null ACI or entryOwner value.

## Retrieving the ACI/entry owner values

The effective ACI or entryOwner values can be retrieved by simply specifying the desired ACL or entryOwner attributes in a search, for example,

```
idsldapsearch -b "cn=object A, o=ibm" -s base "objectclass=*"
aclentry aclpropagate aclsource entryowner ownerpropagate ownersource
ibm-filterAclEntry ibm-filterAclInherit ibm-effectiveAcl
```

returns all ACL or entryOwner information that is used in access evaluation on object A. Note that the returned values might not look exactly the same as they are first defined. The values are the equivalent of the original form.

Searching on the `ibm-filterAclEntry` attribute alone only returns the values specific to the containing entry.

A read-only operational attribute, `ibm-effectiveAcl`, is used to show the accumulated effective access. A search request for `ibm-effectiveAcl` returns the effective access that applies to the target object based on: non-filter ACLs, or filter ACLs, depending on how they have been distributed in the DIT.

Because filter-based ACLs might come from several ancestor sources, a search on the `aclSource` attribute produces a list of the associated sources.

---

## Subtree replication considerations

For non-filter-based access to be included in subtree replication, any `aclEntry` attributes must reside at the associated `ibm-replicationContext` entry. Because effective access cannot be propagated from an ancestor entry above a replicated subtree, the `aclPropagate` attribute must be set to a value of **true**.

For filter-based access to be included in subtree replication, any `ibm-filterAclEntry` attributes must reside at, or below, the associated `ibm-replicationContext` entry. Because effective access cannot be accumulated from an ancestor entry above a replicated subtree, the `ibm-filterAclInherit` attribute must be set to a value of **false**, and reside at the associated `ibm-replicationContext` entry.

---

## Chapter 18. Groups and roles

---

### Groups

A group is a list, such as a collection of names. A group can be used in **aclentry**, **ibm-filterAclEntry**, and **entryowner** attributes to control access or in application-specific uses such as a mailing list; see Chapter 17, “Access control lists,” on page 309. Groups can be defined as either static, dynamic, or nested.

#### Static groups

A static group defines each member individually using the structural objectclass **groupOfNames**, **groupOfUniqueNames**, **accessGroup**, or **accessRole**; or the auxiliary objectclass **ibm-staticgroup** or **ibm-globalAdminGroup**. A static group using the structural objectclasses **groupOfNames** and **groupOfUniqueNames** require at least one member or **uniqueMember**, respectively.

The IBM Tivoli Directory Server enforces partial referential integrity for static groups. Referential integrity is a database concept that ensures relationships between tables remain consistent. When a static group is added into the directory, the members need not exist in the directory. However, when an object is deleted from the directory, all static groups that have this object as a member are updated automatically to remove this object from their lists of members. In addition, when an object is renamed in the directory, all static groups and nested groups that have this object as a member are updated automatically to rename this object in their lists of members.

**Note:** This concept does not apply to dynamic groups because dynamic groups are search-based. The deletion of an object from the directory automatically causes it to be excluded from the search results.

A typical group entry is:

```
DN: cn=Dev.Staff,ou=Austin,c=US
objectclass: accessGroup
cn: Dev.Staff
member: cn=John Doe,o=IBM,c=US
member: cn=Jane Smith,o=IBM,c=US
member: cn=James Smith,o=IBM,c=US
```

Each group object contains a multivalued attribute consisting of member DNs.

Upon deletion of an access group, the access group is also deleted from all ACLs to which it has been applied.

#### Dynamic groups

A dynamic group defines its members differently than a static group. Instead of listing them individually, the dynamic group defines its members using an LDAP search. The dynamic group uses the structural objectclass **groupOfURLs** (or auxiliary objectclass **ibm-dynamicGroup**) and the attribute, **memberURL** to define the search using a simplified LDAP URL syntax.

```
ldap:///<base DN of search> ?? <scope of search> ? <searchfilter>
```

**Note:** As the example illustrates, the host name must not be present in the syntax. The remaining parameters are just like normal ldap URL syntax. Each

parameter field must be separated by a ?, even if no parameter is specified. Normally, a list of attributes to return would be included between the base DN and scope of the search. This parameter is also not used by the server when determining dynamic membership, and so may be omitted, however, the separator ? must still be present.

where:

#### **base DN of search**

Is the point from which the search begins in the directory. It can be the suffix or root of the directory such as **ou=Austin**. This parameter is required.

#### **scope of search**

Specifies the extent of the search. The default scope is sub.

- base** Returns information only about the base DN specified in the URL
- one** Returns information about entries one level below the base DN specified in the URL. It does not include the base entry.
- sub** Returns information about entries at all levels below and includes the base DN.

#### **searchfilter**

Is the filter that you want to apply to the entries within the scope of the search. See “the ldapsearch filter option” on page 411 for information about the syntax of the searchfilter. The default is objectclass=\*

The search for dynamic members is always internal to the server, so unlike a full ldap URL, a host name and port number is never specified, and the protocol is always **ldap** (never **ldaps**). The **memberURL** attribute may contain any kind of URL, but the server only uses **memberURLs** beginning with **ldap:///** to determine dynamic membership.

### **Examples**

A single entry in which the scope defaults to sub and the filter defaults to objectclass=\*

```
ldap:///cn=John Doe, cn=Employees, o=Acme, c=US
```

All entries that are one-level below cn=Employees, and the filter defaults to objectclass=\*

```
ldap:///cn=Employees, o=Acme, c=US??one
```

All entries that are under o=Acme with the objectclass=person:

```
ldap:///o=Acme, c=US??sub?objectclass=person
```

Depending on the object classes you use to define user entries, those entries might not contain attributes which are appropriate for determining group membership. You can use the auxiliary object class, **ibm-dynamicMember**, to extend your user entries to include the **ibm-group** attribute. This attribute allows you to add arbitrary values to your user entries to serve as targets for the filters of your dynamic groups. For example:

The members of this dynamic group are entries directly under the cn=users,ou=Austin entry that have an ibm-group attribute of GROUP1:

```
dn: cn=GROUP1,ou=Austin
 objectclass: groupOfURLs
 cn: GROUP1
 memberURL: ldap:///cn=users,ou=Austin??one?(ibm-group=GROUP1)
```



Here is an example member of cn=GROUP1,ou=Austin:

```
dn: cn=Group 1 member, cn=users, ou=austin
objectclass: person
objectclass: ibm-dynamicMember
cn: Group 1 member
sn: member
userpassword: memberpassword
ibm-group: GROUP1
```

## Nested groups

The nesting of groups enables the creation of hierarchical relationships that can be used to define inherited group membership. A nested group is defined as a parent group entry that has members that are group entries. A nested group is created by extending one of the structural group object classes by adding the **ibm-nestedGroup** auxiliary object class. After nested group extension, zero or more **ibm-memberGroup** attributes may be added, with their values set to the DNs of nested child groups. For example:

```
dn: cn=Group 2, cn=Groups, o=IBM, c=US
objectclass: groupOfNames
objectclass: ibm-nestedGroup
objectclass: top
cn: Group 2
description: Group composed of static, and nested members.
member: cn=Person 2.1, cn=Dept 2, cn=Employees, o=IBM, c=US
member: cn=Person 2.2, cn=Dept 2, cn=Employees, o=IBM, c=US
ibm-memberGroup: cn=Group 8, cn=Nested Static, cn=Groups, o=IBM, c=US
```

The introduction of cycles into the nested group hierarchy is not allowed. If it is determined that a nested group operation results in a cyclical reference, either directly or through inheritance, it is considered a constraint violation and therefore, the update to the entry fails.

## Hybrid groups

Any of the structural group object classes mentioned can be extended such that group membership is described by a combination of static, dynamic, and nested member types. For example:

```
dn: cn=Group 10, cn=Groups, o=IBM, c=US
objectclass: groupOfURLs
objectclass: ibm-nestedGroup
objectclass: ibm-staticGroup
objectclass: top
cn: Group 10
description: Group composed of static, dynamic, and nested members.
memberURL: ldap:///cn=Austin, cn=Employees, o=IBM, c=US??one?objectClass=person
ibm-memberGroup: cn=Group 9, cn=Nested Dynamic, cn=Groups, o=IBM, c=US
member: cn=Person 10.1, cn=Dept 2, cn=Employees, o=IBM, c=US
member: cn=Person 10.2, cn=Dept 2, cn=Employees, o=IBM, c=US
```

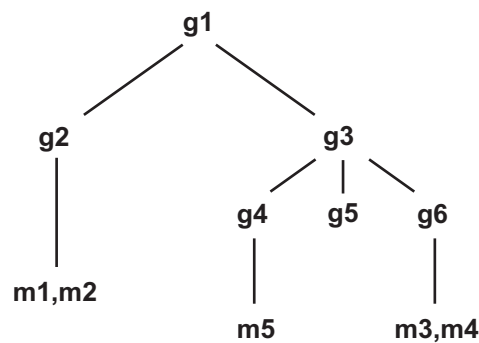
## Determining group membership

Two operational attributes can be used to query aggregate group membership. For a given group entry, the **ibm-allMembers** operational attribute enumerates the aggregate set of group membership, including static, dynamic, and nested members, as described by the nested group hierarchy. For a given user entry, the **ibm-allGroups** operational attribute enumerates the aggregate set of groups, including ancestor groups, to which that user has membership.

**Note:** The values for the **ibm-allMembers** and **ibm-allGroups** operational attributes are determined at runtime. For a large directory, this can mean long operation times.

A requester may only receive a subset of the total data requested, depending on how the ACLs have been set on the data. Anyone can request the **ibm-allMembers** and **ibm-allGroups** operational attributes, but the data set returned only contains data for the LDAP entries and attributes that the requester has access rights to. The user requesting the **ibm-allMembers** or **ibm-allGroups** attribute must have access to the **member** or **uniquemember** attribute values for the group and nested groups in order to see static members, and must be able to perform the searches specified in the **memberURL** attribute values in order to see dynamic members. For examples:

### Hierarchy examples



For this example, **m1** and **m2** are in the **member** attribute of **g2**. The ACL for **g2** allows **user1** to read the member attribute, but **user 2** does not have access to the member attribute. The entry LDIF for the **g2** entry is as follows:

```

dn: cn=g2,cn=groups,o=ibm,c=us
objectclass: accessGroup
cn: g2
member: cn=m1,cn=users,o=ibm,c=us
member: cn=m2,cn=users,o=ibm,c=us
aclentry: access-id:cn=user1,cn=users,o=ibm,c=us:normal:rsc
aclentry: access-id:cn=user2,cn=users,o=ibm,c=us:normal:rsc:at.member:deny:rsc

```

The **g4** entry uses the default **aclentry**, which allows both **user1** and **user2** to read its member attribute. The LDIF for the **g4** entry is as follows:

```

dn: cn=g4, cn=groups,o=ibm,c=us
objectclass: accessGroup
cn: g4
member: cn=m5, cn=users,o=ibm,c=us

```

The **g5** entry is a dynamic group, which gets its two members from the **memberURL** attribute. The LDIF for the **g5** entry is as follows:

```

dn: cn=g5, cn=groups,o=ibm,c=us
objectclass: container
objectclass: ibm-dynamicGroup
cn: g5
memberURL: ldap:///cn=users,o=ibm,c=us??sub?(|(cn=m3)(cn=m4))

```

The entries **m3** and **m4** are members of group **g5** because they match the **memberURL**. The ACL for the **m3** entry allows both **user1** and **user2** to search for it. The ACL for the **m4** entries doesn't allow **user2** to search for it. The LDIF for **m4** is as follows:

```
dn: cn=m3, cn=users,o=ibm,c=us
objectclass:person
cn: m3
sn: three
aclentry: access-id:cn=user1,cn=users,o=ibm,c=us:normal:rsc
aclentry: access-id:cn=user2,cn=users,o=ibm,c=us:normal:rsc
```

```
dn: cn=m4, cn=users,o=ibm,c=us
objectclass:person
cn: m4
sn: four
aclentry: access-id:cn=user1,cn=users,o=ibm,c=us:normal:rsc
aclentry: access-id:cn=user2,cn=users,o=ibm,c=us
```

#### Example 1:

User 1 does a search to get all the members of group **g1**. User 1 has access to all members, so they are all returned.

```
idsldapsearch -D cn=user1,cn=users,o=ibm,c=us -w user1pwd -s base -b cn=g1,
cn=groups,o=ibm,c=us objectclass=* ibm-allmembers
```

```
cn=g1,cn=groups,o=ibm,c=us
ibm-allmembers: CN=M1,CN=USERS,O=IBM,C=US
ibm-allmembers: CN=M2,CN=USERS,O=IBM,C=US
ibm-allmembers: CN=M3,CN=USERS,O=IBM,C=US
ibm-allmembers: CN=M4,CN=USERS,O=IBM,C=US
ibm-allmembers: CN=M5,CN=USERS,O=IBM,C=US
```

#### Example 2:

User 2 does a search to get all the members of group **g1**. User 2 does not have access to members **m1** or **m2** because they do not have access to the member attribute for group **g2**. User 2 has access to the member attribute for **g4** and therefore has access to member **m5**. User 2 can perform the search in the group **g5** memberURL for entry **m3**, so that member are listed, but cannot perform the search for **m4**.

```
idsldapsearch -D cn=user2,cn=users,o=ibm,c=us -w user2pwd -s base -b cn=g1,
cn=groups,o=ibm,c=us objectclass=* ibm-allmembers
```

```
cn=g1,cn=groups,o=ibm,c=us
ibm-allmembers: CN=M3,CN=USERS,O=IBM,C=US
ibm-allmembers: CN=M5,CN=USERS,O=IBM,C=US
```

#### Example 3:

User 2 does a search to see if **m3** is a member of group **g1**. User 2 has access to do this search, so the search shows that **m3** is a member of group **g1**.

```
idsldapsearch -D cn=user2,cn=users,o=ibm,c=us -w user2pwd -s base -b cn=m3,
cn=users,o=ibm,c=us objectclass=* ibm-allgroups
```

```
cn=m3,cn=users,o=ibm,c=us
ibm-allgroups: CN=G1,CN=GROUPS,O=IBM,C=US
```

#### Example 4:

User 2 does a search to see if **m1** is a member of group **g1**. User 2 does not have access to the member attribute, so the search does not show that **m1** is a member of group **g1**.

```
idsldapsearch -D cn=user2,cn=users,o=ibm,c=us -w user2pwd -s base -b
cn=m1,cn=users,o=ibm,c=us objectclass=* ibm-allgroups
```

```
cn=m1,cn=users,o=ibm,c=us
```

## Group object classes

### **ibm-dynamicGroup**

This auxiliary class allows the optional **memberURL** attribute. Use it with a structural class such as **groupOfNames** to create a hybrid group with both static and dynamic members.

### **ibm-dynamicMember**

This auxiliary class allows the optional **ibm-group** attribute. Use it as a filter attribute for dynamic groups.

### **ibm-nestedGroup**

This auxiliary class allows the optional **ibm-memberGroup** attribute. Use it with a structural class such as **groupOfNames** to enable sub-groups to be nested within the parent group.

### **ibm-staticGroup**

This auxiliary class allows the optional **member** attribute. Use it with a structural class such as **groupOfURLs** to create a hybrid group with both static and dynamic members.

**Note:** The **ibm-staticGroup** is the only class for which **member** is *optional*, all other classes taking **member** require at least 1 member.

### **groupOfNames**

Defines entries for a group of names. Represents a list containing an unordered list of names.

### **groupOfUniqueNames**

Defines entries for a group of unique names.

### **accessGroup**

A group that is used for access control.

### **groupOfURLs**

Represents a group of URLs.

## Group attribute types

### **ibm-allGroups**

Shows all groups to which an entry belongs. An entry can be a member directly by the **member**, **uniqueMember**, or **memberURL** attributes, or indirectly by the **ibm-memberGroup** attribute. This **Read-only** operational attribute is not allowed in a search filter.

### **ibm-allMembers**

Shows all members of a group. An entry can be a member directly by the **member**, **uniqueMember**, or **memberURL** attributes, or indirectly by the **ibm-memberGroup** attribute. This **Read-only** operational attribute is not allowed in a search filter.

### **ibm-group**

Is an attribute taken by the auxiliary class **ibm-dynamicMember**. Use it to define arbitrary values to control membership of the entry in dynamic groups. For example, add the value "Bowling Team" to include the entry in any **memberURL** that has the filter "ibm-group=Bowling Team".

### **ibm-memberGroup**

Is an attribute taken by the auxiliary class **ibm-nestedGroup**. It identifies sub-groups of a parent group entry. Members of all such sub-groups are considered members of the parent group when processing ACLs or the

**ibm-allMembers** and **ibm-allGroups** operational attributes. The sub-group entries themselves are *not* members. Nested membership is recursive.

**member**

Identifies the distinguished names for each member of the group.

**uniquemember**

Identifies a group of names associated with an entry where each name was given a uniqueIdentifier to ensure its uniqueness. A value for the uniqueMember attribute is a DN followed by the uniqueIdentifier.

**memberURL**

Identifies an URL associated with each member of a group. Any type of labeled URL can be used.

The following tasks utilize the entries contained in the sample.ldif file that is located in the **examples** directory of the IBM Tivoli Directory Server.

You are going to create three groups to organize a lunch club. The first group is a static group that lists those people who like to meet for lunch on Monday. The second group that meets for lunch on Tuesday is a dynamic group. This group lists all the members of a department (the Widget division). The advantage of a dynamic group is that the changes that you make to the subtree entry, such as adding a new person entry, is dynamically changed in the group as well. The third group is a nested group that is a container for the other two groups.

---

## Creating a static group entry

If you have not done so already, expand the **Directory management** category in the navigation area.

1. Click **Add an entry**.
2. Select the **Groups** filter object class from the drop-down menu and click **Refresh**.
3. Select one **Structural object class** from the list box. For this example **GroupOfNames**.
4. Click **Next**.
5. Select the **Groups** filter object class from the drop-down menu and click **Refresh**.
6. Select any **Auxiliary object classes** you wish to use from the Available box. For this example **ibm-staticGroup** and click **Add**. Repeat this process for each auxiliary object class you want to add. You can also delete an auxiliary object class from the Selected box by selecting it and clicking **Remove**.
7. Click **Next**.
8. In the **Relative DN** field, enter the relative distinguished name (RDN) of the entry that you are adding, for example, cn=Monday.
9. In the **Parent DN** field, enter the distinguished name of the tree entry you selected, for example, ou=Groups,o=ibm,c=us. You can also click **Browse** to select the Parent DN from the list. You can also expand the selection to view other choices lower in the subtree. Specify your choice and click **Select** to specify the Parent DN that you want. The **Parent DN** defaults to the entry selected in the tree.

**Note:** If you started this task from the **Manage entries** panel, this field is prefilled for you. You selected the **Parent DN** before clicking **Add** to start the add entry process.

10. At the **Required attributes** tab enter the values for the required attributes. For this example in the **cn** field type **Monday**.

**Notes:**

- a. If you want to add more than one value for a particular attribute, click **Multiple values**. Supply the additional value for the attribute and click **Add**. Repeat this for each additional value. To remove a value, select the value and click **Remove**. Click **OK** when you have finished adding the multiple values. The values are added to a drop-down menu displayed below the attribute.
  - b. If your server has language tags enabled, you can click **Language tag value** to add or remove language tag descriptors. See “Language tags” on page 297 for more information.
11. In the **member** field, add the DN for at least one member. For example `cn=Bob Garcia,ou=austin,o=ibm,c=us`.

**Note:** This member does not have to be a preexisting entry. It can be created later.

- a. Click **Multiple values**.
  - b. In the **member** field, type `cn=Ricardo Garcia,ou=austin,o=ibm,c=us`.
  - c. Click **Add**.
  - d. Click **OK**.
12. Click **Optional attributes**.
  13. At the **Optional attributes** tab enter the values as appropriate for the other attributes. For example in the **Description** field, type Monday lunch group. See “Binary data for attributes” on page 295 for information on adding binary values.
  14. Click **Finish** to create the entry.

See “Managing members of group entries” on page 339 to add additional members to this group.

---

## Creating a dynamic group entry

For this example, you are creating a dynamic group for the organization `ou=Widget Division,ou=Austin,o=ibm,c=us`.

If you have not done so already, expand the **Directory management** category in the navigation area.

1. Click **Add an entry**.
2. If not already selected, choose the **All** filter object class from the drop-down menu and click **Refresh**.
3. Select one **Structural object class** from the list box. For this example **container**.
4. Click **Next**.
5. Select the **Groups** filter object class from the drop-down menu and click **Refresh**.
6. Select any **Auxiliary object classes** you wish to use from the Available box. For this example **ibm-dynamicGroup** and click **Add**. Repeat this process for each auxiliary object class you want to add. You can also delete an auxiliary object class from the Selected box by selecting it and clicking **Remove**.
7. Click **Next**.

8. In the **Relative DN** field, enter the relative distinguished name (RDN) of the entry that you are adding, for example, cn=Tuesday.
9. In the **Parent DN** field, enter the distinguished name of the tree entry you selected, for example, ou=Groups,o=ibm,c=us. You can also click **Browse** to select the Parent DN from the list. You can also expand the selection to view other choices lower in the subtree. Specify your choice and click **Select** to specify the Parent DN that you want. The **Parent DN** defaults to the entry selected in the tree.

**Note:** If you started this task from the **Manage entries** panel, this field is prefilled for you. You selected the **Parent DN** before clicking **Add** to start the add entry process.

10. At the **Required attributes** tab enter the values for the required attributes. In this example, in the **cn** field type Tuesday.

**Notes:**

- a. If you want to add more than one value for a particular attribute, click **Multiple values**. Supply the additional value for the attribute and click **Add**. Repeat this for each additional value. To remove a value, select the value and click **Remove**. Click **OK** when you have finished adding the multiple values. The values are added to a drop-down menu displayed below the attribute.
  - b. If your server has language tags enabled, you can click **Language tag value** to add or remove language tag descriptors. See “Language tags” on page 297 for more information.
11. Click **Optional attributes**.
  12. At the **Optional attributes** tab enter the values as appropriate for the other attributes. In this example for **memberURL** type **ldap:///ou=Widget Division,ou=Austin,o=ibm,c=us??sub?**.
  13. Click **Finish** to create the entry.

---

## Creating a nested group entry

In this task you are creating a nested group that is a container for the other two groups.

If you have not done so already, expand the **Directory management** category in the navigation area.

1. Click **Add an entry**.
2. If not already selected, choose the **All** filter object class from the drop-down menu and click **Refresh**.
3. Select one **Structural object class** from the list box. For this example **container**.
4. Click **Next**.
5. Select the **Groups** filter object class from the drop-down menu and click **Refresh**.
6. Select any **Auxiliary object classes** you wish to use from the Available box. For this example **ibm-nestedGroup** and click **Add**. Repeat this process for each auxiliary object class you want to add. You can also delete an auxiliary object class from the Selected box by selecting it and clicking **Remove**.
7. Click **Next**.
8. In the **Relative DN** field, enter the relative distinguished name (RDN) of the entry that you are adding, for example, cn=Lunch bunch.

9. In the **Parent DN** field, enter the distinguished name of the tree entry you selected, for example, `ou=Groups,o=ibm,c=us`. You can also click **Browse** to select the Parent DN from the list. You can also expand the selection to view other choices lower in the subtree. Specify the your choice and click **Select** to specify the Parent DN that you want. The **Parent DN** defaults to the entry selected in the tree.

**Note:** If you started this task from the **Manage entries** panel, this field is prefilled for you. You selected the **Parent DN** before clicking **Add** to start the add entry process.

10. At the **Required attributes** tab enter the values for the required attributes. In this example, in the **cn** field type `Lunch bunch`.

**Notes:**

- a. If you want to add more than one value for a particular attribute, click **Multiple values**. Supply the additional value for the attribute and click **Add**. Repeat this for each additional value. To remove a value, select the value and click **Remove**. Click **OK** when you have finished adding the multiple values. The values are added to a drop-down menu displayed below the attribute.
  - b. If your server has language tags enabled, you can click **Language tag value** to add or remove language tag descriptors. See “Language tags” on page 297 for more information.
11. Click **Optional attributes**.
  12. At the **Optional attributes** tab enter the values as appropriate for the other attributes. In this example for **ibm-memberGroup** type `cn=Monday,ou=Groups,o=ibm,c=us`.
    - a. Click **Multiple values**.
    - b. In the **member** field, type `cn=Tuesday,ou=Groups,o=ibm,c=us`.
    - c. Click **Add**.
    - d. Click **OK**.
  13. Click **Finish** to create the entry.

---

## Verifying the group task

To verify that you created the groups in the previous tasks correctly:

If you have not done so already, expand the **Directory management** category in the navigation area.

1. Click **Manage entries**.
2. Select `o=ibm,c=us` and click **Expand**.
3. Select `ou=Groups` and click **Expand**.
4. Select `cn=Lunch bunch`.
5. Expand the **Select Action** menu, select **Manage Members** and click **Go**.

**Note:** On the Nested groups tab, `cn=monday,ou=group,o=ibm,c=us` and `cn=tuesday,ou=group,o=ibm,c=us` are listed.

6. Click the **Effective group members** tab.
7. Click **Load**. All the members of both groups are displayed.



---

## Managing members of group entries

You can add and remove members from group entries.

### Adding a member to a group entry

1. From the navigation area, expand the **Directory management** topic.
2. Click **Manage entries**.
3. Expand the various subtrees and select the group entry that you want to work on. For example, select the group `cn=Monday,ou=groups,o=ibm,c=us` that was created in the creating a static group entry task.
4. From the **Select action** drop-down menu, select **Manage members** and click **Go**.
5. The Static group members tab is highlighted. Click **Load** to display the existing members of the group. In this example `cn=Bob Garcia,ou=austin,o=ibm,c=us` and `cn=Ricardo Garcia,ou=austin,o=ibm,c=us` are displayed in the table.

#### Notes:

- a. You can add new members without clicking **Load**. This is beneficial when you have large groups.
- b. If you add new members, and one of the new members you are adding already exists, then when you click **Load**, the duplicate new member that you added is ignored.

#### Note:

6. Type the name of entry that you want to add as a member of the group for example `cn=Kyle Nguyen,ou=austin,o=ibm,c=us` in the member field or select it using the **Browse** function (Expand `o=ibm,c=us` → Expand `ou=Austin` → Select `cn=Kyle Nguyen,ou=austin,o=ibm,c=us`).
7. Click **Add**.
8. `cn=Kyle Nguyen,ou=austin,o=ibm,c=us` is displayed in the table. Click **Apply** to save the change and continue adding additional members or click **Ok** to save the changes and return to the manage entries panel. `cn=Bob Garcia,ou=austin,o=ibm,c=us`, `cn=Ricardo Garcia,ou=austin,o=ibm,c=us` and `cn=Kyle Nguyen,ou=austin,o=ibm,c=us` are now members of the Monday group.
9. If you click on the **Effective group members** tab and click **Refresh**, `cn=Bob Garcia,ou=austin,o=ibm,c=us`, `cn=Ricardo Garcia,ou=austin,o=ibm,c=us` and `cn=Kyle Nguyen,ou=austin,o=ibm,c=us` are now displayed as members.

### Removing a member from a group entry

To remove a member from the group entry:

1. From the navigation area, expand the **Directory management** topic.
2. Click **Manage entries**.
3. Expand the various subtrees and select the group entry that you want to work on. For example, select the group `cn=lunch bunch,ou=groups,o=ibm,c=us` that was created in the creating a group entry task.
4. From the **Select Action** drop-down menu, select **Manage Members** and click **Go**.
5. Select the appropriate group tab for the entry you want to remove. For this example click **Static group members**.
6. Click **Load** to show the group members.

7. Select the entry you want to remove and click **Remove**. If you want to remove all the members from the group entry, click **Remove all**.
8. You are prompted to confirm the removal. Click **OK** to remove the member.
9. Click **Apply** to save the change and continue removing additional members or click **Ok** to save the changes and return to the manage entries panel.

---

## Managing memberships for an entry

You can add and remove static memberships from an entries.

### Adding a group membership

1. From the navigation area, expand the **Directory management** topic.
2. Click **Manage entries**.
3. Expand the various subtrees and select the entry, such as `cn=Bob Garcia,ou=austin,o=ibm,c=us`.
4. From the **Select Action** drop-down menu, select **Manage Memberships** and click **Go**.
5. On the Effective memberships tab, click **Load** to display the group memberships for Bob Garcia.

**Note:** If you have selected a group entry, no effective group memberships can be displayed unless it is a member of a another static or dynamic group. No membership is displayed, if the group entry is a member of a nested group only.

6. Select the Static memberships tab.
7. Select **All suffixes** or select a suffix to limit the groups that you want to view. For this example select **cn=ibmpolicies**.
8. Click **Browse groups** to show all the static groups for that suffix.
9. Select `globalGroupName=GlobalAdminGroup,cn=ibmpolicies`.
10. Click **Select**.

**Note:** Alternatively, you could type `globalGroupName=GlobalAdminGroup,cn=ibmpolicies` in the **Group DN** field or click **Browse** to select it from the directory and click **Add**.

11. If you did not click **Load** to display the memberships for the entry or, if there were no memberships for the entry, a message is displayed: You have not loaded entries from the server. Only your changes will be displayed in the table. Do you want to continue?, click **OK**.
12. `globalGroupName=GlobalAdminGroup,cn=ibmpolicies` is displayed in the table. Click **Apply** to save the change and continue adding additional members or click **Ok** to save the changes and return to the manage entries panel. `cn=Bob Garcia,ou=austin,o=ibm,c=us` is now a member of the global administration group.
13. If you click on the **Effective group members** tab and click **Refresh**, `globalGroupName=GlobalAdminGroup,cn=ibmpolicies` is now displayed as a group membership for the entry `cn=Bob Garcia,ou=austin,o=ibm,c=us`.

### Removing a group membership from an entry

1. From the navigation area, expand the **Directory management** topic.
2. Click **Manage entries**.

3. Expand the various subtrees and select the entry, such as cn=Bob Garcia,ou=austin,o=ibm,c=us.
4. From the **Select Action** drop-down menu, select **Manage Memberships** and click **Go**.
5. On the Static memberships tab, click **Load** to display the group memberships for Bob Garcia.
6. Select the group membership that you want to remove and click **Remove**. If you want to remove all the memberships from the user entry, click **Remove all**.
7. You are prompted to confirm the removal. Click **OK** to remove the member.
8. Click **Apply** to save the change and continue removing additional members or click **Ok** to save the changes and return to the manage entries panel.

---

## Editing a memberURL in a dynamic group

To edit a memberURL in a dynamic group:

1. From the navigation area, expand the **Directory management** topic.
2. Click **Manage entries**.
3. Expand the various subtrees and select the group entry that you want to work on. For example, select the group cn=lunch bunch,ou=groups,o=ibm,c=us that was created in the creating a group entry task.

**Note:** The group entry you select must be a dynamic group.

4. From the **Select Action** drop-down menu, select **Manage Members** and click **Go**.
5. In the Dynamic group filter tab, click **Edit**.
6. You can edit the **Base DN**. The base DN is the DN on which the search is performed. You can use the **Browse** button to locate the desired DN. Clicking **Browse** takes you to the "Browse entries" panel. Select the desired entry from the table and click **Select**.
7. Select the scope for the memberURL. The options include:
  - **Object** – search only within the selected (base) entry.
  - **Single level** – search only within the immediate children of the selected (base) entry.

**Note:** This does not include the base entry.

- **Subtree** – search all descendants of the selected entry, including the base entry.
8. Enter a search filter string. You can click **Edit** to launch a panel that will help you create a search filter string. This new panel has the following options:
    - Simple
    - Advanced
    - Manual

For more information, see "Search filters" on page 305.

---

## Roles

Role-based authorization is a conceptual complement to the group-based authorization, and is useful in some cases. As a member of a role, you have the authority to do what is needed for the role in order to accomplish a job. Unlike a group, a role comes with an implicit set of permissions. There is not a built-in assumption about what permissions are gained (or lost) by being a member of a group.

Roles are similar to groups in that they are represented in the directory by an object. Additionally, roles contain a group of DNs. Roles which are to be used in access control must have an objectclass of 'AccessRole'. The 'Accessrole' objectclass is a subclass of the 'GroupOfNames' objectclass.

For example, if there are a collection of DNs such as 'sys admin', your first reaction may be to think of them as the 'sys admin group' (since groups and users are the most familiar types of privilege attributes). However, since there are a set of permissions that you would expect to receive as a member of 'sys admin' the collection of DNs may be more accurately defined as the 'sys admin role'.

---

## Chapter 19. Managing search limit groups

In the IBM Tivoli Directory Server, in order to prevent a user's search requests from consuming too many resources and consequently impairing the server's performance, search limits are imposed on these requests for any given server. The administrator sets these search limits on the size and duration of searches, when configuring the server. See "Setting Searches" on page 99 for more information.

Only the administrator and members of the local or global administrative groups are exempt from these search limits that apply to all other users. However, depending upon your needs, you can create search limit groups that can have more flexible search limits than the general user. The individual members or groups contained in the search limit group are granted the search limitations specified in the search limit group.

When a user initiates a search, the search request limitations are first checked. If a user is a member of a search limit group, the limitations are compared. If the search limit group limitations are higher than those of the search request, the search request limitations are used. If the search request limitations are higher than those of the search limit group, the search limit group limitations are used. If no search limit group entries are found, the same comparison is made against the server search limitations. If no server search limitations have been set, the comparison is made against the default server setting. The limitations used are always the lowest settings in the comparison.

If a user belongs to multiple search limit groups, the user is granted up to the highest level of search capability. For example, the user belongs to search group 1 that grants search limits of search size 2000 entries and search time of 4000 seconds and to search group 2 that grants search limits of search size unlimited entries and a search time of 3000 seconds. The user has the search limitations of search size unlimited and search time of 4000 seconds.

Search limit groups can be stored under either localhost or IBMpolicies. Search limit groups under IBMpolicies are replicated, those under localhost are not. You can store the same search limit group under both localhost and IBMpolicies. If the search limit group is not stored under one of these DNs, the server ignores the search limit part of the group and treats it as a normal group.

When a user initiates a search, the search limit group entries under localhost are checked first. If no entries are found for the user, the search limit group entries under IBMpolicies are then searched. If entries are found under localhost, the search limit group entries under IBMpolicies are not checked. The search limit group entries under localhost have priority over those under IBMpolicies.

---

### Creating a search limit group

To create a search limit group, you must create a group entry using either the Web Administration Tool or the command line.

#### Using Web Administration:

If you have not done so already, expand the **Directory management** category in the navigation area.

1. Click **Add an entry** or click **Manage entries** and select the location (cn=ibmPolicies or cn=localhost) and click **Add**.
2. Select one of the group object classes from **Structural object class** menu.
  - accessGroup
  - accessRole
  - AIXaccessGroup
  - eNTGroup
  - groupofNames
  - groupofUniqueNames
  - groupofURLs
  - ibm-nestedGroup
  - ibm-proxyGroup
  - ibm-staticGroup
  - ibm-dynamicGroup
3. Click **Next**.
4. Select **ibm-searchLimits** auxiliary object class you want to use from the **Available** menu and click **Add**. Repeat this process for each additional auxiliary object class you want to add. You can also delete an auxiliary object class from the **Selected** menu by selecting it and clicking **Remove**.
5. Click **Next**.
6. In the **Relative DN** field, enter the relative distinguished name (RDN) of the group that you are adding, for example, cn=Search Group1.
7. In the **Parent DN** field, enter the distinguished name of the tree entry you are selecting, for example, cn=localhost. You can also click **Browse** to select the Parent DN from the list. Select your choice and click **Select** to specify the Parent DN that you want. The **Parent DN** defaults to the entry selected in the tree.

**Note:** If you started this task from the **Manage entries** panel, this field is prefilled for you. You selected the **Parent DN** before clicking **Add** to start the add entry process.

8. At the **Required attributes** tab enter the values for the required attributes.
  - **cn** is the relative DN you specified earlier.
  - In the the **ibm-searchSizeLimit** field specify the number of entries that define the size of the search . This number can range between 0 and 2,147,483,647. A setting of 0 is the same as **Unlimited**.
  - In the the **ibm-searchTimeLimit** field specify the number of seconds that define the duration of the search . This number can range between 0 and 2,147,483,647. A setting of 0 is the same as **Unlimited**.
  - Depending on the object class you selected, you might see a **Member** or **uniqueMember** field. These are the members of the group you are creating. The entry is in the form of a DN, for example, cn=Bob Garcia,ou=austin,o=ibm,c=us.

**Notes:**

- a. If the attribute is multi-valued and you want to add more than one value for a particular attribute, click **Multiple values**. See “Multiple values for attributes” on page 295.
- b. If an attribute requires binary data, click **Binary data**. See “Binary data for attributes” on page 295

- c. If your server has language tags enabled, click **Language tag value** to add or remove language tag descriptors. See “Language tags” on page 297 and “Language tag values for attributes” on page 298 for more information.
  - d. If an attribute contains referrals, click **Manage referral**. See Chapter 12, “Referrals,” on page 153 and “Creating default referrals” on page 157 for more information.
9. Click **Optional attributes**.
  10. At the **Optional attributes** tab enter the values as appropriate for the attributes.
  11. Click **Finish** to create the entry.

## Using the command line:

To set search limits of 4000 seconds and 2000 entries for user1 and user2 in cn=localhost location, issue the following command:

```
idsldapmodify -a -D <adminDN> -w <adminPW> -i <filename>
```

where <filename> contains:

```
Dn: cn=Search1, cn=localhost
Cn: Search1
member: cn=user1,o=ibm
member: cn=user2,o=ibm
ibm-searchTimeLimit: 4000
ibm-searchSizeLimit: 2000
objectclass: top
objectclass: ibm-searchLimits
objectclass: groupofNames
```

---

## Modifying a search limit group

You can modify a search limit group, such as changing the size or time limits of the search, or adding or deleting members of the group by using either the Web Administration Tool or the command line.

### Using Web Administration:

To modify a search limit group, see “Modifying an entry” on page 301.

### Using the command line:

To change the searchTimeLimit to 3000 seconds and change the searchSizeLimit to unlimited, as well as add a new member (Bob Garcia), issue the following command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where <filename> contains:

```
dn: cn=Search1, cn=localhost
changetype: modify
replace: ibm-searchTimeLimit
ibm-searchTimeLimit: 3000
-
replace: ibm-searchSizeLimit
ibm-searchSizeLimit: 0
-
add: member
member: cn=Bob Garcia,ou=austin,o=ibm,c=us
```

---

## Copying a search limit group

Copying a search limit group is useful if you want to have the same search limit group under both localhost and IBMpolicies. It is also useful if you want to create a new group that has similar information to an existing group, but has minor differences.

### Using Server Administration:

To copy a search limit group, see “Copying an entry” on page 302.

### Using the command line:

To view the search groups contained in localhost, issue the command:

```
idsldapsearch -b cn=localhost objectclass=ibm-searchLimits
```

Select the search limit group that you want to copy. Use an editor to change the appropriate information and save the changes to *<filename>*. Then issue the following command:

```
idsldapmodify -a -D <adminDN> -w <adminPW> -i <filename>
```

where *<filename>* contains:

```
Dn: cn=NewSearch1, cn=localhost
Cn: NewSearch1
member: cn=user1,o=ibm
member: cn=user2,o=ibm
ibm-searchTimeLimit: 4000
ibm-searchSizeLimit: 2000
objectclass: top
objectclass: ibm-searchLimits
objectclass: groupofNames
```

---

## Removing a search limit group

To remove a search limit group you can use either the Web Administration Tool or the command line.

### Using Web Administration:

To remove a search limit group, see “Deleting an entry” on page 300.

### Using the command line:

To remove a search limit group using the command line, issue the following command:

```
idsldapdelete -D <adminDN> -w<adminPW> -i<filename>
```

where *<filename>* contains:

```
#list additional DNs here, one per line
cn=Search1, cn=localhost
```

To remove multiple search limit groups, list the DNs. Each DN must be on a separate line.



---

## Chapter 20. Managing a proxy authorization group

The proxy authorization is a special form of authentication. By using this proxy authorization mechanism, a client application can bind to the directory with its own identity but is allowed to perform operations on behalf of another user to access the target directory. A set of trusted applications or users can access the IBM Tivoli Directory Server on behalf of multiple users.

The members in the proxy authorization group can assume any authenticated identities except for the administrator or members of the local or global administrative groups. Members of the proxy authorization group also have the authority to use the group authorization control.

**Note:** The administrator and members of the local administrative group have the authority to assume the identity of a global administrator group member by sending a group authorization control for the global administrator group.

The proxy authorization group can be stored under either localhost or IBMpolicies. A proxy authorization group under IBMpolicies is replicated. A proxy authorization group under localhost is not. You can store the proxy authorization group under both localhost and IBMpolicies. If the proxy group is not stored under one of these DNs, the server ignores the proxy part of the group and treats it as a normal group.

As an example, a client application, client1, can bind to the IBM Tivoli Directory Server with a high level of access permissions. UserA with limited permissions sends a request to the client application. If the client is a member of the proxy authorization group, instead of passing the request to the IBM Tivoli Directory Server as client1, it can pass the request as UserA using the more limited level of permissions. What this means is that instead of performing the request as client1, the application server can access only that information or perform only those actions that UserA is able to access or perform. It performs the request on behalf of or as a proxy for UserA.

**Note:** The attribute member must have its value in the form of a DN. Otherwise an Invalid DN syntax message is returned. A group DN is not permitted to be a member of the proxy authorization group.

Administrators and administrative group members are not permitted to be members of the proxy authorization group. All administrators have authority to use the proxy authorization control, without having to be in that group.

The audit log records both the bind DN and the proxy DN for each action performed using proxy authorization.

---

### Creating a proxy authorization group

To create a proxy authorization group, you must create a group entry using either the Web Administration Tool or the command line.

## Using Web Administration:

If you have not done so already, expand the **Directory management** category in the navigation area.

1. Do one of the following:
  - Click **Add an entry**.
  - Click **Manage entries** and select the location (cn=ibmPolicies or cn=localhost) and click **Add**.
2. Select the **groupofNames** object classes from **Structural object class** menu.
3. Click **Next**.
4. Select **ibm-proxyGroup** auxiliary object class from the **Available** menu and click **Add**. Repeat this process for each additional auxiliary object class you want to add. You can also delete an auxiliary object class from the **Selected** menu by selecting it and clicking **Remove**.
5. Click **Next**.
6. In the **Relative DN** field, enter **cn=proxyGroup**.
7. In the **Parent DN** field, enter the distinguished name of the tree entry you are selecting, for example, cn=localhost. You can also click **Browse** to select the Parent DN from the list. Select your choice and click **Select** to specify the Parent DN that you want. The **Parent DN** defaults to the entry selected in the tree.

**Note:** If you started this task from the **Manage entries** panel, this field is prefilled for you. You selected the **Parent DN** before clicking **Add** to start the add entry process.

8. At the **Required attributes** tab enter the values for the required attributes.
  - **cn** is proxyGroup.
  - **Member** is in the form of a DN, for example, cn=Bob Garcia,ou=austin,o=ibm,c=us.

### Notes:

- a. If the attribute is multi-valued and you want to add more than one value for a particular attribute, click **Multiple values**. Do not create multiple values for cn value. The proxy authorization group must have the well known name, proxyGroup. See “Multiple values for attributes” on page 295.
  - b. If an attribute requires binary data, click **Binary data**. See “Binary data for attributes” on page 295
  - c. If your server has language tags enabled, click **Language tag value** to add or remove language tag descriptors. See “Language tags” on page 297 and “Language tag values for attributes” on page 298 for more information.
  - d. If an attribute contains referrals, click **Manage referral**. See Chapter 12, “Referrals,” on page 153 and “Creating default referrals” on page 157 for more information.
9. Click **Optional attributes**.
  10. At the **Optional attributes** tab enter the values as appropriate for the attributes.
  11. Click **Finish** to create the entry.

## Using the command line:

To create the proxy authentication group with an initial member in the cn=localhost location, issue the following command:

```
idsldapadd -D <adminDN> -w<adminPW> -i<filename>
```

where <filename> contains:

```
dn: cn=proxyGroup,cn=localhost
cn: proxyGroup
member: cn=client1, ou=austin, o=ibm, c=us
objectclass: top
objectclass: container
objectclass: groupOfNames
objectclass: ibm-proxyGroup
```

To add an additional member, issue the command:

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where <filename> contains:

```
dn: cn=proxyGroup,cn=localhost
cn: proxyGroup
changetype: modify
add: member
member: cn=client2, ou=austin, o=ibm, c=us
```

---

## Modifying a proxy authorization group

### Using Server Administration:

To modify the proxy authorization group such as adding or deleting members of the group, see “Modifying an entry” on page 301.

### Using the command line:

To modify the proxy authorization group in the cn=IBMpolicies location, issue the following command:

**Note:** This command deletes user1, and adds user2 and user3.

```
idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
```

where <filename> contains:

```
dn: cn=proxyGroup,cn=IBMpolicies
changetype: modify
delete: member
member: cn=client1, ou=austin, o=ibm, c=us
-
add: member
member: cn=client2, ou=austin, o=ibm, c=us
-
add: member
member: cn=client3, ou=austin, o=ibm, c=us
```

---

## Copying a proxy authorization group

### Using Server Administration:

Copying a proxy authorization group is useful if you want to have the same proxy authorization group under both localhost and IBMpolicies.

To copy a proxy authorization group, see “Copying an entry” on page 302.

## Using the command line:

To view the proxy authorization group contained in localhost, issue the command:  
`idsldapsearch -D <adminDN> -w <adminPW> -b cn=localhost objectclass=ibm-proxyGroup`

The following is the output for this command:

```
Dn: cn=proxyGroup, cn=localhost
Cn: proxyGroup
objectclass: ibm-proxyGroup
objectclass: groupOfNames
member: cn=client1, ou=austin, o=ibm, c=us
member: cn=client2, ou=austin, o=ibm, c=us
member: cn=client3, ou=austin, o=ibm, c=us
```

Select the proxy authorization group. Use an editor to change `cn=localhost` to `cn=IBMpolicies`, and save the changes to `<filename>`.

Then issue the following command:

```
idsldapmodify -a -D <adminDN> -w <adminPW> -i <filename>
```

where `<filename>` contains:

```
Dn: cn=proxyGroup, cn=IBMpolicies
Cn: proxyGroup
objectclass: ibm-proxyGroup
objectclass: groupOfNames
member: cn=client1, ou=austin, o=ibm, c=us
member: cn=client2, ou=austin, o=ibm, c=us
member: cn=client3, ou=austin, o=ibm, c=us
```

---

## Removing the proxy authorization group

To remove a member from the proxy authorization group use either of the following methods.

### Using Web Administration:

To remove a proxy authorization group, see “Deleting an entry” on page 300.

### Using the command line:

To remove the proxy authorization group issue the command:

```
idsldapdelete -D <adminDN> -w <adminpw> -s "cn=ProxyGroup,cn=IBMpolicies"
```

Although the proxy authorization group can be managed by the Web Administration Tool, proxy authorization is not recognized by any of the other Web Administration Tool functions. The proxy authorization function is utilized by including the Proxy Authorization Control with your LDAP operations or using the LDAP commands with the `-y` option. For example:

```
idsldapsearch -D "cn=client1,ou=austin,o=ibm,c=us" -w <client1password>
-y "cn=userA,o=ibm,c=us" -b "o=ibm,c=us" -s sub ou=austin
```

Based on the above `idsldapsearch` specification, `client1` can read from the target directories whatever `userA` has permission to read.

---

## **Part 4. User-related tasks**



---

## Chapter 21. Realms, templates, users, and groups

A realm is a collection of users and the groups to which they belong. For example a company, a bowling team, or a club could all be realms.

Realms are defined by creating entries of object class "ibm-realm" anywhere in a user naming context (not under cn=localhost, cn=schema or cn=configuration). The ibm-realm object defines the realm's name (cn), a group of realm administrators (ibm-realmAdminGroup), a user-template object (ibm-realmUserTemplate) specifying the object classes and attributes for users in the realm, and the location of container entries under which user and group entries are stored (ibm-realmUserContainer and ibm-realmGroupContainer). The directory administrator and members of the administrative group are responsible for managing user-templates, realms and realm administrator groups. After a realm is created, members of that realm's administrator group (realm administrators) are responsible for managing the users and groups within that realm.

---

### Creating a realm

Expand the **Realms and templates** category in the navigation area of the Web Administration Tool.

1. Click **Add realm**.
  - Enter the name for the realm. For example **realm1**.
  - Enter the Parent DN that identifies the location of the realm. This entry is in the form of a suffix, for example o=ibm,c=us. You can also click **Browse** to select the location of the subtree that you want.
2. Click **Next** to continue.
3. Review the information. At this point you haven't actually created the realm, so **User template** and **User search filter** can be ignored.
4. Click **Finish** to create the realm.

---

### Creating a realm administrator

To create a realm administrator, you must first create an administration group for the realm.

#### Creating the realm administration group

Expand the **Directory management** category in the navigation area of the Web Administration Tool.

1. Click **Manage entries**.
2. Expand the tree for the parent DN that identifies the location of the realm you just created, and select the realm you just created, **cn=realm1,o=ibm,c=us**.
3. Expand the **Select Action** menu, select **Edit ACL** and click **Go**.
4. Click the **Owners** tab.
5. Ensure that **Propagate owner** is checked.
6. Enter the Subject DN for the realm, **cn=realm1,o=ibm,c=us**.
7. Change the Subject type to **group**.
8. Click **Add**.

9. Click **OK** to save your changes and return to the **Manage entries** panel.

## Creating the administrator entry

If you do not already have a user entry for the administrator, you must create one.

Expand the **Directory management** category in the navigation area of the Web Administration Tool.

1. Click **Manage entries**.
2. Expand the tree to the location where you want the administrator entry to reside.

**Note:** Locating the administrator entry outside of the realm avoids giving the administrator the ability to accidentally delete him or herself. In this example the location might be **o=ibm,c=us**.

3. Click **Add**.
4. Select the **Structural object class**, for example **person**.
5. Click **Next**.
6. Select any auxiliary object class you want to add.
7. Click **Next**.
8. Enter the required attributes for the entry. For example,
  - **Relative DN** cn=John Doe
  - **Parent DN** o=ibm,c=us (This is pre-filled for you.)
  - **cn** John Doe
  - **sn** Doe

### Notes:

- a. If the attribute is multi-valued and you want to add more than one value for a particular attribute, click **Multiple values**. See “Multiple values for attributes” on page 295.
  - b. If an attribute requires binary data, click **Binary data**. See “Binary data for attributes” on page 295
  - c. If your server has language tags enabled, click **Language tag value** to add or remove language tag descriptors. See “Language tags” on page 297 and “Language tag values for attributes” on page 298 for more information.
  - d. If an attribute contains referrals, click **Manage referral**. See Chapter 12, “Referrals,” on page 153 and “Creating default referrals” on page 157 for more information.
9. On the **Optional attributes** tab ensure that you have assigned a user password.
  10. When you are done, click **Finish**.

## Adding the administrator to the administration group.

Expand the **Directory management** category in the navigation area of the Web Administration Tool.

1. Click **Manage entries**.
2. Expand the tree (o=ibm,c=us) and select the realm you just created, **cn=realm1,o=ibm,c=us**.
3. Expand the **Select Action** menu, select **Manage members** and click **Go**.



4. The Static group members tab is highlighted. Click **Load** to display the members of the group. In this example, you have not added any members yet so no entries are displayed in the table.
5. Type the name of entry that you want to add as a member of the group, for example the entry you created in the previous task, **cn=John Doe,o=ibm,c=us** in the member field or select it using the **Browse** function (expand **o=ibm,c=us** and select **cn=John Doe,o=ibm,c=us**).
6. Click **Add**.
7. **cn=John Doe,o=ibm,c=us** is displayed in the table. Click **Apply** to save the change and continue adding additional members or if you are finished, click **Ok** to save the changes and return to the manage entries panel.

You have created an administrator that can manage entries within the realm. See “Managing members of group entries” on page 339 for additional information about adding members to a group.

---

## Creating a template

After you have created a realm, your next step is to create a user template. A template helps you to organize the information you want to enter. Expand the **Realms and templates** category in the navigation area of the Web Administration Tool.

1. Click **Add user template**.
  - If you have preexisting templates, you can select a template to have its settings copied to the template you are creating. However, in this task you are creating your first template.
  - Enter the name for the template, for example, **template1**.
  - Enter the location where the template is going to reside. For replication purposes, locate the template in the subtree of the realm that is going to use this template. For example, for the realm you created in the previous operations **cn=realm1,o=ibm,c=us**, locate the template in the subtree **o=ibm,c=us**. You can also click **Browse** to select a different subtree for the location of the template.
2. Click **Next**. You can click **Finish** to create an empty template. You can later add information to the template, see “Editing a template” on page 361.
3. If you clicked **Next**, choose the structural object class for the template, for example **inetOrgPerson**. You can also add any auxiliary object classes that you want.
4. Click **Next**.
5. Select a naming attribute from the **Naming attribute** drop-down menu. This attribute is used for the RDN of each entry in a realm that uses the template. The naming attribute, for example **givenName**, must have a value that is unique to each member in the realm that uses this template. The value is the display name for the user entry in the user lists for user and group tasks. For example, if the **givenName** is the naming attribute and **Bob Garcia** is entered, the entry appears as **Bob Garcia** in the appropriate user lists.
6. A **Required** tab has been created on the template. You can modify the information contained on this tab.
  - a. Select **Required** in the tab menu and click **Edit**. The Edit tab panel is displayed. You see the name of the tab **Required** and the selected attributes that are required by the object class, **inetOrgPerson**:
    - \*sn - surname

- \*cn - common name

**Note:** The \* denotes required information.

- b. If you want to add additional information to this tab, select the attribute from the **Attributes** menu. For example, select **departmentNumber** and click **Add**. Select **employeeNumber** and click **Add**. Select **title** and click **Add**. The **Selected attributes** menu now reads:
  - title
  - employeeNumber
  - departmentNumber
  - \*sn
  - \*cn
- c. You can rearrange the way that these fields appear on the template by highlighting the selected attribute and clicking **Move up** or **Move down**. This changes the position of the attribute by one position. Repeat this procedure until you have the attributes arranged in the order you want them. For example,
  - \*sn
  - \*cn
  - title
  - employeeNumber
  - departmentNumber
- d. You can also modify each selected attribute.
  - 1) Highlight the attribute in the **Selected attributes** box and click **Edit**.
  - 2) You can change the display name of the field used on the template. For example, if you want **departmentNumber** to be displayed as **Department number** enter that into the **Display name** field.
  - 3) You can also supply a default value to prefill the attribute field in the template. For example, if most of the users that are going to be entered are members of Department 789, you can enter 789 as the default value. The field on the template is prefilled with 789. The value can be changed when you add the actual user information.
  - 4) Click **OK**.
- e. Click **OK**.
7. To create another tab category for additional information, click **Add**.
  - Enter the name for the new tab. For example, Address information.
  - For this tab, select the attributes from the **Attributes** menu. For example, select **homePostalAddress** and click **Add**. Select **postOfficeBox** and click **Add**. Select **telephoneNumber** and click **Add**. Select **homePhone** and click **Add**. Select **facsimileTelephoneNumber** and click **Add**. The **Selected attributes** menu reads:
    - homePostalAddress
    - postOfficeBox
    - telephoneNumber
    - homePhone
    - facsimileTelephoneNumber
  - You can rearrange the way that these fields appear on the template by highlighting the selected attribute and clicking **Move up** or **Move down**.

This changes the position of the attribute by one position. Repeat this procedure until you have the attributes arranged in the order you want them. For example,

- homePostalAddress
  - postOfficeBox
  - telephoneNumber
  - facsimileTelephoneNumber
  - homePhone
- Click **OK**.
8. Repeat this process for as many tabs as you want to create. When you are finished click **Finish** to create the template.

---

## Adding the template to a realm

After you have created a realm and a template, you need to add the template to the realm. Expand the **Realms and templates** category in the navigation area of the Web Administration Tool.

1. Click **Manage realms**.
2. Select the realm you want to add the template to, in this example, **cn=realm1,o=ibm,c=us** and click **Edit**.
3. Scroll down to **User template** and expand the drop-down menu.
4. Select the template, in this example, **cn=template1,o=ibm,c=us**.
5. Click **OK**.
6. Click **Close**.

---

## Creating groups

Expand the **Users and groups** category in the navigation area of the Web Administration Tool.

1. Click **Add group**.
2. Enter the name of the group that you want to create. For example **group1**.
3. Select the realm that you want to add the user to from the drop-down menu. In this case **realm1**.
4. Click **Next**.
5. Click **Finish** to create the group. If you already have users in the realm you can click **Next** and select users to add to group1. Then click **Finish**.

See "Groups" on page 329 for additional information.

---

## Adding a user to the realm

Expand the **Users and groups** category in the navigation area of the Web Administration Tool.

1. Click **Add user**.
2. Select the realm that you want to add the user to from the drop-down menu. In this case **realm1**.
3. Click **Next**. The template that you just created, **template1**, is displayed. Fill in the required fields, denoted by an asterisk (\*) and any of the other fields on the tabs.
4. If you have already created groups within the realm, you can also add the user to one or more groups.

- a. Select the **User group** tab.
  - b. Click **Add**.
  - c. Either type the name of the group (Group1) in the **Group name** field or click **Available groups** and select the group or groups that you want to add the user to from the list. You can also select a group and click **View** to see the existing members of that group. See “Managing memberships for an entry” on page 340 for additional information on group memberships.
5. When you are done, click **Finish**.

---

## Managing realms

After you have set up and populated your initial realm, you can add more realms or modify existing realms.

Expand the **Realms and templates** category in the navigation area and click **Manage realms**. A list of existing realms is displayed. From this panel you can add a realm, edit a realm, remove a realm or edit the access control list (acIs) of the realm.

### Adding a realm

Expand the **Realms and templates** category in the navigation area of the Web Administration Tool.

1. Click **Add realm**.
  - Enter the name for the realm. For example **realm2**.
  - If you have preexisting realms, for example **realm1**, you can select a realm to have its settings copied to the realm you are creating.
  - Enter the Parent DN that identifies the location of the realm. This entry is in the form of a suffix, for example **o=ibm,c=us**. You can also click **Browse** to select the location of the subtree that you want.
2. Click **Next** to continue or click **Finish**.
3. If you clicked **Next**, review the information.
4. Select a **User template** from the drop-down menu. If you copied the settings from a preexisting realm, its template is prefilled in this field.
5. Enter a **User search filter**.
6. Click **Finish** to create the realm.

### Editing a realm

Expand the **Realms and templates** category in the navigation area of the Web Administration Tool.

- Click **Manage realms**.
- Select the realm that you want to edit from the list of realms.
- Click **Edit**.
  - You can use the **Browse** buttons to change the
    - Administrator group
    - Group container
    - User container
  - You can select a different template from the drop-down menu.
  - Click **Edit** to modify the **User search filter**.
- Click **OK** when you are finished.

## Removing a realm

Expand the **Realms and templates** category in the navigation area of the Web Administration Tool.

1. Click **Manage realms**.
2. Select the realm you want to remove.
3. Click **Delete**.
4. When prompted to confirm the deletion, click **OK**.
5. The realm is removed from the list of realms.

## Editing ACLs on the realm

To view ACL properties using the Web Administration Tool utility and to work with ACLs, see “Working with ACLs” on page 318.

See Chapter 17, “Access control lists,” on page 309 for additional information.

---

## Managing templates

After you have created your initial template, you can add more templates or modify existing templates.

Expand the **Realms and templates** category in the navigation area and click **Manage user templates**. A list of existing templates is displayed. From this panel you can add a template, edit a template, remove a template or edit the access control list (ACLs) of the template.

## Adding a user template

Expand the **Realms and templates** category in the navigation area of the Web Administration Tool.

1. Click **Add user template** or click **Manage user templates** and click **Add**.
  - If you have preexisting templates, for example **template1**, you can select a template to have its settings copied to the template you are creating.
  - Enter the name for the new template. For example **template2**.
  - Enter the Parent DN that identifies the location of the template. This entry is in the form of a DN, for example **o=ibm,c=us**. You can also click **Browse** to select the location of the subtree that you want.
2. Click **Next**. You can click **Finish** to create an empty template. You can later add information to the template see “Editing a template” on page 361.
3. If you clicked **Next**, choose the structural object class for the template, for example **inetOrgPerson**. You can also add any auxiliary object classes that you want.
4. Click **Next**.
5. From the **Naming attribute** drop-down menu, select the attribute that is used for the RDN of each entry in a realm that uses the template. This naming attribute, for example **employeeNumber**, must have a value that is unique to each member in the realm that uses this template. The value of this naming attribute is the display name for the user entry in the user lists for user and group tasks. For example, if the **employeeNumber** is the naming attribute and **1234abc** is entered, the entry appears as **1234abc** in the appropriate user lists.
6. A **Required** tab has been created on the template. You can modify the information contained on this tab.

- a. Select **Required** in the tab menu and click **Edit**. The Edit tab panel is displayed. You see the name of the tab **Required** and the selected attributes that are required by the object class, **inetOrgPerson**:
  - \*sn - surname
  - \*cn - common name

**Note:** The \* denotes required information.

- b. If you want to add additional information to this tab, select the attribute from the **Attributes** menu. For example, select **departmentNumber** and click **Add**. Select **employeeNumber** and click **Add**. Select **title** and click **Add**. The **Selected attributes** menu now reads:

- title
- employeeNumber
- departmentNumber
- \*sn
- \*cn

- c. You can rearrange the way that these fields appear on the template by highlighting the selected attribute and clicking **Move up** or **Move down**. This changes the position of the attribute by one position. Repeat this procedure until you have the attributes arranged in the order you want them. For example,

- \*sn
- \*cn
- title
- employeeNumber
- departmentNumber

- d. You can also modify each selected attribute.

- 1) Highlight the attribute in the **Selected attributes** box and click **Edit**.
- 2) You can change the display name of the field used on the template. For example, if you want **departmentNumber** to be displayed as **Department number** enter that into the **Display name** field.
- 3) You can also supply a default value to prefill the attribute field in the template. For example, if most of the users that are going to be entered are members of Department 789, you can enter 789 as the default value. The field on the template is prefilled with 789. The value can be changed when you add the actual user information.
- 4) Click **OK**.

- e. Click **OK**.

7. To create another tab category for additional, click **Add**.

- Enter the name for the new tab. For example, Address information.
- To this tab, select the attribute from the **Attributes** menu. For example, select **homePostalAddress** and click **Add**. Select **postOfficeBox** and click **Add**. Select **telephoneNumber** and click **Add**. Select **homePhone** and click **Add**. Select **facsimileTelephoneNumber** and click **Add**. The **Selected attributes** menu reads:
  - homePostalAddress
  - postOfficeBox
  - telephoneNumber
  - homePhone

- facsimileTelephoneNumber
  - You can rearrange the way that these fields appear on the template by highlighting the selected attribute and clicking **Move up** or **Move down**. This changes the position of the attribute by one position. Repeat this procedure until you have the attributes arranged in the order you want them. For example,
    - homePostalAddress
    - postOfficeBox
    - telephoneNumber
    - facsimileTelephoneNumber
    - homePhone
  - Click **OK**.
8. Repeat this process for as many tabs as you want to create. When you are finished click **Finish** to create the template.

## Editing a template

Expand the **Realms and templates** category in the navigation area of the Web Administration Tool.

- Click **Manage user templates**.
- Select the template that you want to edit from the list of templates.
- Click **Edit**.
- If you have preexisting templates, for example template1, you can select a template to have its settings copied to the template you are editing.
- Click **Next**.
  - You can use the drop-down menu to change the structural object class of the template
  - You can add or remove auxiliary object classes.
- Click **Next**.
- You can modify the tabs and attributes contained in the template. See 6 on page 359 for information on how to modify the tabs.
- When you are done, click **Finish**.

## Removing a template

Expand the **Realms and templates** category in the navigation area of the Web Administration Tool.

1. Click **Manage user templates**.
2. Select the template that you want to remove.
3. Click **Delete**.
4. When prompted to confirm the deletion, click **OK**.
5. The template is removed from the list of templates.

## Editing ACLs on the template

Expand the **Realms and template** category in the navigation area of the Web Administration Tool.

1. Click **Manage user templates**.
2. Select the template for which you want to edit the ACLs.
3. Click **Edit ACL**.

To view ACL properties using the Web Administration Tool utility and to work with ACLs, see “Working with ACLs” on page 318.

See Chapter 17, “Access control lists,” on page 309 for additional information.

---

## Managing users

After you have set up your realms and templates, you can populate them with users.

### Adding users

Expand the **Users and groups** category in the navigation area of the Web Administration Tool.

1. Click **Add user** or click **Managing users** and click **Add**.
2. Select the realm that you want to add the user to from the drop-down menu.
3. Click **Next**. The template that is associated with that realm, is displayed. Fill in the required fields, denoted by an asterisk (\*) and any of the other fields on the tabs. If you have already created groups within the realm, you can also add the user to one or more groups.
4. When you are done, click **Finish**.

### Finding users within the realm

Expand the **Users and groups** category in the navigation area of the Web Administration Tool.

1. Click **Manage users**.
2. Expand the **Select Actions** menu, select **Show find toolbar** and click **Go**.
3. Select the realm that you want to search to from the **Select realm** field.
4. Enter the search string in the **Search** field. See “Finding” on page 25 for information about how to use the Find utility.
5. You can perform the following operations on a selected user:
  - **Add** - “Adding users.”
  - **Edit** - See “Editing a user’s information.”
  - **Copy** - See “Copying a user” on page 363.
  - **Delete** - See “Removing a user” on page 363.
6. When you are done, click **OK**.

### Editing a user’s information

Expand the **Users and groups** category in the navigation area of the Web Administration Tool.

1. Click **Manage users**.
2. Select a realm from the drop-down menu. Click **View users**, if the users are not already displayed in the **Users** box.
3. Select the user you want to edit and click **Edit**.
4. Modify the information on the tabs, modify group membership.
5. When you are done click, **OK**.



## Copying a user

If you need to create a number of users that have mostly identical information, you can create the additional users by copying the initial user and modifying the information.

Expand the **Users and groups** category in the navigation area of the Web Administration Tool.

1. Click **Manage users**.
2. Select a realm from the drop-down menu. Click **View users**, if the users are not already displayed in the **Users** box.
3. Select the user you want to copy and click **Copy**.
4. Modify the appropriate information for the new user, for example the required information that identifies a specific user, such as sn or cn. Information that is common to both users need not be changed.
5. When you are done click, **OK**.

## Removing a user

Expand the **Users and groups** category in the navigation area of the Web Administration Tool.

1. Click **Manage users**.
2. Select a realm from the drop-down menu. Click **View users**, if the users are not already displayed in the **Users** box.
3. Select the user you want to remove and click **Delete**.
4. When prompted to confirm the deletion, click **OK**.
5. The user is removed from the list of users.

---

## Managing groups

After you have set up your realms and templates, you can create groups.

### Adding groups

Expand the **Users and groups** category in the navigation area of the Web Administration Tool.

1. Click **Add group** or click **Manage groups** and click **Add**.
2. Enter the name of the group that you want to create.
3. Select the realm that you want to add the group to from the drop-down menu.
4. Click **Finish** to create the group. If you already have users in the realm you can click **Next** and select users to add to the group. Then click **Finish**.

See “Groups” on page 329 for additional information.

### Finding groups within the realm

Expand the **Users and groups** category in the navigation area of the Web Administration Tool.

1. Click **Manage groups**.
2. Expand the **Select Actions** menu, select **Show find toolbar** and click **Go**.
3. Select the realm that you want to search to from the **Select realm** field.
4. Enter the search string in the **Search** field. See “Finding” on page 25 for information about how to use the Find utility.
5. You can perform the following operations on a selected group:

- **Add** - See “Adding groups” on page 363.
  - **Edit** - See “Editing a group’s information.”
  - **Copy** - See “Copying a group.”
  - **Delete** - See “Removing a group.”
6. When you are done, click **Close**.

## Editing a group’s information

Expand the **Users and groups** category in the navigation area of the Web Administration Tool.

1. Click **Manage groups**.
2. Select a realm from the drop-down menu. Click **View groups**, if the groups are not already displayed in the **Groups** box.
3. Select the group you want to edit and click **Edit**.
4. You can add or remove users from the group.
5. When you are done click, **OK**.

## Copying a group

If you need to create a number of groups that have mostly the same members, you can create the additional groups by copying the initial group and modifying the information.

Expand the **Users and groups** category in the navigation area of the Web Administration Tool.

1. Click **Manage groups**.
2. Select a realm from the drop-down menu. Click **View groups**, if the users are not already displayed in the **Groups** box.
3. Select the group you want to copy and click **Copy**.
4. Change the group name in the **Group name** field. The new group has the same members as the original group.
5. You can **Add** new group members, **Delete** group members or **View** a group member’s information by selecting the group member and clicking the appropriate operation.
6. When you are done click, **OK**. The new group is created and contains the same members as the original group with any addition or removal modifications you made during the copy procedure.

## Removing a group

Expand the **Users and groups** category in the navigation area of the Web Administration Tool.

1. Click **Manage groups**.
2. Select a realm from the drop-down menu. Click **View groups**, if the groups are not already displayed in the **Groups** box.
3. Select the group you want to remove and click **Delete**.
4. When prompted to confirm the deletion, click **OK**.
5. The group is removed from the list of groups.

---

## Part 5. Command line utilities

Use these utilities as an alternative method of administering your IBM Tivoli Directory Server.



---

## Chapter 22. Command line utilities

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

**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

**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:

- `idldapchangepwd`
- `idldapdelete`
- `idldapexop`
- `idldapmodify`, `idldapadd`
- `idldapmodrdn`
- `idldapsearch`

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

### Client Utilities

- “`idsdirctl`, `ibmdirctl`” on page 369
- “`idldapchangepwd`, `ldapchangepwd`” on page 370
- “`idldapdelete`, `ldapdelete`” on page 373
- “`idldapdiff`, `ldapdiff`” on page 377
- “`idldapexop`, `ldapexop`” on page 386
- “`idldapmodify`, `ldapmodify`, `idldapadd`, `ldapadd`” on page 396
- “`idldapmodrdn`, `ldapmodrdn`” on page 402
- “`idldapsearch`, `ldapsearch`” on page 406
- “`idldaptrace`, `ldaptrace`” on page 415
- “`tbindmsg`” on page 418

### Server Utilities

- “`ddsetup`” on page 420
- “`idsbulkload`, `bulkload`” on page 423
- “`idscfgchglg`” on page 429
- “`idscfgdb`” on page 431
- “`idscfgsch`” on page 432
- “`idscfgsuf`” on page 433
- “`idsdbback`, `dbback`” on page 434
- “`idsdbrestore`, `dbrestore`” on page 435
- “`idsdb2ldif`, `db2ldif`” on page 436
- “`idsdiradm`, `ibmdiradm`” on page 438
- “`idsdnpw`” on page 440
- “`idsgendirksf`” on page 441

- “idsicrt” on page 442
- “idsidrop” on page 445
- “idsilist” on page 446
- “idsimigr” on page 448
- “idsldif2db, ldif2db” on page 448
- “idslink” on page 450
- “idslogmgmt” on page 449
- “IDSProgRunner” on page 450
- “idsrunstats, runstats” on page 451
- “idssethost” on page 451
- “idssetport” on page 452
- “idsslapd, ibmslapd” on page 454
- “idssnmp” on page 455
- “idssupport” on page 455
- “idsucfgchglg” on page 455
- “idsucfgdb” on page 456
- “idsucfgsch” on page 457
- “idsucfgsuf” on page 459
- “ldtrc” on page 460
- “runscript” on page 462

The client utilities all use the `ldap_sasl_bind` API. When `bind` is invoked, several results can be returned. Following are 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.

---

## Client utilities

This section provides a description of the client utilities. They are also documented in “Chapter 2. LDAP Utilities” in the *IBM Tivoli Directory Server Version 6.0: Client SDK Programming Reference*.

## idsdirctl, ibmdirctl

The administration daemon control program. The administration daemon (**idsdiradm**) must be running. See “Starting an instance of the directory administration daemon” on page 17 and Chapter 4, “Directory administration daemon,” on page 17.

**Note:** Only the administrator may use this utility.

### Synopsis

```
ibmdirctl [-d | D adminDN] [-h | H hostname] [-K keyfile] [-N key_name]
 [-p port] [-P key_pw] [-v] [-w | W adminPW | ?] [-Y] [-Z] [-?]
command -- [idsslapd options]
```

where *command* is {start|stop|restart|status|statusreturn|admstop}

### 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 | D adminDN**

Use adminDN to bind to the LDAP directory. The adminDN is a string-represented DN (see LDAP Distinguished Names).

#### **-h | H hostname**

Specify an alternate host on which the ldap server and the admin daemon are running.

#### **-K keyfile**

Specifies the file to use for keys.

#### **-N key\_name**

Specifies the private key name to use in keyfile.

#### **-p port**

Specify an alternate TCP port where the admin daemon is listening. The default LDAP port is 3538.

#### **-P key\_pw**

Specifies the key file password.

#### **-v**

Specifies to run in verbose mode.

#### **-w | 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.

#### **-Y**

Specifies to use a secure LDAP connection (TLS).

#### **-Z**

Specifies to use a secure LDAP connection (SSL).

#### **-?**

Displays the syntax format.

*command*

- **start** - Start the server.
- **stop** - Stop the server.

- **restart** - Stop then start the server.
- **status** - Query the status the server.
- **statusreturn** - Sets exit code (0=running, 1=starting, 2=stopped)
- **admstop** - Stop the IBM Tivoli Directory Server administration daemon.

**Note:** The **stop** command can be issued directly to the ldap server.

### -- idsslapd options

The only **idsslapd** options that the **idsslapd** process takes at startup time are:

- **-a** - Start the server in configuration only mode.
- **-n** - Do not start the server, if the server is unable to start with the database backends (no configuration only mode).

### Notes:

1. If **idsslapd** options are requested, they must be preceded by the **--**.
2. The **idsslapd** options are ignored if the **stop**, **status** or **admstop** commands are issued.

### 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] [-G realm] [-h ldaphost]
[-K keyfile] [-m mechanism] [-M] [-N certificatename]
[-O maxhops] [-p ldapport] [-P keyfilepw] [-R]
[-U username] [-v] [-V version] [-x] [-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 DN's 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 "IANA character sets supported by platform" on page 483 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 “Debugging levels” on page 462 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:".

**-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.

**-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/V6.0*
- HP-UX operating systems - */opt/IBM/ldap/V6.0*
- Linux operating systems - */opt/ibm/ldap/V6.0*
- Solaris operating systems - */opt/IBM/ldap/V6.0*
- Windows operating systems - *<local\_drive>:\Program Files\IBM\LDAP\V6.0* (This is the default install location. The actual *IDS\_LDAP\_HOME* is determined during installation.)

See *IBM Directory C-Client SDK 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. For more information on managing an SSL or TLS key database, see “Using *gsk7ikm*” on page 121. Also see the “Security functions” on page 373 and “Secure Sockets Layer” on page 115 for more information about SSL and certificates.

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 *ldappport*  
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.
- R Specifies that referrals are not to be automatically followed.
- 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).
- 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 419.

### 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

## idsldapdelete, ldapdelete

The LDAP delete-entry tool

### Synopsis

```
idsldapdelete | ldapdelete [-c] [-C charset] [-d debuglevel][-D binddn]
[-f file] [-G realm] [-h ldaphost] [-i file] [-k]
[-K keyfile] [-l] [-m mechanism] [-M] [-n]
[-N certificatename] [-O maxops] [-p ldapport]
[-P keyfilepw] [-R] [-s] [-U username} [-v] [-V version]
[-w passwd | ?] [-x] [-y proxydn] [-Y] [-Z] [dn]...
```

### 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 DN's are deleted. Each DN is a string-represented DN. If no DN arguments are provided, a list of DN's 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 mode. Errors are reported, but **idsldapdelete** continues with modifications. Otherwise the default action is to exit after reporting an error.
- C *charset* Specifies that the DN's supplied as input to the **idsldapdelete** 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 "IANA character sets supported by platform" on page 483 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 "Debugging levels" on page 462 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:".

**-f** *file*

Read a series of lines from *file*, performing one LDAP delete for each line in the file. Each line in the file should contain a single distinguished name.

**-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 delete for each line in the file. Each line in the file should contain a single distinguished name.

**-k**

Specifies to use server administration control.

This option sends the Server administration control. See "Server administration control" in the *IBM Tivoli Directory Server C-Client SDK 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/V6.0*
- HP-UX operating systems - */opt/IBM/ldap/V6.0*
- Linux operating systems - */opt/ibm/ldap/V6.0*
- Solaris operating systems - */opt/IBM/ldap/V6.0*
- Windows operating systems - *<local\_drive>:\Program Files\IBM\LDAP\V6.0* (This is the default install location. The actual **IDS\_LDAP\_HOME** is determined during installation.)

See *IBM Directory C-Client SDK 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. For more information on managing an SSL or TLS key database, see "Using gsk7ikm" on page 121. Also see the "Security functions" on page 376 and "Secure Sockets Layer" on page 115 for more information about SSL and certificates.

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

**-l** Do not replicate the entry.

This option sends the Do not replicate control. See "Do not replicate control" in the *IBM Tivoli Directory Server C-Client SDK Programming Reference*.

**-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 don't actually modify entries. Useful for debugging in conjunction with **-v**.

**-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.

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

**-s** Use this option to delete the subtree rooted at the specified entry.

This option sends the Subtree delete control. See "Subtree delete control" in the *IBM Tivoli Directory Server C-Client SDK Programming Reference*.

- 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 **idsldapdelete** 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 **idsldapdelete**, 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).
- 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**  
Specifies one or more DN arguments. Each DN should be a string-represented DN.

## 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+C or Ctrl+D.

## Security functions

To use the SSL or TLS -related functions associated with this utility, see "SSL, TLS notes" on page 419.

## 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]
```

**Note:** On UNIX systems, you must run the **idsldapdiff** command with a fully qualified path name, for example:

```
/opt/IBM/LDAP/V6.0/bin/idsldapdiff
```

Or you can also do one of the following:

- Set `/opt/IBM/LDAP/V6.0/bin` in the `$PATH` environment variable.
- Create a link manually. Use the following command:

```
ln -s /opt/IBM/LDAP/V6.0/bin/idsldapdiff /usr/bin/idsldapdiff
```

### 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 encryption 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 the 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.
- S** Specifies to compare the schema on both of the servers. Compares and fixes using **-S** can be made with any bind DN.
- 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/V6.0`
- HP-UX operating systems - `/opt/IBM/ldap/V6.0`
- Linux operating systems - `/opt/ibm/ldap/V6.0`
- Solaris operating systems - `/opt/IBM/ldap/V6.0`
- Windows operating systems - `<local_drive>:\Program Files\IBM\LDAP\V6.0` (This is the default install location. The actual `IDS_LDAP_HOME` is determined during installation.)

See *IBM Directory C-Client SDK 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. For more information on managing an SSL key database, see "Using `gsk7ikm`" on page 121. Also see the "Security functions" on page 386 and "Secure Sockets Layer" on page 115 for more information about SSL and certificates.

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/V6.0`
- HP-UX operating systems - `/opt/IBM/ldap/V6.0`
- Linux operating systems - `/opt/ibm/ldap/V6.0`
- Solaris operating systems - `/opt/IBM/ldap/V6.0`
- Windows operating systems - `<local_drive>:\Program Files\IBM\LDAP\V6.0` (This is the default install location. The actual `IDS_LDAP_HOME` is determined during installation.)

See *IBM Directory C-Client SDK 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. For more information on managing an SSL key database, see "Using gsk7ikm" on page 121. Also see the "Security functions" on page 386 and "Secure Sockets Layer" on page 115 for more information about SSL and certificates.

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/V6.0*
- HP-UX operating systems - */opt/IBM/ldap/V6.0*
- Linux operating systems - */opt/ibm/ldap/V6.0*
- Solaris operating systems - */opt/IBM/ldap/V6.0*
- Windows operating systems - *<local\_drive>:\Program Files\IBM\LDAP\V6.0* (This is the default install location. The actual *IDS\_LDAP\_HOME* is determined during installation.)

See *IBM Directory C-Client SDK 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. For more information on managing an SSL key database, see "Using *gsk7ikm*" on page 121. Also see the "Security functions" on page 386 and "Secure Sockets Layer" on page 115 for more information about SSL and certificates.

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/V6.0
- HP-UX operating systems - /opt/IBM/ldap/V6.0
- Linux operating systems - /opt/ibm/ldap/V6.0
- Solaris operating systems - /opt/IBM/ldap/V6.0
- Windows operating systems - <local\_drive>:\Program Files\IBM\LDAP\V6.0 (This is the default install location. The actual IDS\_LDAP\_HOME is determined during installation.)

See *IBM Directory C-Client SDK 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. For more information on managing an SSL key database, see "Using gsk7ikm" on page 121. Also see the "Security functions" on page 386 and "Secure Sockets Layer" on page 115 for more information about SSL and certificates.

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=ibm,c=us
objectclass: inetOrgPerson
objectclass: organizationalPerson
objectclass: person
objectclass: top
objectclass: ePerson
sn: entry1
cn: testEntry
```

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

### replica.ldif with entries

```
dn: cn=Entry2,o=ibm,c=uschangeType: add
objectclass: inetOrgPerson
objectclass: organizationalPerson
objectclass: person
objectclass: top
objectclass: ePerson
sn: abcd
cn: testEntry
```

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

Run the **idsldapdiff** command:

```
idsldapdiff -b o=ibm,c=us -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=ibm,c=us gets added on Replica server. This entry is on the master server, but was not on the replica server.
2. Entry cn=Entry2,o=ibm,c=us 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=ibm,c=us 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 419.

## 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] [-G realm]
[-h ldaphost] [-help][-K keyfile] [-m mechanism] [-N certificatename]
[-p ldapport] [-P keyfilepw] [-?] [-U username] [-v] [-w passwd | ?]
[-x] [-y proxyDN] [-Y] [-Z]
-op {acctstatus | cascrepl | clearlog | controlqueue | controlrepl |
controlreplerr | evaluategroups | getattributes | getlogsize |
getusertype | quiesce | readconfig | readlog | repltopology |
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 “IANA character sets supported by platform” on page 483 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 “Debugging levels” on page 462 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.

**-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.

**-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/V6.0*
- HP-UX operating systems - */opt/IBM/ldap/V6.0*
- Linux operating systems - */opt/ibm/ldap/V6.0*
- Solaris operating systems - */opt/IBM/ldap/V6.0*
- Windows operating systems - *<local\_drive>:\Program Files\IBM\LDAP\V6.0* (This is the default install location. The actual **IDS\_LDAP\_HOME** is determined during installation.)

See *IBM Directory C-Client SDK 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. For more information on managing an SSL or TLS key database, see "Using gsk7ikm" on page 121. Also see the "Security functions" on page 394 and "Secure Sockets Layer" on page 115 for more information about SSL and certificates.

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.

**-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).

**-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=ibm,c=us
```

- **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=ibm,c=us" -a
departmentNumber=G8R
```

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

**-attrType** {operational | language\_tag | attribute\_cache | unique | configuration}

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
```

- **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) : server_config_administrator directory_administrator
```

See “User type and user roles for extended operations” on page 394 for more information.

- **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=ibm,c=us"
```

- **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
```

- **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 419.

## 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.

## User type and user roles for extended operations

The following are the users and their roles for extended operations.

**Root administrator:** An administrative user whose simple and External with SSL or TLS bind credentials are stored under the cn=Configuration entry. This user's Kerberos bind credentials (optional) are stored under the cn=Kerberos,cn=Configuration entry. This user's Digest-MD5 bind credentials (optional) are stored under the cn=Digest,cn=Configuration entry. In addition, this type of user can bind to the Admin Daemon.

*Roles:*

### Server configuration administrator

This user has unrestricted access to all information in the configuration backend and can start/stop the server. The user can issue dynamic configuration updates.

### Directory administrator

This user has unrestricted access to directory data outside the configuration backend (schema, and RDBM backends). This user can search for one or two attributes in the configuration backend. This user may not have any authority to the operating specific backends (i5/OS system projection backend, z/OS RACF® SDBM).

**Administrative group member:** An administrative user whose simple, External with SSL or TLS, Kerberos (optional), and Digest-MD5 (optional) credentials are



stored under an entry in the subtree `cn=AdminGroup,cn=Configuration`. In addition, this type of user can bind to the Admin Daemon.

*Roles:*

#### **Server configuration group member**

This user has access to all configuration information except the administrator and admin group credentials. This user has the ability to start and stop the server. The user does not have the ability to add or remove members from the administrative group. The user cannot modify the administration password policy. The user is not able to modify the DN, password, Kerberos ID, or Digest-MD5 ID of any administrative group member entry under `cn=AdminGroup,cn=Configuration`. If the user is an Administrative Group Member the user is able to modify his own password, but is not able to modify his own DN, Kerberos ID, or Digest-MD5 ID. This user is also not able to see the password of any other administrative group member or the IBM Tivoli Directory Server administrator.

Members of the administrative group can view the administration daemon audit log and settings but not modify them. Only the administrator is enabled to access, change or clear the administration daemon audit log files.

In addition, this user is not able to add, delete, or modify the audit log setting (the entire `cn=Audit,cn=Configuration` entry) or clear the audit log. The user is not able to add or delete the `cn=Kerberos,cn=Configuration` or `cn=Digest,cn=Configuration` entries, but is able to search all attributes under these entries. The user is able to modify all attributes under these entries except the Kerberos and Digest-MD5 root administrator bind attributes. These users are not able to search or modify the `ibm-slapdAdminDN`, `ibm-slapdAdminGroupEnabled` or `ibm-slapdAdminPW` attributes under the `cn=Configuration` entry. The user can issue dynamic configuration updates.

#### **Directory administrator**

This user has unrestricted access to directory data outside the configuration backend (schema, and RDBM backends). This user can search for one or two attributes in the configuration backend. This user may not have any authority to the operating specific backends (i5/OS system projection backend, z/OS RACF SDBM).

**LDAP user type:** A regular LDAP user whose credentials are stored in the DIT of the LDAP Server. The user's simple and external with SSL or TLS bind DN is the DN of an entry in the DIT. The user's password is stored in the `userpassword` attribute of this entry.

*Roles:*

#### **LDAP User Role**

A user having almost no access to the configuration backend. This user can search for one or two attributes in the configuration backend. The user's access to directory data (schema, and RDBM backends) is controlled by ACLs.

**Global administration group member:** This user has his credentials stored in the same location as the `"ldap_user_type"` and has the same bind DN and password attribute settings. This user differs from the `"ldap_user_type"` in that he belongs to

the global administration group entry that is stored in globalGroupName=globalAdminGroup,cn=ibmpolicies.

*Roles:*

### Directory administrator

This user has unrestricted access to directory data outside the configuration backend (schema, and RDBM backends). This user may not have any authority to the operating specific backends (i5/OS system projection backend, z/OS RACF SDBM).

### See also

idsldapadd, idsldapchangepwd, idsldapdelete, idsldapmodify, idsldapmodrdrn, idsldapsearch

## idsldapmodify, ldapmodify, idsldapadd, ldapadd

The LDAP modify-entry and LDAP add-entry tools

### Synopsis

```
idsldapmodify | ldapmodify [-a] [-b] [-c] [-C charset] [-d debuglevel][-D binddn]
[-e errorfile] [-f file] [-g] [-G realm] [-h ldaphost] [-i file]
[-k] [-K keyfile] [-l] [-m mechanism] [-M] [-n] [-N certificatename]
[-O maxhops] [-p ldapport] [-P keyfilepw] [-r] [-R] [-U username]
[-v] [-V] [-w passwd | ?] [-x] [-y proxydn] [-Y] [-Z]
```

```
idsldapadd | ldapadd [-a] [-b] [-c] [-C charset] [-d debuglevel][-D binddn]
[-e errorfile] [-f file] [-g] [-G realm] [-h ldaphost] [-i file]
[-k] [-K keyfile] [-l] [-m mechanism] [-M] [-n] [-N certificatename]
[-O maxhops] [-p ldapport] [-P keyfilepw] [-r] [-R] [-U username]
[-v] [-V] [-w passwd | ?] [-x] [-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 value.
- 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 "IANA character sets supported by platform" on page 483 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 "Debugging levels" on page 462 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.

**-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.

**-k** Specifies to use server administration control.

This option sends the Server administration control. See "Server administration control" in the *IBM Tivoli Directory Server C-Client SDK 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/V6.0`
- HP-UX operating systems - `/opt/IBM/ldap/V6.0`
- Linux operating systems - `/opt/ibm/ldap/V6.0`
- Solaris operating systems - `/opt/IBM/ldap/V6.0`
- Windows operating systems - `<local_drive>:\Program Files\IBM\LDAP\V6.0` (This is the default install location. The actual `IDS_LDAP_HOME` is determined during installation.)

See the *IBM Directory C-Client SDK 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. For more information on managing an SSL or TLS key database, see "Using `gsk7ikm`" on page 121. Also see the "Security functions" on page 402 and "Secure Sockets Layer" on page 115 for more information about SSL and certificates.

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

**-l** Do not replicate the entry.

This option sends the Do not replicate control. See "Do not replicate control" in the *IBM Tivoli Directory Server C-Client SDK Programming Reference*.

**-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 *ldappport*  
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.
- r Replace existing values by default.
- R Specifies that referrals are not to be automatically followed.
- 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 **idslldapmodify** 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 **idslldapmodify**, 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).
- 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 419.

## 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] [-f file]
[-G realm] [-h ldaphost] [-i file] [-k] [-K keyfile] [-l]
[-m mechanism] [-M] [-n] [-N certificatename] [-O hopcount]
[-p ldapport] [-P keyfilepw] [-r] [-R] [-s newSuperior]
[-U username] [-v] [-V] [-w passwd | ?] [-x] [-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 DN (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 “IANA character sets supported by platform” on page 483 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 “Debugging levels” on page 462 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:".

**-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*").

**-k** Specifies to use server administration control.

This option sends the Server administration control. See "Server administration control" in the *IBM Tivoli Directory Server C-Client SDK 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/V6.0
- HP-UX operating systems - /opt/IBM/ldap/V6.0
- Linux operating systems - /opt/ibm/ldap/V6.0
- Solaris operating systems - /opt/IBM/ldap/V6.0
- Windows operating systems - *<local\_drive>*:\Program Files\IBM\LDAP\V6.0 (This is the default install location. The actual IDS\_LDAP\_HOME is determined during installation.)

See *IBM Directory C-Client SDK 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. For more information on managing an SSL or TLS key database, see "Using gsk7ikm" on page 121. Also see the "Security functions" on page 406 and "Secure Sockets Layer" on page 115 for more information about SSL and certificates.

This parameter effectively enables the -Z switch.

**-l** Do not replicate the entry.

This option sends the Do not replicate control. See "Do not replicate control" in the *IBM Tivoli Directory Server C-Client SDK Programming Reference*.

- 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 don't actually 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** *ldappport*  
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.
- 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 ""**).
- 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).

**-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=IBM,c=US" "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 419.

## 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 charset]
[-d debuglevel] [-D binddn] [-f file] [-F sep] [-G realm] [-h ldaphost] [-i file]
[-k] [-K keyfile] [-l timelimit] [-L] [-m mechanism] [-M] [-n] [-N certificatename]
[-o attr_type] [-O maxhops] [-p ldapport] [-P keyfilepw] [-q pagesize]
[-R] [-s scope] [-t] [-T seconds] [-U username] [-v] [-V version]
[-w passwd | ?] [-x] [-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 `ldap_search` in the *IBM Tivoli Directory Server Version 6.0 C-Client SDK 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.

### 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 **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 "IANA character sets supported by platform" on page 483 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 "Debugging levels" on page 462 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.
- 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=ibm,c=us" -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=ibm,c=us` 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.

**-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/V6.0`
- HP-UX operating systems - `/opt/IBM/ldap/V6.0`
- Linux operating systems - `/opt/ibm/ldap/V6.0`
- Solaris operating systems - `/opt/IBM/ldap/V6.0`
- Windows operating systems - `<local_drive>:\Program Files\IBM\LDAP\V6.0` (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 C-Client SDK 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. For more information on managing an SSL or TLS key database, see "Using gsk7ikm" on page 121. Also see the "Security functions" on page 415 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 don't actually 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 C-Client SDK 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 C-Client SDK Programming Reference*.

**-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.
- 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 `idslldapmodify` 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 `idslldapmodify`, 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).
- 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.

## 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: UF1
description: Shaper of young minds
```

...

This command:

```
idsldapsearch -b "o=ibm,c=us" -o ibm-slapdDN "objectclass=person" ibm-slapdDN
```

performs a subtree level search at the o=ibm,c=us 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=A1 Edwards,ou=Widget Division,ou=Austin,o=IBM,c=US
cn=A1 Garcia,ou=Home Entertainment,ou=Austin,o=IBM,c=US
cn=Amy Nguyen,ou=In Flight Systems,ou=Austin,o=IBM,c=US
cn=Arthur Edwards,ou=Widget Division,ou=Austin,o=IBM,c=US
cn=Becky Garcia,ou=In Flight Systems,ou=Austin,o=IBM,c=US
cn=Ben Catu,ou=In Flight Systems,ou=Austin,o=IBM,c=US
cn=Ben Garcia Jr,ou=Home Entertainment,ou=Austin,o=IBM,c=US
cn=Bill Keller Jr.,ou=In Flight Systems,ou=Austin,o=IBM,c=US
cn=Bob Campbell,ou=In Flight Systems,ou=Austin,o=IBM,c=US
```

This command:

```
idsldapsearch -b "o=ibm,c=us" -s base "objectclass=*" numSubordinates
```

performs a one- level search at the o=ibm,c=us 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=IBM,c=US  
numSubordinates=2

## Security functions

To use the SSL or TLS -related functions associated with this utility, see “SSL, TLS notes” on page 419.

## 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 -h hostname -K keyfile -m debugLevel
-N key_name -o debugFile -p port -P key_pw -t [start|stop]
-v -w adminPW[? -x -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.

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

**-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/V6.0`
- HP-UX operating systems - `/opt/IBM/ldap/V6.0`
- Linux operating systems - `/opt/ibm/ldap/V6.0`
- Solaris operating systems - `/opt/IBM/ldap/V6.0`
- Windows operating systems - `<local_drive>:\Program Files\IBM\LDAP\V6.0` (This is the default install location. The actual `IDS_LDAP_HOME` is determined during installation.)

See *IBM Directory C-Client SDK 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. For more information on managing an SSL or TLS key database, see "Using `gsk7ikm`" on page 121. Also see the "Security functions" on page 394 and "Secure Sockets Layer" on page 115 for more information about SSL and certificates.

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.

**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 Unix-based 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 “Debugging levels” on page 462 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.

**-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).

**-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 -- -| 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 460

## 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 "LDAP\_SSL" in the *IBM Directory C-Client SDK 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. For more information on this Java utility, see "Using `gsk7ikm`" on page 121. 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.

---

## Server utilities

This sections describes the server utilities.

### Notes:

1. Except for the `idsdb2ldif` utility, the server must be stopped before using the server utilities.
2. Ensure that no applications are attached to the directory database. If there are applications attached, none of the server utilities will run.

**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 I, “Synchronizing two-way cryptography between server instances,” on page 537 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 LDIF file is split into multiple output files based on a hash function.

`ddsetup` uses its own configuration file to specify output filenames and the desired partitioning characteristics.

### Synopsis

```
ddsetup [-I instancename [-r] [-f configfile] [-d debug level]] | -?
```

### Options

`-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 “Debugging levels” on page 462 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.
- r  
Specifies to run the tool in recovery mode.
- ?  
Displays the syntax format.

## 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. You need to export the entries to an LDIF file so that the entries can be distributed among the five back-end servers. See “idsdb2ldif, db2ldif” on page 436 for information on how to do this.

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

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

2. Create a configuration file, for example **newhash.conf**, where **newhash.conf** contains:

```
BaseDirectory: <BaseDirectory>
Action type : splitonly
inputFile : <pathname>/mydata.ldif
defaultOutputFile : default.ldif
SplitBY: rdnhash
```

```
BaseDN: o=ibm,c=US
URL: ldap://ServerA:389
FILE: out1.ldif
URL: ldap://ServerB:169
File: out2.ldif
URL: ldap://ServerC:389
File: out3.ldif
URL: ldap://ServerD:389
File: out4.ldif
URL: ldap://ServerE:389
File: out5.ldif
```

**Note:** If you specify the `BaseDirectory` parameter, it must come before any files in the configuration file. This includes the default ldif output file, the logfile, and any other output LDIF files.

3. Issue the command:

```
ddsetup -f ~/newhash.conf
```

The `ddsetup` command divides the `mydata.ldif` file into multiple LDIF output files. 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.

4. 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. Create a configuration file, for example **newhash.conf**. Where newhash.conf contains:

```
Action type : splitonly
inputFile : <pathname>/mydata.ldif
defaultOutputFile : default.ldif
SplitBY: rdncore
```

```
BaseDN: ou=austin,o=ibm,c=US
URL: ldap://ServerA:389
FILE: out1_ServerA.ldif
URL: ldap://ServerB:169
File: out2_ServerB.ldif
URL: ldap://ServerC:1389
File: out3_ServerC.ldif
URL: ldap://ServerD:2389
File: out4_ServerD.ldif
URL: ldap://ServerE:389
File: out5_ServerE.ldif
```

```
BaseDN: ou=raleigh,o=ibm,c=US
URL: ldap://ServerF:389
FILE: out1_ServerF.ldif
URL: ldap://ServerG:169
File: out2_ServerG.ldif
URL: ldap://ServerH:389
File: out3_ServerH.ldif
```

```
BaseDN: ou=poughkeepsie,o=ibm,c=US
URL: ldap://ServerI:389
FILE: out1_ServerI.ldif
URL: ldap://ServerJ:2169
File: out2_ServerJ.ldif
URL: ldap://ServerK:389
File: out3_ServerK.ldif
URL: ldap://ServerL:3389
File: out4_ServerL.ldif
```

**Note:** When splitting data among servers, giving the output files a name associated with the server and the partition index value might make it easier for you to load the correct files to the correct back-end servers.

3. Issue the command:  
`ddsetup -f ~/newhash.conf`

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.

4. 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.

## 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 I, “Synchronizing two-way cryptography between server instances,” on page 537 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 Version 6.0. 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>] [-D <yes|no>] [-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 acs.

**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 “Debugging levels” on page 462 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**.

**-D <yes|no>**

Specifies whether to drop the indexes before the load. The default is **no**.

- 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/V6.0
  - HP-UX operating systems - /opt/IBM/ldap/V6.0
  - Linux operating systems - /opt/ibm/ldap/V6.0
  - Solaris operating systems - /opt/IBM/ldap/V6.0
  - Windows operating systems - <local\_drive>:\Program Files\IBM\LDAP\V6.0 (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
  - Windows operating systems in <ITDS home directory>\idsslapd-<directory server instance name>\tmp\ldapimport
- n | -N  
Specifies that the load is nonrecoverable. With this option, **idsbulkload** uses less disk space 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/sql1lib/tmp/".
```

you need to 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 `<ITDS home directory>\idsslapd-<directory server instance name>\tmp\ldapimport`
- AIX, Linux, Solaris, and HP-UX operating systems in `<ITDS 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.

**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 “Debugging levels” on page 462 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:

```
idscfgchlg -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:

```
idscfgchlg -y 1 -h -l
```

**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 a database for a directory server instance. The database instance must already exist and for UNIX and Linux platforms, the local loopback service must be registered in the `/etc/services` file. Otherwise, the command fails.

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 “Debugging levels” on page 462 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 UNIX or Linux 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 “Debugging levels” on page 462 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 directory server instance is a proxy server or 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 “Debugging levels” on page 462 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=ibm,c=us` on a machine with a single directory server instance, issue the command:

```
idscfgsuf -s o=ibm,c=us
```

To configure the suffix `o=ibm,c=us` on a machine with a multiple directory server instances, issue the command:

```
idscfgsuf -I <instancename> -s o=ibm,c=us
```

## 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.

### 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 “Debugging levels” on page 462 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
```

## 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 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 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 might also need to do 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 need to do 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.

## 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 “Debugging levels” on page 462 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 I, “Synchronizing two-way cryptography between server instances,” on page 537 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 D, “ASCII characters from 33 to 126,” on page 487 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]
 [-k ?|keyseed -t keysalt] [-j]
 [[-s subtreeDN [-x]] | [-p on|off] [-l]] [-W] [-x]] |
 -?
```

## Options

All options are case sensitive.

**-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 D, “ASCII characters from 33 to 126,” on page 487. 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.
- 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. 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 D, "ASCII characters from 33 to 126," on page 487. Use the `?` to generate a password prompt. Using this prompt prevents your encryption salt from being visible through the **ps** command. 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]
 [[[-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 “Debugging levels” on page 462 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 UNIX and Linux systems only:

**-c** Specifies to run the server in console mode.

## 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 V6.0 - *<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 V6.0 - *<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 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 “Debugging levels” on page 462 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. See "Summary of settings for an EAL4 secure configuration" on page 134 and "Setting the administration password and lockout policy" on page 136.

## 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-slapdCryptoSync`. These two values regenerate the instance's `ibmslapddir.ksf` 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 "Debugging levels" on page 462 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 D, "ASCII characters from 33 to 126," on page 487. 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

See "Password policy configuration and administrator password" in the *IBM Tivoli Directory Server Common Criteria Guide Version 6.0*.

- 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 D, "ASCII characters from 33 to 126," on page 487.
- 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:

```
idsgendirsf -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] [-p port] [-s secureport] [-a admport]
 [-t dbinstance] [-c admsecureport] [-i ipaddress] [-l instlocation]
 [-r description] [-C] [-d debuglevel] [-b outputfile] [-q] [-n]] | -v | -?
```

### Description

The `idsicrt` command can only be run by root on UNIX or Linux 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. 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 UNIX or Linux 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 exist 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.

## 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 "Debugging levels" on page 462 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 D, "ASCII characters from 33 to 126," on page 487.

**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

See "Password policy configuration and administrator password" in the *IBM Tivoli Directory Server Common Criteria Guide Version 6.0*.

- 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.
- ?  
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 mysecretseed, and a DB2 instance with the name of myinst, issue the command:

```
idsicrt -I myinst -p 389 -s 636 -e mysecretseed
```

If the directory server instance already existed, this command fails. If you did not specify the encryption seed you are 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 in the encryption seed and press Enter, the command attempts to create the directory server instance.

```
idsicrt -I myinst -p 389 -s 636
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 mysecretseed -h 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 mysecretseed, and a DB2 instance with the name of mydbin, then issue the command:

```
idsicrt -I myinst -p 389 -s 636 -e mysecretseed -t mydbin
```

**Note:** The idsicrt command creates a DB2 database instance, only if the server component (RDBM) is installed.

When created, a database instance normally requires 10 to 20 MB of space. This space is not used, however, if the directory server instance is configured as a proxy server.

## 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 “Debugging levels” on page 462 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 “Debugging levels” on page 462 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
```

The output is:

Instance 1:

```
Name: myinst1
Version: 6.0
Location: c:
Description: IBM Tivoli Directory Server Instance V6.0
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: 6.0
Location: c:
Description: IBM Tivoli Directory Server Instance V6.0
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#6.0#c:#IBM Tivoli Directory Server Instance V6.0#All available
#389#636#3538#3539#Directory Server
myinst2#6.0#c:#IBM Tivoli Directory Server Instance V6.0#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

For information about the idsimigr utility, see "The command-line migration utility (idsimigr)" in the *IBM Tivoli Directory Server Version 6.0 Installation and Configuration Guide*.

## idsldif2db, ldif2db

Command to load LDIF file entries into a database.

**Synopsis**

```
idsldif2db | ldif2db [-i inputfile -I instancename [-f configfile]
[-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 a 6.0 server 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 I, "Synchronizing two-way cryptography between server instances," on page 537 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.

**-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 UNIX or Linux 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 6.0 Installation and Configuration Guide* for more information about installing IBM Tivoli Directory Integrator.

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. See “Log management tool” on page 268 for more information.

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/V6.0`
- 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/V6.0`
- 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.

## idslink

For information about the `idslink` utility, see "logidslink log on AIX, Linux, and Solaris operating systems" in the *IBM Tivoli Directory Server Version 6.0 Problem Determination Guide*.

## 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 “Debugging levels” on page 462 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 "Debugging levels" on page 462 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 `idslldapmodify` 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 "Debugging levels" on page 462 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 UNIX or Linux 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]
 [[[-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 “Debugging levels” on page 462 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 daemon.
- 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 UNIX and Linux systems only.

- c** Specifies to run the server in console mode.

### 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 “Using the command line – idssnmp” on page 494.

## idssupport

For information about the idssupport utility, see “IBM Tivoli Directory Server Support Tool” in the *IBM Tivoli Directory Server Version 6.0 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 “Debugging levels” on page 462 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 “Debugging levels” on page 462 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 schemafilename] [-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 "Debugging levels" on page 462 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** *<schemafilename>*  
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`



## 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 directory server instance is a proxy server or 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 "Debugging levels" on page 462 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=ibm,c=us` from the `ibmslapd.conf` file on a machine with a single directory server instance, issue the command:

```
idsucfgsuf -s o=ibm,c=us
```

To remove the suffix `o=ibm,c=us` 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=ibm,c=us
```

**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 Unix-based 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.

#### 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 (maxSeverErrors) 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.

**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
```

## runscript

The runscript command is used by UNIX and Linux 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.

---

## 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 15. 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                        |

Table 15. Debug categories (continued)

| Hex    | Decimal | Value                  | Description                               |
|--------|---------|------------------------|-------------------------------------------|
| 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.



---

## Part 6. Appendixes





## Appendix A. Error codes

The possible values for an LDAP error code are shown in the following tables:

Table 16. General return codes

| Dec value | Value                               | Hex value | Brief description                     | Detailed description                                      |
|-----------|-------------------------------------|-----------|---------------------------------------|-----------------------------------------------------------|
| 00        | LDAP_SUCCESS                        | 00        | Success                               | The request was successful.                               |
| 01        | LDAP_OPERATIONS_ERROR               | 01        | Operations error                      | An operations error occurred.                             |
| 02        | LDAP_PROTOCOL_ERROR                 | 02        | Protocol error                        | A protocol violation was detected.                        |
| 03        | LDAP_TIMELIMIT_EXCEEDED             | 03        | Time limit exceeded                   | An LDAP time limit was exceeded.                          |
| 04        | LDAP_SIZELIMIT_EXCEEDED             | 04        | Size limit exceeded                   | An LDAP size limit was exceeded.                          |
| 05        | LDAP_COMPARE_FALSE                  | 05        | Compare false                         | A compare operation returned false.                       |
| 06        | LDAP_COMPARE_TRUE                   | 06        | Compare true                          | A compare operation returned true.                        |
| 07        | LDAP_STRONG_AUTH_NOT_SUPPORTED      | 07        | Strong authentication not supported   | The LDAP server does not support strong authentication.   |
| 08        | LDAP_STRONG_AUTH_REQUIRED           | 08        | Strong authentication required        | Strong authentication is required for the operation.      |
| 09        | LDAP_PARTIAL_RESULTS                | 09        | Partial results and referral received | Partial results only returned.                            |
| 10        | LDAP_REFERRAL                       | 0A        | Referral returned                     | Referral returned.                                        |
| 11        | LDAP_ADMIN_LIMIT_EXCEEDED           | 0B        | Administration limit exceeded         | Administration limit exceeded.                            |
| 12        | LDAP_UNAVAILABLE_CRITICAL_EXTENSION | 0C        | Critical extension not supported      | Critical extension is not supported.                      |
| 13        | LDAP_CONFIDENTIALITY_REQUIRED       | 0D        | Confidentiality is required           | Confidentiality is required.                              |
| 14        | LDAP_SASLBIND_IN_PROGRESS           | 0E        | SASL bind in progress                 | An SASL bind is in progress.                              |
| 16        | LDAP_NO_SUCH_ATTRIBUTE              | 10        | No such attribute                     | The attribute type specified does not exist in the entry. |
| 17        | LDAP_UNDEFINED_TYPE                 | 11        | Undefined attribute type              | The attribute type specified is not valid.                |
| 18        | LDAP_INAPPROPRIATE_MATCHING         | 12        | Inappropriate matching                | Filter type not supported for the specified attribute.    |

Table 16. General return codes (continued)

| Dec value | Value                       | Hex value | Brief description            | Detailed description                                                                                                                           |
|-----------|-----------------------------|-----------|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| 19        | LDAP_CONSTRAINT_VIOLATION   | 13        | Constraint violation         | An attribute value specified violates some constraint (for example, a postal address has too many lines, or a line that is too long).          |
| 20        | LDAP_TYPE_OR_VALUE_EXISTS   | 14        | Type or value exists         | An attribute type or attribute value specified already exists in the entry.                                                                    |
| 21        | LDAP_INVALID_SYNTAX         | 15        | Invalid syntax               | An attribute value that is not valid was specified.                                                                                            |
| 32        | LDAP_NO_SUCH_OBJECT         | 20        | No such object               | The specified object does not exist in the directory.                                                                                          |
| 33        | LDAP_ALIAS_PROBLEM          | 21        | Alias problem                | An alias in the directory points to a nonexistent entry.                                                                                       |
| 34        | LDAP_INVALID_DN_SYNTAX      | 22        | Invalid DN syntax            | A DN that is syntactically not valid was specified.                                                                                            |
| 35        | LDAP_IS_LEAF                | 23        | Object is a leaf             | The object specified is a leaf.                                                                                                                |
| 36        | LDAP_ALIAS_DEREF_PROBLEM    | 24        | Alias dereferencing problem  | A problem was encountered when dereferencing an alias.                                                                                         |
| 48        | LDAP_INAPPROPRIATE_AUTH     | 30        | Inappropriate authentication | Inappropriate authentication was specified (for example, LDAP_AUTH_SIMPLE was specified and the entry does not have a userPassword attribute). |
| 49        | LDAP_INVALID_CREDENTIALS    | 31        | Invalid credentials          | Invalid credentials were presented (for example, the wrong password).                                                                          |
| 50        | LDAP_INSUFFICIENT_ACCESS    | 32        | Insufficient access          | The user has insufficient access to perform the operation.                                                                                     |
| 51        | LDAP_BUSY                   | 33        | DSA is busy                  | The DSA is busy.                                                                                                                               |
| 52        | LDAP_UNAVAILABLE            | 34        | DSA is unavailable           | The DSA is unavailable.                                                                                                                        |
| 53        | LDAP_UNWILLING_TO_PERFORM   | 35        | DSA is unwilling to perform  | The DSA is unwilling to perform the operation.                                                                                                 |
| 54        | LDAP_LOOP_DETECT            | 36        | Loop detected                | A loop was detected.                                                                                                                           |
| 64        | LDAP_NAMING_VIOLATION       | 40        | Naming violation             | A naming violation occurred.                                                                                                                   |
| 65        | LDAP_OBJECT_CLASS_VIOLATION | 41        | Object class violation       | An object class violation occurred (for example, a "required" attribute was missing from the entry).                                           |

Table 16. General return codes (continued)

| Dec value | Value                       | Hex value | Brief description                | Detailed description                                                                    |
|-----------|-----------------------------|-----------|----------------------------------|-----------------------------------------------------------------------------------------|
| 66        | LDAP_NOT_ALLOWED_ON_NONLEAF | 42        | Operation not allowed on nonleaf | The operation is not allowed on a nonleaf object.                                       |
| 67        | LDAP_NOT_ALLOWED_ON_RDN     | 43        | Operation not allowed on RDN     | The operation is not allowed on an RDN.                                                 |
| 68        | LDAP_ALREADY_EXISTS         | 44        | Already exists                   | The entry already exists.                                                               |
| 69        | LDAP_NO_OBJECT_CLASS_MODS   | 45        | Cannot modify object class       | Object class modifications are not allowed.                                             |
| 70        | LDAP_RESULTS_TOO_LARGE      | 46        | Results too large                | Results too large.                                                                      |
| 71        | LDAP_AFFECTS_MULTIPLE_DSAS  | 47        | Affects multiple DSAs            | Affects multiple DSAs.                                                                  |
| 80        | LDAP_OTHER                  | 50        | Unknown error                    | An unknown error occurred.                                                              |
| 81        | LDAP_SERVER_DOWN            | 51        | Can't contact LDAP server        | The LDAP library cannot contact the LDAP server.                                        |
| 82        | LDAP_LOCAL_ERROR            | 52        | Local error                      | Some local error occurred. This is usually a failed memory allocation.                  |
| 83        | LDAP_ENCODING_ERROR         | 53        | Encoding error                   | An error was encountered encoding parameters to send to the LDAP server.                |
| 84        | LDAP_DECODING_ERROR         | 54        | Decoding error                   | An error was encountered decoding a result from the LDAP server.                        |
| 85        | LDAP_TIMEOUT                | 55        | Timed out                        | A time limit was exceeded while waiting for a result.                                   |
| 86        | LDAP_AUTH_UNKNOWN           | 56        | Unknown authentication method    | The authentication method specified on a bind operation is not known.                   |
| 87        | LDAP_FILTER_ERROR           | 57        | Bad search filter                | An invalid filter was supplied to ldap_search (for example, unbalanced parentheses).    |
| 88        | LDAP_USER_CANCELLED         | 58        | User cancelled operation         | The user cancelled the operation.                                                       |
| 89        | LDAP_PARAM_ERROR            | 59        | Bad parameter to an LDAP routine | An LDAP routine was called with a bad parameter (for example, a NULL ld pointer, etc.). |
| 90        | LDAP_NO_MEMORY              | 5A        | Out of memory                    | A memory allocation (for example malloc) call failed in an LDAP library routine.        |
| 91        | LDAP_CONNECT_ERROR          | 5B        | Connection error                 | Connection error.                                                                       |

Table 16. General return codes (continued)

| Dec value | Value                           | Hex value | Brief description                                                      | Detailed description                                                                                             |
|-----------|---------------------------------|-----------|------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|
| 92        | LDAP_NOT_SUPPORTED              | 5C        | Not supported                                                          | Not supported.                                                                                                   |
| 93        | LDAP_CONTROL_NOT_FOUND          | 5D        | Control not found                                                      | Control not found.                                                                                               |
| 94        | LDAP_NO_RESULTS_RETURNED        | 5E        | No results returned                                                    | No results returned.                                                                                             |
| 95        | LDAP_MORE_RESULTS_TO_RETURN     | 5F        | More results to return                                                 | More results to return.                                                                                          |
| 96        | LDAP_URL_ERR_NOTLDAP            | 60        | URL doesn't begin with ldap://                                         | The URL does not begin with ldap://.                                                                             |
| 97        | LDAP_URL_ERR_NODN               | 61        | URL has no DN (required)                                               | The URL does not have a DN (required).                                                                           |
| 98        | LDAP_URL_ERR_BADSCOPE           | 62        | URL scope string is invalid                                            | The URL scope string is not valid.                                                                               |
| 99        | LDAP_URL_ERR_MEM                | 63        | Can't allocate memory space                                            | Cannot allocate memory space.                                                                                    |
| 100       | LDAP_CLIENT_LOOP                | 64        | Client loop                                                            | Client loop.                                                                                                     |
| 101       | LDAP_REFERRAL_LIMIT_EXCEEDED    | 65        | Referral limit exceeded                                                | Referral limit exceeded.                                                                                         |
| 112       | LDAP_SSL_ALREADY_INITIALIZED    | 70        | ldap_ssl_client_init successfully called previously in this process    | The ldap_ssl_client_init was successfully called previously in this process.                                     |
| 113       | LDAP_SSL_INITIALIZE_FAILED      | 71        | Initialization call failed                                             | SSL Initialization call failed.<br><b>Note:</b> GSKit must be installed and the GSKit libraries must be present. |
| 114       | LDAP_SSL_CLIENT_INIT_NOT_CALLED | 72        | Must call ldap_ssl_client_init before attempting to use SSL connection | Must call ldap_ssl_client_init before attempting to use SSL connection.                                          |
| 115       | LDAP_SSL_PARAM_ERROR            | 73        | Invalid SSL parameter previously specified                             | An SSL parameter that was not valid was previously specified.                                                    |
| 116       | LDAP_SSL_HANDSHAKE_FAILED       | 74        | Failed to connect to SSL server                                        | Failed to connect to SSL server.                                                                                 |
| 117       | LDAP_SSL_GET_CIPHER_FAILED      | 75        | Not used                                                               | Deprecated.                                                                                                      |
| 118       | LDAP_SSL_NOT_AVAILABLE          | 76        | SSL library cannot be located                                          | Ensure that GSKit has been installed.                                                                            |
|           | LDAP_SSL_KEYRING_NOT_FOUND      | 77        |                                                                        |                                                                                                                  |
|           | LDAP_SSL_PASSWORD_NOT_SPECIFIED | 78        |                                                                        |                                                                                                                  |
| 128       | LDAP_NO_EXPLICIT_OWNER          | 80        | No explicit owner found                                                | No explicit owner was found.                                                                                     |
| 129       | LDAP_NO_LOCK                    | 81        | Could not obtain lock                                                  | Client library was not able to lock a required resource.                                                         |

In addition, the following DNS-related error codes are defined in the ldap.h file:

Table 17. DNS-related return codes

| Dec value | Value                    | Hex value | Detailed description                      |
|-----------|--------------------------|-----------|-------------------------------------------|
| 133       | LDAP_DNS_NO_SERVERS      | 85        | No LDAP servers found                     |
| 134       | LDAP_DNS_TRUNCATED       | 86        | Warning: truncated DNS results            |
| 135       | LDAP_DNS_INVALID_DATA    | 87        | Invalid DNS Data                          |
| 136       | LDAP_DNS_RESOLVE_ERROR   | 88        | Can't resolve system domain or nameserver |
| 137       | LDAP_DNS_CONF_FILE_ERROR | 89        | DNS Configuration file error              |

The following UTF8-related error codes are defined in the ldap.h file:

Table 18. UTF8-related return codes

| Dec value | Value                                 | Hex value | Detailed description       |
|-----------|---------------------------------------|-----------|----------------------------|
| 160       | LDAP_XLATE_E2BIG                      | A0        | Output buffer overflow     |
| 161       | LDAP_XLATE_EINVAL                     | A1        | Input buffer truncated     |
| 162       | LDAP_XLATE_EILSEQ                     | A2        | Unusable input character   |
| 163       | LDAP_XLATE_NO_ENTRY                   | A3        | No codeset point to map to |
|           | LDAP_REG_FILE_NOT_FOUND               | B0        |                            |
|           | LDAP_REG_CANNOT_OPEN                  | B1        |                            |
|           | LDAP_REG_ENTRY_NOT_FOUND              | B2        |                            |
|           | LDAP_CONF_FILE_NOT_OPENED             | C0        |                            |
|           | LDAP_PLUGIN_NOT_LOADED                | C1        |                            |
|           | LDAP_PLUGIN_FUNCTION_<br>NOT_RESOLVED | C2        |                            |
|           | LDAP_PLUGIN_NOT_INITIALIZED           | C3        |                            |
|           | LDAP_PLUGIN_COULD_NOT_BIND            | C4        |                            |
|           | LDAP_SASL_GSS_NO_SEC_CONTEXT          | D0        |                            |



---

## Appendix B. Object Identifiers (OIDs) and attributes in the root DSE

The OIDs and attributes shown in the following sections are used in IBM Tivoli Directory Server 6.0. These OIDs and attributes are in the root DSE. The root DSE entry contains information about the server itself.

IBM Tivoli Directory Server defines a root DSE entry that an LDAP server provides to supply you with information about the LDAP server. For example, you might want to know what version of LDAP a server supports.

To list the OIDs and attributes in the root DSE, run the following command:

```
idsldapsearch -D <AdminDN> -w <Adminpw> -s base
-b "" objectclass=*
```

For more detailed information, see the *IBM Tivoli Directory Server Version 6.0 C-Client SDK Programming Reference*.

---

### Attributes in the root DSE

The following attributes are in the root DSE:

#### **namingcontexts**

The naming contexts held in the server.

The values of this attribute correspond to the naming contexts that this server masters or shadows. If the server does not master or shadow any information (for example, it is an LDAP gateway to a public X.500 directory), this attribute is absent. If the server believes it contains the entire directory, the attribute has a single value, and that value is an empty string (indicating the null DN of the root). This allows a client to choose suitable base objects for searching when it has contacted a server (the list of highest level suffixes the user defines in the configuration).

#### **ibm-configurationnamingcontext**

The suffix where the server's configuration entries are stored. For version 6.0 this is cn=configuration.

#### **subschemasubentry**

The value of this attribute is the name of a subschema entry in which the server makes available attributes specifying the schema. It is set to cn=schema.

#### **security**

The secure SSL port the server is listening on. For example 636.

**port** The nonsecure port the server is listening on. For example 389. This is only present only if the server does not have a secure port enabled.

#### **supportedsaslmmechanisms**

A list of supported SASL security features.

The values of this attribute are the names of supported SASL mechanisms that the server supports. If the server does not support any mechanisms then this attribute is absent. This attribute contains any SASL mechanism that is registered to the server.

**supportedldapversion**

LDAP versions implemented by the current server.

The values of this attribute are the versions of the LDAP protocol that the server implements. The values are 2 and 3.

**ibmdirectoryversion**

The version of IBM Tivoli Directory Server installed on this server. The current version is 6.0.

**ibm-enabledcapabilities**

Lists the server capabilities currently enabled on the server. See "OIDs for supported and enabled capabilities" on page 475 for the values.

**ibm-ldapservicename**

Specifies the host name of the server. If a Kerberos realm is defined, the form is hostname@realmname.

**ibm-serverId**

The unique ID assigned to the server at the initial startup of the server. This ID is used in replication topology to determine a server's role.

**vendorname**

The supplier of this version of LDAP. For IBM Tivoli Directory Server, this is set to International Business Machines (IBM).

**vendorversion**

For IBM Tivoli Directory Server 6.0, the vendor version is set to 6.0.

**ibm-slapedSizeLimit**

Limits the number of entries returned by a search initiated by nonadministrative users.

**ibm-slapedTimeLimit**

Specifies in seconds the maximum amount of time the server spends processing a search request initiated by nonadministrative users.

**ibm-slapedDerefAliases**

Describes how the server is configured to handle dereferencing.

**ibm-supportedAuditVersion**

The supported version of auditing. For example, in version 6.0 the server supports auditing version 3 that enables auditing of extended operations.

**ibm-supportedACIMechanisms**

Lists the ACL models the server supports. See "OIDs for ACI mechanisms" on page 477 for the values.

**ibm-supportedcapabilities**

Lists the server capabilities currently supported by the server. See "OIDs for supported and enabled capabilities" on page 475 for the values.

**ibm-sasldigestrealmname****ibm-slapedServerInstanceName**

Name of the directory server instance running on the server.

**ibm-slapedisconfigurationmode**

Identifies whether the server is running in configuration mode. If TRUE, the server is in configuration mode. If FALSE, the server is not in configuration mode.



## OIDs for supported and enabled capabilities

The following table shows OIDs for supported and enabled capabilities. You can use these OIDs to see if a particular server supports these features.

Table 19. OIDs for supported and enabled capabilities

| Short name                                 | Description                                                                                                                                  | OID assigned     |
|--------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| Enhanced Replication Model                 | Identifies the replication model introduced in IBM Directory Server v5.1 including subtree and cascading replication.                        | 1.3.18.0.2.32.1  |
| Entry Checksum                             | Indicates that this server supports the <code>ibm-entrychecksum</code> and <code>ibm-entrychecksumop</code> features.                        | 1.3.18.0.2.32.2  |
| Entry UUID                                 | This value is listed in the <code>ibm-capabilities</code> Subentry for those suffixes that support the <code>ibm-entryuuid</code> attribute. | 1.3.18.0.2.32.3  |
| Filter ACLs                                | Identifies that this server supports the IBM Filter ACL model                                                                                | 1.3.18.0.2.32.4  |
| Password Policy                            | Identifies that this server supports password policies                                                                                       | 1.3.18.0.2.32.5  |
| Sort by DN                                 | Enables searches sorted by DNs in addition to regular attributes.                                                                            | 1.3.18.0.2.32.6  |
| Administration Group Delegation            | Server supports the delegation of server administration to a group of administrators that are specified in the configuration backend.        | 1.3.18.0.2.32.8  |
| Denial of Service Prevention               | Server supports the denial of service prevention feature, including read/write time-outs and the emergency thread.                           | 1.3.18.0.2.32.9  |
| Dereference Alias Option                   | Server supports an option to not dereference aliases by default                                                                              | 1.3.18.0.2.32.10 |
| Admin Daemon Audit Logging                 | Server supports the auditing of the admin daemon.                                                                                            | 1.3.18.0.2.32.11 |
| 128 Character Table Names                  | The server feature to allow name of unique attributes to be higher than 18 characters (with the maximum of 128 characters).                  | 1.3.18.0.2.32.12 |
| Attribute Caching Search Filter Resolution | The server supports attribute caching for search filter resolution.                                                                          | 1.3.18.0.2.32.13 |
| Dynamic Tracing                            | Server supports active tracing for the server with an LDAP extended operation.                                                               | 1.3.18.0.2.32.14 |
| Entry And Subtree Dynamic Updates          | The server supports dynamic configuration updates on entries and subtrees.                                                                   | 1.3.18.0.2.32.15 |
| Globally Unique Attributes                 | The server feature to enforce globally unique attribute values.                                                                              | 1.3.18.0.2.32.16 |
| Group-Specific Search Limits               | Supports extended search limits for a group of people.                                                                                       | 1.3.18.0.2.32.17 |
| IBMpolicies Replication Subtree            | Server supports the replication of the <code>cn=IBMpolicies</code> subtree.                                                                  | 1.3.18.0.2.32.18 |
| Max Age ChangeLog Entries                  | Specifies that the server is capable of retaining changelog entries based on age.                                                            | 1.3.18.0.2.32.19 |

Table 19. OIDs for supported and enabled capabilities (continued)

| Short name                                              | Description                                                                                                                                                                                                             | OID assigned           |
|---------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|
| Monitor Logging Counts                                  | The server provides monitor logging counts for messages added to server, command-line interface, and audit log files.                                                                                                   | 1.3.18.0.2.32.20       |
| Monitor Active Workers Information                      | The server provides monitor information for active workers (cn=workers,cn=monitor).                                                                                                                                     | 1.3.18.0.2.32.21       |
| Monitor Connection Type Counts                          | The server provides monitor connection type counts for SSL and TLS connections.                                                                                                                                         | 1.3.18.0.2.32.22       |
| Monitor Connections Information                         | The server provides monitor information for connections by IP address instead of connection ID (cn=connections, cn=monitor)                                                                                             | 1.3.18.0.2.32.23       |
| Monitor Operation Counts                                | The server provides new monitor operation counts for initiated and completed operation types.                                                                                                                           | 1.3.18.0.2.32.24       |
| Monitor Tracing Info                                    | The server provides monitor information for tracing options currently being used.                                                                                                                                       | 1.3.18.0.2.32.25       |
| Null Base Subtree Search                                | Server allows null based subtree search, which searches the entire DIT defined in the server.                                                                                                                           | 1.3.18.0.2.32.26       |
| Proxy Authorization                                     | Server supports Proxy Authorization for a group of users.                                                                                                                                                               | 1.3.18.0.2.32.27       |
| TLS Capabilities                                        | Specifies that the server is actually capable of doing TLS.                                                                                                                                                             | 1.3.18.0.2.32.28       |
| Non-Blocking Replication                                | The server is capable of ignoring some errors received from a consumer (replica) that would normally cause an update to be re-transmitted periodically until a successful result code was received.                     | 1.3.18.0.2.32.29       |
| Kerberos Capability                                     | Specifies that the server is capable of using Kerberos.                                                                                                                                                                 | 1.3.18.0.2.32.30       |
| ibm-allMembers and ibm-allGroups operational attributes | Indicates whether or not a backend supports searching on the ibm-allGroups and ibm-allMembers operational attributes.                                                                                                   | 1.3.18.0.2.32.31       |
| Language Tags                                           | Server supports language tags.                                                                                                                                                                                          | 1.3.6.1.4.1.4203.1.5.4 |
| FIPS mode for GSKit                                     | Enables the server to use the encryption algorithms from the ICC FIPS-certified library                                                                                                                                 | 1.3.18.0.2.32.32       |
| Modify DN (leaf move)                                   | Indicates if modify DN operation supports new superior for leaf entries. Note that this capability is implied by the pre-existing Modify DN (subtree move) capability. Applications should check for both capabilities. | 1.3.18.0.2.32.35       |
| Filtered Referrals                                      | The server supports limited filtered referrals.                                                                                                                                                                         | 1.3.18.0.2.32.36       |
| Simplify resizing of attributes                         | Allows customers to increase the maximum length of attributes through the schema modification facilities.                                                                                                               | 1.3.18.0.2.32.37       |
| Global Administration Group                             | Server supports the delegation of server administration to a group of administrators that are specified in the RDBM backend. Global Administrators do not have any authority to the configuration file or log files.    | 1.3.18.0.2.32.38       |
| AES Encryption Option                                   | Server supports auditing of compare operations.                                                                                                                                                                         | 1.3.18.0.2.32.39       |

Table 19. OIDs for supported and enabled capabilities (continued)

| Short name                                       | Description                                                                                                                               | OID assigned     |
|--------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| Auditing of Compare                              | Server supports auditing of compare operations.                                                                                           | 1.3.18.0.2.32.40 |
| Log Management                                   | Identifies that this server supports log management.                                                                                      | 1.3.18.0.2.32.41 |
| Multi-threaded Replication                       | Replication agreements can specify using multiple threads and connections to a consumer.                                                  | 1.3.18.0.2.32.42 |
| Supplier Replication Configuration               | Server configuration of suppliers for replication.                                                                                        | 1.3.18.0.2.32.43 |
| Using CN=IBMPOLICIES for Global Updates          | Server supports the replication of global updates using the replication topology in cn=IBMpolicies subtree.                               | 1.3.18.0.2.32.44 |
| Multihomed configuration support                 | Server supports configuration on multiple IP addresses (multihomed).                                                                      | 1.3.18.0.2.32.45 |
| Multiple Directory Server Instances Architecture | Server is designed to run with multiple directory server instances on the same machine.                                                   | 1.3.18.0.2.32.46 |
| Configuration Tool Auditing                      | Server supports the auditing of the the configuration tools.                                                                              | 1.3.18.0.2.32.47 |
| Audit Configuration Settings Consolidation       | Identifies that the audit configuration settings are now residing in the ibmslapd configuration file only.                                | 1.3.18.0.2.32.58 |
| Proxy Server                                     | Describes whether this server is capable of acting as a proxy server or regular RDBM server. Optional Information.                        | 1.3.18.0.2.32.49 |
| Replication conflict resolution max entry size   | Based on this number, a supplier may decide if an entry should be re-added to a target server in order to resolve a replication conflict. | 1.3.18.0.2.32.51 |
| LostAndFound log file                            | Supports LostAndFound file for archiving replaced entries as a result of replication conflict resolution.                                 | 1.3.18.0.2.32.52 |
| Password Policy Account Lockout                  | Identifies that this server supports password policy Account Locked feature.                                                              | 1.3.18.0.2.32.53 |
| Password Policy Admin                            | Identifies that this server supports Admin Password Policy.                                                                               | 1.3.18.0.2.32.54 |
| IDS 6.0 ibm-entrychecksumop                      | Identifies that the 6.0 version of the ibm-entrychecksumop calculation was used on the server.                                            | 1.3.18.0.2.32.56 |

## OIDs for ACI mechanisms

The following table shows the OIDs for ACI mechanisms.

Table 20. OIDs for ACI mechanisms

| Short name                     | Description                                                                          | OID assigned    |
|--------------------------------|--------------------------------------------------------------------------------------|-----------------|
| IBM SecureWay V3.2 ACL Model   | Indicates that the LDAP server supports the IBM SecureWay V3.2 ACL model             | 1.3.18.0.2.26.2 |
| IBM Filter Based ACL Mechanism | Indicates that the LDAP server supports IBM Directory Server v5.1 filter based ACLs. | 1.3.18.0.2.26.3 |

Table 20. OIDs for ACL mechanisms (continued)

| Short name                    | Description                                                                               | OID assigned    |
|-------------------------------|-------------------------------------------------------------------------------------------|-----------------|
| System Restricted ACL Support | Server supports specification and evaluation of ACLs on system and restricted attributes. | 1.3.18.0.2.26.4 |

## OIDs for extended operations

The following table shows OIDs for extended operations.

Table 21. OIDs for extended operations

| Short name                                               | Description                                                                                                                                                                                                                                                              | OID assigned     |
|----------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| Account status extended operation                        | This extended operation sends the server a DN of an entry which contains a userPassword attribute, and the server sends back the status of the user account being queried: <ul style="list-style-type: none"> <li>• open</li> <li>• locked</li> <li>• expired</li> </ul> | 1.3.18.0.2.12.58 |
| Attribute type extended operations                       | Retrieve attributes by supported capability: operational, language tag, attribute cache, unique or configuration.                                                                                                                                                        | 1.3.18.0.2.12.46 |
| Begin transaction extended operation                     | Begin a Transactional context.                                                                                                                                                                                                                                           | 1.3.18.0.2.12.5  |
| Cascading replication operation extended operation       | This operation performs the requested action on the server it is issued to and cascades the call to all consumers beneath it in the replication topology.                                                                                                                | 1.3.18.0.2.12.15 |
| Clear log extended operation                             | Request to Clear log file.                                                                                                                                                                                                                                               | 1.3.18.0.2.12.20 |
| Control replication extended operation                   | This operation is used to force immediate replication, suspend replication, or resume replication by a supplier. This operation is allowed only when the client has update authority to the replication agreement                                                        | 1.3.18.0.2.12.16 |
| Control queue extended operation                         | This operation marks items as "already replicated" for a specified agreement. This operation is allowed only when the client has update authority to the replication agreement.                                                                                          | 1.3.18.0.2.12.17 |
| DN normalization extended operation                      | Request to normalize a DN or a sequence of DNs.                                                                                                                                                                                                                          | 1.3.18.0.2.12.30 |
| Dynamic server trace extended operation                  | Activate or deactivate tracing in the IBM Tivoli Directory Server.                                                                                                                                                                                                       | 1.3.18.0.2.12.40 |
| Dynamic update requests extended operation               | Request to update server configuration for IBM Tivoli Directory Server.                                                                                                                                                                                                  | 1.3.18.0.2.12.28 |
| End transaction extended operation                       | End Transactional context (commit/rollback).                                                                                                                                                                                                                             | 1.3.18.0.2.12.6  |
| Event notification register request extended operation   | Request registration for events notification.                                                                                                                                                                                                                            | 1.3.18.0.2.12.1  |
| Event notification unregister request extended operation | Unregister for events that were registered for using an Event Registration Request.                                                                                                                                                                                      | 1.3.18.0.2.12.3  |
| Get lines extended operation                             | Request to get lines from a log file.                                                                                                                                                                                                                                    | 1.3.18.0.2.12.22 |

Table 21. OIDs for extended operations (continued)

| Short name                                                  | Description                                                                                                                                                                                                                                    | OID assigned           |
|-------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|
| Get number of lines extended operation                      | Request number of lines in a log file.                                                                                                                                                                                                         | 1.3.18.0.2.12.24       |
| Group evaluation extended operation                         | Requests all the groups that a given user belongs to.                                                                                                                                                                                          | 1.3.18.0.2.12.50       |
| Kill connection extended operation                          | Request to kill connections on the server. The request can be to kill all connections or kill connections by bound DN, IP, or a bound DN from a particular IP.                                                                                 | 1.3.18.0.2.12.35       |
| LDAP trace facility extended operation                      | Use this extended operation to control LDAP Trace Facility remotely using the Admin Daemon.                                                                                                                                                    | 1.3.18.0.2.12.41       |
| Quiesce or unquiesce replication context extended operation | This operation puts the subtree into a state where it does not accept client updates (or terminates this state), except for updates from clients authenticated as directory administrators where the Server Administration control is present. | 1.3.18.0.2.12.19       |
| Replication error log extended operation                    | Maintenance of a replication error table.                                                                                                                                                                                                      | 1.3.18.0.2.12.56       |
| Replication topology extended operation                     | Trigger a replication of replication topology-related entries under a given replication context.                                                                                                                                               | 1.3.18.0.2.12.54       |
| Start, stop server extended operations                      | Request to start, stop or restart an LDAP server.                                                                                                                                                                                              | 1.3.18.0.2.12.26       |
| Start TLS extended operation                                | Request to start Transport Layer Security.                                                                                                                                                                                                     | 1.3.6.1.4.1.1466.20037 |
| Unique attributes extended operation                        | Feature to enforce attribute uniqueness.                                                                                                                                                                                                       | 1.3.18.0.2.12.44       |
| Update configuration extended operation                     | Request to update server configuration for IBM Tivoli Directory Server.                                                                                                                                                                        | 1.3.18.0.2.12.28       |
| Update event notification extended operation                | Request that the event notification plug-in get the updated configuration from the server.                                                                                                                                                     | 1.3.18.0.2.12.31       |
| Update log access extended operation                        | Request that the log access plug-in get the updated configuration from the server.                                                                                                                                                             | 1.3.18.0.2.12.32       |
| User type extended operation                                | Request to get the User Type of the bound user.                                                                                                                                                                                                | 1.3.18.0.2.12.37       |

## OIDs for controls

The following table shows OIDs for controls.

Table 22. OIDs for controls

| Short name       | Description                                                                                                                                                                                                     | OID assigned     |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| AES bind control | This control enables the IBM Tivoli Directory Server to send updates to the consumer server with passwords already encrypted using AES.                                                                         | 1.3.18.0.2.10.28 |
| Audit control    | The control sends a sequence of uniqueid strings and a source ip string to the server. When the server receives the control, it audits the list of uniqueids and sourceip in the audit record of the operation. | 1.3.18.0.2.10.22 |

Table 22. OIDs for controls (continued)

| Short name                                 | Description                                                                                                                                                                                                                                                                                                  | OID assigned              |
|--------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|
| Do not replicate control                   | This control can be specified on an update operation (add, delete, modify, modDn, modRdn).                                                                                                                                                                                                                   | 1.3.18.0.2.10.23          |
| Group authorization control                | The control sends a list of groups that a user belongs to.                                                                                                                                                                                                                                                   | 1.3.18.0.2.10.21          |
| Manage DSAIT control                       | Causes entries with the "ref" attribute to be treated as normal entries, allowing clients to read and modify these entries.                                                                                                                                                                                  | 2.16.840.1.113730.3.4.2   |
| Modify groups only control                 | Attached to a delete or modify DN request to cause the server to do only the group referential integrity processing for the delete or rename request without doing the actual delete or rename of the entry itself. The entry named in the delete or modify DN request does not need to exist on the server. | 1.3.18.0.2.10.25          |
| No replication conflict resolution control | When present, a replica server accepts a replicated entry without trying to resolve any replication conflict for this entry.                                                                                                                                                                                 | 1.3.18.0.2.10.27          |
| Omit group referential integrity control   | Omits the group referential integrity processing on a delete or modrdn request.                                                                                                                                                                                                                              | 1.3.18.0.2.10.26          |
| Paged search results control               | Allows management of the amount of data returned from a search request.                                                                                                                                                                                                                                      | 1.2.840.113556.1.4.319    |
| Password policy request control            | Password policy request or response                                                                                                                                                                                                                                                                          | 1.3.6.1.4.1.42.2.27.8.5.1 |
| Proxy authorization control                | The Proxy Authorization Control enables a bound user to assert another user's identity. The server uses this asserted identity in the evaluation of ACLs for the operation.                                                                                                                                  | 2.16.840.1.113730.3.4.18  |
| Refresh entry control                      | This control is returned when a target server detects a conflict ( $T0 \neq T2$ & $T1 > T2$ ) during a replicated modify operation.                                                                                                                                                                          | 1.3.18.0.2.10.24          |
| Replication supplier bind control          | This control is added by the supplier, if the supplier is a gateway server.                                                                                                                                                                                                                                  | 1.3.18.0.2.10.18          |
| Replication update ID control              | This control was created for serviceability. If the supplier server is set to issue the control, each replicated update is accompanied by this control.                                                                                                                                                      | 1.3.18.0.2.10.29          |
| Server administration control              | Allows an update operation by the administrator under conditions when the operation would normally be refused (server is quiesced, a read-only replica, etc.)                                                                                                                                                | 1.3.18.0.2.10.15          |
| Sorted search results control              | Allows a client to receive search results sorted by a list of criteria, where each criterion represents a sort key.                                                                                                                                                                                          | 1.2.840.113556.1.4.473    |
| Subtree delete control                     | This control is attached to a Delete request to indicate that the specified entry and all descendent entries are to be deleted.                                                                                                                                                                              | 1.2.840.113556.1.4.805    |
| Transaction control                        | Marks the operation as part of a transactional context.                                                                                                                                                                                                                                                      | 1.3.18.0.2.10.5           |

---

## Appendix C. LDAP data interchange format (LDIF)

This documentation describes the LDAP Data Interchange Format (LDIF), as used by the `idsldapmodify`, `idsldapsearch` and `idsldapadd` utilities. The LDIF specified here is also supported by the server utilities provided with the IBM Directory.

LDIF is used to represent LDAP entries in text form. The basic form of an LDIF entry is:

```
dn: <distinguished name>
<attrtype> : <attrvalue>
<attrtype> : <attrvalue>
...
```

A line can be continued by starting the next line with a single space or tab character, for example:

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

Multiple attribute values are specified on separate lines, for example:

```
cn: John E Doe
cn: John Doe
```

If an `<attrvalue>` contains a non-US-ASCII character, or begins with a space or a colon `':'`, the `<attrtype>` is followed by a double colon and the value is encoded in base-64 notation. For example, the value " begins with a space" would be encoded like this:

```
cn:: IGJlZ21ucyB3aXRoIGEgc3BhY2U=
```

Multiple entries within the same LDIF file are separated by a blank line. Multiple blank lines are considered a logical end-of-file.

---

### LDIF example

Here is an example of an LDIF file containing three entries.

```
dn: cn=John E Doe, o=University of High
 er Learning, c=US
cn: John E Doe
cn: John Doe
objectclass: person
sn: Doe

dn: cn=Bjorn L Doe, o=University of High
 er Learning, c=US
cn: Bjorn L Doe
cn: Bjorn Doe
objectclass: person
sn: Doe

dn: cn=Jennifer K. Doe, o=University of High
 er Learning, c=US
cn: Jennifer K. Doe
cn: Jennifer Doe
objectclass: person
sn: Doe
```

```
jpegPhoto:: /9j/4AAQSkZJRgABAAAAQABAAD/2wBDABALD
A4MChAODQ4SERATGCgaGBYWGDEjJR0o0jM9PDkzODdASFxOQ
ERXRTc4UG1RV19iZ2hnPk1xeXBkeFxlZ2P/2wBDARESEhgVG
...
```

The jpegPhoto in Jennifer Doe's entry is encoded using base-64. The textual attribute values can also be specified in base-64 format. However, if this is the case, the base-64 encoding must be in the code page of the wire format for the protocol (that is, for LDAP V2, the IA5 character set and for LDAP V3, the UTF-8 encoding).

---

## Version 1 LDIF support

The client utilities (idsldapmodify and idsldapadd) have been enhanced to recognize the latest version of LDIF, which is identified by the presence of the "version: 1" tag at the head of the file. Unlike the original version of LDIF, the newer version of LDIF supports attribute values represented in UTF-8 (instead of the very limited US-ASCII).

However, manual creation of an LDIF file containing UTF-8 values may be difficult. In order to simplify this process, a charset extension to the LDIF format is supported. This extension allows an IANA character set name to be specified in the header of the LDIF file (along with the version number). A limited set of the IANA character sets are supported. See "IANA character sets supported by platform" on page 483 for the specific charset values that are supported for each operating system platform.

The version 1 LDIF format also supports file URLs. This provides a more flexible way to define a file specification. File URLs take the following form:

```
attribute:< file:///path (where path syntax depends on platform)
```

For example, the following are valid file Web addresses:

```
jpegphoto:< file:///d:\temp\photos\myphoto.jpg (DOS/Windows style paths)
jpegphoto:< file:///etc/temp/photos/myphoto.jpg (UNIX or Linux style paths)
```

**Note:** The IBM Directory utilities support both the new file URL specification as well as the older style (e.g. "jpegphoto: /etc/temp/myphoto"), regardless of the version specification. In other words, the new file URL format can be used without adding the version tag to your LDIF files.

---

## Version 1 LDIF examples

You can use the optional charset tag so that the utilities will automatically convert from the specified character set to UTF-8 as in the following example:

```
version: 1
charset: ISO-8859-1

dn: cn=Juan Griego, o=University of New Mexico, c=US
cn: Juan Griego
sn: Griego
description:: V2hhdCBhIGNhcmVmdWwgcmlVhZGVyIH1vd
title: Associate Dean
title: [title in Spanish]
jpegPhoto:< file:///usr/local/photos/jgriego.jpg
```

In this instance, all values following an attribute name and a single colon are translated from the ISO-8859-1 character set to UTF-8. Values following an attribute name and a double colon (such as description:: V2hhdCBhIGNhcm... ) must be



base-64 encoded, and are expected to be either binary or UTF-8 character strings. Values read from a file, such as the jpegPhoto attribute specified by the Web address in the previous example, are also expected to be either binary or UTF-8. No translation from the specified "charset" to UTF-8 is done on those values.

In this example of an LDIF file without the charset tag, content is expected to be in UTF-8, or base-64 encoded UTF-8, or base-64 encoded binary data:

```
IBM Directorysample LDIF file
#
The suffix "o=IBM, c=US" should be defined before attempting to load
this data.

version: 1

dn: o=IBM, c=US
objectclass: top
objectclass: organization
o: IBM

dn: ou=Austin, o=IBM, c=US
ou: Austin
objectclass: organizationalUnit
seealso: cn=Linda Carlesberg, ou=Austin, o=IBM, c=US
```

This same file could be used without the version: 1 header information, as in previous releases of the IBM Directory:

```
IBM Directorysample LDIF file
#
The suffix "o=IBM, c=US" should be defined before attempting to load
this data.

dn: o=IBM, c=US
objectclass: top
objectclass: organization
o: IBM

dn: ou=Austin, o=IBM, c=US
ou: Austin
objectclass: organizationalUnit
seealso: cn=Linda Carlesberg, ou=Austin, o=IBM, c=US
```

**Note:** The textual attribute values can be specified in base-64 format.

---

## 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 23. IANA-defined character sets

| Character<br>Set Name | Locale                      |                      |     |     |         | DB2 Code Page |      |
|-----------------------|-----------------------------|----------------------|-----|-----|---------|---------------|------|
|                       | HP-UX                       | Linux,<br>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  |
| 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<br>non-<br>GB18030) |                      |     |     |         |               |      |

\* Supported at Solaris 7.

**Notes:**

1. The new Chinese character set standard (GB18030) is supported with appropriate patches available from [www.sun.com](http://www.sun.com) and [www.microsoft.com](http://www.microsoft.com)
2. On the Windows 2000 operating system, you must set the environment variable zhCNGB18030=TRUE.



## Appendix D. 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 E. IPv6 support

Internet Protocol Version 6 (IPv6) is the protocol designed by the IETF to replace the current version Internet Protocol, IP Version 4 (IPv4). IPv6 fixes a number of problems in IPv4, such as the limited number of available IPv4 addresses. IPv6 uses a wider address (128-bit vs 32-bit) than IPv4, and this has an impact on the TCP application level. It also has improvements in areas such as routing and network autoconfiguration. IPv6 is expected to gradually replace IPv4.

All supported servers and clients for IBM Tivoli Directory Server version 6.0 are enabled to support IPv6 as well as IPv4 nodes. The following are examples of the format of LDAP URLs for IPv4 and IPv6 .

**Note:** If *:portnumber* is not specified in the URL, the default ports (389 for non-SSL and 636 for SSL) are used.

- To use a literal IPv4 address in a URL, the format is *x.x.x.x:port*. An example of an LDAP server name in a URL for non-SSL communication listening on port 80 is:

– `ldap://9.53.90.21:80`

An example of an LDAP server name in a URL for SSL communication listening on the default port of 636 is:

– `ldaps://9.53.90.21`

- To comply with RFC 2732, literal IPv6 address in URLs must be enclosed in [ and ] characters. Examples of LDAP server names in URLs for non-SSL communication listening on the respective ports of 80 and the default of 389 are:

– `ldap://[107:0:0:0:200:7051]:80`

– `ldap://[::ffff:9.53.96.21]`

Examples of LDAP server names in URLs for SSL communication listening on the respective ports of 80 and the default of 636 are:

– `ldaps://[107:0:0:0:200:7051]:80`

– `ldaps://[::ffff:9.53.96.21]`

### Notes:

1. If you are using the IPv6 URL format in a mixed environment with directory servers that are not IPv6 enabled, for example, IBM Directory Server Version 5.1 or earlier versions, the IPv6 URL format is not recognized by the non-IPv6 enabled clients and servers. For example:
  - Referrals do not work if a non-IPv6 enabled client receives a URL address in the IPv6 format.
  - Replication does not work if a non-IPv6 enabled consumer server receives its supplier URL information in the IPv6 format.
2. Linux systems require an interface ID for resolving the link-local IP address. The `getaddrinfo` or other interface conversion routines work, but then the resolved IP address does not work for the `connect()` function. Use the following format to specify an IP address with interfaces ID:

`ldap://[xxxx:xxxx:xxxx:xxxx:xxxx%InterfaceID]`

The link-local IPv6 address with scope:local does not work on Linux systems. IBM Tivoli Directory Server 6.0 supports scope:global only in IPv6 addresses on Linux systems.



---

## Appendix F. Simple Networking Management Protocol

To monitor the IBM Tivoli Directory server, you need to modify the properties and configuration files for the Simple Networking Management Protocol (SNMP) Agent.

**Note:** You must have IBM Tivoli Directory Integrator 6.0 fix pack 1 installed to use the SNMP agent. If you acquire IBM Tivoli Directory Server through Passport Advantage®, it includes a copy of IBM Tivoli Directory Integrator 6.0 fix pack 1 for limited use with IBM Tivoli Directory Server 6.0. (Installation instructions are included with the package.)

You also need to add a user to the directory and place ACLs on the suffixes of the directory, denying the user any permission to access the Data Information Tree (DIT) data. This user is created for performing monitor searches only and must exist across all monitored instances.

The properties file, `idssnmp.properties`, contains the user ID of the directory user, and the user password. The `idssnmp.properties` file is encrypted by default once the `idssnmp` agent is started. This file is located in the `<install_directory>\idstools\snmp` directory. The `idssnmp.properties` file contains the following:

```
snmpuserid:<user_ID>
snmpuserpwd:<user_pwd>
```

where `<user_ID>` is the user ID used to access the DIT, and `<user_pwd>` is the user's password.

The configuration file, `idssnmp.conf`, is in the standard SNMP format, that is, space separated with certain keywords. This configuration file contains the port number on which the SNMP agent runs, at least one IP address or host name, the IP address of the network management system (NMS) to where the connector sends its traps, and the communities that this SNMP Agent responds to. This file is located in the `<install_directory>\idstools\snmp` directory.

1. Edit the port number in the configuration file for the IBM Tivoli Directory Server SNMP agent. The SNMP Agent monitors the IBM Tivoli Directory Server. If you want to monitor something other than the directory server, the SNMP agent for the IBM Tivoli Directory Server must be run on a nonstandard port. The nonstandard port is necessary to avoid a port conflict with the agent for the other application.

```
Port 161
```

The example shows that the SNMP agent runs on port 161. If more than one port is specified, only the first line of type `Port` is read, others are ignored

2. Add lines with `"Server"` and `"SecureServer"` as tags and the IP address and the port number on which the monitored LDAP servers are running. The keyword `Server` specifies a non-encrypted connection. It is followed by the IP Address of the LDAP instance and then the port number of the LDAP instance. The IP Address should be IPV4. The keyword `SecureServer` indicates that the communication between the LDAP instance and the SNMP Agent is SSL encrypted.

```
Server 13.14.15.16 389
SecureServer 9.10.11.12 636
```

In this example, two servers are configured, one on a regular port, and one on a secure port. The first server is using unencrypted authentication. The second server is using SSL authentication. If a server is configured for both (simple and SSL) then it is the prerogative of the directory administrator to decide which port (secured or unsecured) of the LDAP server is accessed.

3. To properly receive any traps, you must edit the line in the SNMP configuration file that has the keyword Trap by adding the IP address of the NMS receiving the traps (by default the value is 127.0.0.1), its port number and the community string it expects to receive from the agent. You can repeat the line to specify multiple machines that are receiving the traps. For example:

```
Trap 5.4.3.2 162 public
```

This example shows that any traps that are generated are sent to a machine with the IP address 5.4.3.2 on port 162 using the community string "public".

4. Specify a polling interval in seconds. After the specified number of seconds the agent polls the servers to discover their status.

```
Poll 600
```

In this example the agent checks the servers every 600 seconds, that is, every 10 minutes.

5. If you want to restrict access to the agent, you can specify an optional community string. If you specify community, you must provide the string. For example:

```
Community dirServer
```

Any machine supplying the community string, dirServer, has access to the data. If the community string is not specified, authorization is not restricted. To further restrict access, you can provide other tokens such as the IP address in the community string line that the machine originating the request must have:

```
Community dirServer 1.2.3.4
```

If no IP Address is specified, then any machine supplying the community string has access to the data. If additional access restrictions are needed, you can also specify the supported access right, readOnly, to the elements of the community and lastly the view of the subtree. Please note that the data is implicitly read only and that readOnly is used to maintain the SNMP configuration file standards. If you specify community, the string is required. The IP address, access right and view are optional, however these restrictions are sequential in nature. You can optionally specify IP address or IP address and access right, but you could not optionally specify the access right and view without IP address.

This example is the most restrictive and illustrates the correct sequence of the tokens.

```
Community dirServer 1.2.3.4 readOnly 1.5.4.3.2.1
```

In this example, the requesting NMSs must supply "dirServer" as a community string. The requests must originate from a machine with IP address 1.2.3.4 and all elements in this community are read only and the view is 1.5.4.3.2.1.

**Note:** With restricted authorization, if more than one machine is running an NMS authorized to perform get operation on the Directory SNMP Agent, the community line will need to be duplicated.

6. If you need to divide the SNMP OID tree, you can specify a view of the subtree.

View 1.5.4.3.2.1

This example indicates that the agent deals with all the subtrees under the OID 1.5.4.3.2.1.

**Notes:**

1. When Server and SecureServer are specified in the idssnmp.conf file, their order in the file is inconsequential to the way the SNMP agent returns the next value for a get\_next operation.

For example, if the idssnmp.conf file contains the following:

```
Port 161
Community public 9.182.182.134
Trap 9.182.190.159 162 public
SecureServer 9.53.190.143 1636
Server 9.53.190.176 389
SecureServer 9.53.190.143 636
```

Then the following occurs:

- The get-next for 1.3.6.1.4.1.2.6.199.2.1.63.1.4.9.53.190.143.636 returns SNMPv2-SMI::enterprises.2.6.199.2.1.63.1.4.9.53.190.143.1636
- The get-next for 1.3.6.1.4.1.2.6.199.2.1.63.1.4.9.182.190.143.1636 returns SNMPv2-SMI::enterprises.2.6.199.2.1.63.1.4.9.53.190.176.389
- The get-next for 1.3.6.1.4.1.2.6.199.2.1.63.1.4.9.53.190.176.389 returns an error message that indicates there are no more results in the Management information base (MIB) tree.

The order of the servers is determined by the value of the server's IP and port. The servers with lower values are at the beginning of the MIB tree. You can see from the preceding example that the server IPs are identical until the 143 and 176 are reached. Since 143 is lower than 176, the servers with 9.53.190.143 are placed ahead of the 9.53.190.176 server. In the case where the 9.53.190.143 servers are matching until the port is reached, the port numbers are compared. Because 636 is lower than 1636, the 9.53.190.143 636 is placed ahead of the 9.53.190.143 1636 in the MIB tree.

2. Load the following MIBS to your NMS:

```
<install_directory>/idstools/snmp/IBM-DIRECTORYSERVER-MIB
<install_directory>/idstools/snmp/INET-ADDRESS-MIB
```

The SNMP agent can be started by running the idssnmp script located in the <install\_directory>\sbin directory.

See the IBM Tivoli Directory Integrator documentation for information on how to install IBM Tivoli Directory Integrator and how to setup SSL (*IBM Tivoli Directory Integrator Users Guide*).

---

## Logging

By default, the idssnmp application logs its data to the file /var/idldap/V6.0/idssnmp.log on the UNIX platform and <ldap install dir>\var\idssnmp.log on the Windows platform.

In addition to the tool's main log file, `idssnmp.log`, there are two additional log files that ITDI produces:

- `ibmdi.log`
- `idssnmpinit.log`

These files are produced because the ITDI application writes logs to a static location. After the `idssnmp` tool is initialized, most of the log statements are written to `idssnmp.log`. The `ibmdi.log` file and `idssnmpinit.log` file are written to the following directories:

- `/var/idsldap/V6.0` (UNIX)
- `<install_directory>\var` (Windows)

If these directories are not created, then the logs are placed in the current working directory. The `ibmdi.log` and `idssnmpinit.log` are overwritten each time the `idssnmp` tool is run so the filesize can remain small.

The following command line option:

`-D DEBUG`

can be specified to debug `idssnmp`. The log can then have more detailed information of the agent's execution.

---

## 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 G. Password policy operational attributes

The following operational attributes are provided by the password policy feature:

| Attribute name       | Syntax          | Description                                                                                                                                            |
|----------------------|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| pwdChangedTime       | GeneralizedTime | Contains the time the password was last changed.                                                                                                       |
| pwdAccountLockedTime | GeneralizedTime | Contains the time at which the account was locked. If the account is not locked, this attribute is not present.                                        |
| pwdExpirationWarned  | GeneralizedTime | Contains the time at which the password expiration warning was first sent to the client.                                                               |
| pwdFailureTime       | GeneralizedTime | A multi-valued attribute containing the times of previous consecutive login failures. If the last login was successful, this attribute is not present. |
| pwdGraceUseTime      | GeneralizedTime | A multi-valued attribute containing the times of the previous grace logins.                                                                            |
| pwdReset             | Boolean         | Contains the value TRUE if the password has been reset and must be changed by the user. The value is FALSE or not present otherwise.                   |
| ibm-pwdAccountLocked | Boolean         | Indicates that the account has been administratively locked.                                                                                           |

---

### Password policy queries

The password policy operational attributes can be used to view the status of a directory entry or to query for entries matching specified criteria. Operational attributes are returned on a search request only when specifically requested by the client. To use these attributes in search operations, you must have permission to critical attributes, or permission to the specific attributes used.

To view all password policy attributes for a given entry:

```
idsldapsearch -b "uid=user1,cn=users,o=ibm" -s base "(objectclass=*)" pwdChangedTime
pwdAccountLockedTime pwdExpirationWarned pwdFailureTime pwdGraceUseTime
pwdReset
```

To query for entries for which the password is about to expire, use the `pwdChangedTime`. For example, to find passwords which expire on August 26, 2004, with a password expiration policy of 186 days, query for entries for which the password was changed at least 186 days ago (February 22, 2004):

```
idsldapsearch -b "cn=users,o=ibm" -s sub "(!(pwdChangedTime>20040222000000Z))" <dn>
```

where the filter is equivalent to `pwdChangedTime` is less than or equal to midnight, February 22, 2004.

To query for locked accounts, use the `pwdAccountLockedTime`:

```
idsldapsearch -b "cn=users,o=ibm" -s sub "(pwdAccountLockedTime=*)" <dn>
```

To query for accounts for which the password must be changed because the password was reset, use the `pwdReset` attribute:

```
idsldapsearch -b "cn=users,o=ibm" -s sub "(pwdReset=TRUE)" <dn>
```

---

## Overriding password policy and unlocking accounts

A directory administrator can override normal password policy behavior for specific entries by modifying the password policy operational attributes and using the server administration control (`-k` option of the LDAP command line utilities).

You can prevent the password for a particular account from expiring by setting the `pwdChangedTime` attribute to a date far in the future when setting the `userPassword` attribute. The following example sets the time to midnight, January 1, 2200.

```
idsldapmodify -D cn=root -w ? -k
dn: uid=wasadmin,cn=users,o=ibm
changetype: modify
replace: pwdChangedTime
pwdChangedTime: 22000101000000Z
```

You can unlock an account which has been locked due to excessive login failures by removing the `pwdAccountLockedTime` and `pwdFailureTime` attributes:

```
idsldapmodify -D cn=root -w ? -k
dn: uid=user1,cn=users,o=ibm
changetype: modify
delete: pwdAccountLockedTime
-
delete: pwdFailureTime
```

You can unlock an expired account by changing the `pwdChangedTime` and clearing the `pwdExpirationWarned` and `pwdGraceUseTime` attributes:

```
idsldapmodify -D cn=root -w ? -k
dn: uid=user1,cn=users,o=ibm
changetype: modify
replace: pwdChangedTime
pwdChangedTime: yyyymmddhhss.Z
-
delete: pwdExpirationWarned
-
delete: pwdGraceUseTime
```

You can clear and then reset the "password must be changed" status by deleting and adding the `pwdReset` attribute:

```
idsldapmodify -D cn=root -w ? -k
dn: uid=user1,cn=users,o=ibm
changetype: modify
delete: pwdReset
```

```
idsldapmodify -D cn=root -w ? -k
dn: uid=user2,cn=users,o=ibm
changetype: modify
replace: pwdReset
pwdReset: TRUE
```

An account can be administratively locked by setting the `ibm-pwdAccountLocked` operational attribute to `TRUE`. The account can be unlocked by setting the attribute to `FALSE`. Unlocking an account in this way does not affect the state of the account with respect to being locked due to excessive password failures or an expired password.

The user setting this attribute must have permission to write the `ibm-pwdAccountLocked` attribute, which is defined as being in the `CRITICAL` access class.

```
idsldapmodify -D uid=useradmin,cn=users,o=ibm -w ?
dn: uid=user1,cn=users,o=ibm
changetype: modify
replace: ibm-pwdAccountLocked
ibm-pwdAccountLocked: TRUE
```

To unlock the account:

```
idsldapmodify -D uid=useradmin,cn=users,o=ibm -w ?
dn: uid=user1,cn=users,o=ibm
changetype: modify
replace: ibm-pwdAccountLocked
ibm-pwdAccountLocked: FALSE
```

If the account is locked because the attribute `ibm-pwdAccountLocked` is set to `TRUE` and if the administrator clears this attribute (sets it to `FALSE`) and uses the administrative control (`-k` option), then the account is completely unlocked. The `pwdAccountLockedTime` and `pwdFailureTime` attributes are also cleared and reset.

---

## Replicating password policy operational attributes

The user-related elements of the password policy are stored in the entries as operational attributes. These attributes are subject to modifications even on a read-only replica, so replicating these attributes must be carefully considered.

### **pwdChangedTime**

The `pwdChangedTime` attribute must be replicated on all replicas, to enable expiration of the password.

### **pwdReset**

The `pwdReset` attribute must be replicated on all replicas, to deny access to operations other than bind and modify password.

### **pwdHistory**

The `pwdHistory` attribute must be replicated to writable replicas. This attribute does not need to be replicated to a read-only replica, as the password is never directly modified on this server.

### **pwdAccountLockedTime, pwdExpirationWarned, pwdFailureTime, pwdGraceUseTime**

The `pwdAccountLockedTime`, `pwdExpirationWarned`, `pwdFailureTime` and `pwdGraceUseTime` attributes must be replicated to writable replicas, making the password policy global for all servers. When the user entry is replicated to a read-only replica, these attributes must not be replicated. This means that the number of failures, the number of grace logins and the locking take place on each replicated server. For example, the effective number of failed attempts on a user password is:

$N \times M$

where N is the number of servers and M is the value of `pwdMaxFailure` attribute. Replicating these attributes to a read-only replica can reduce the number of tries globally but can also introduce some inconsistencies in the way the password policy is applied.

There are times when the values of `pwdAccountLockedTime`, `pwdExpirationWarned`, `pwdFailureTime` and `pwdGraceUseTime` are replicated. If the user's password is reset, thereby clearing some of these attributes, this action is replicated to the read-only replicas. Also, if an administrator on the master server uses the administrative control to overwrite the values of these attributes on the master server, this forced write of the operational attributes is also replicated to read-write and read-only replicas.

#### **ibm-pwdAccountLocked**

When the `ibm-pwdAccountLocked` attribute is set or cleared on the master server, this attribute is also replicated to the replicas. When this attribute is cleared while using the administrative control on the operation, the `pwdAccountLockedTime` attribute is also cleared so that the account is totally unlocked when this attribute is set to `FALSE`.

---

## **Forcing an add or update for an entry**

When an administrative user updates or adds an entry, specifying a password policy operational attribute as one of the attributes to be changed or in the case of a new entry, the administrative user specifies a value for one or more of the operational attributes, then the administrative user is performing a forced add/update for the entry.

A forced add/update of an entry means that the normal password policy processing is not performed for that entry. Only those password policy operational attributes specified on the operation are changed as indicated.

Normally the forced add/update is indicated by using the administrative control on the operation while specifying a password policy attribute.

When updating the `ibm-pwdAccountLocked` attribute, the administrative control does not need to be sent.

When the administrator is performing a forced add/update to an entry, the administrator has the intention to set all of the password policy attributes as the entry requires.

Do not force an add unless all of the normal password policy operational attributes have been given an appropriate value, such as `pwdReset` and `pwdChangedTime`. If `pwdChangedTime` is not given a value on a forced add, then this attribute is not set until the user first attempts to bind to the server, or until another forced update creates a time for this attribute.

If any of the password policy attributes need to be specifically set on an add operation, the new entry should be created first and a separate modify operation should be used to set any other password policy attribute.

If the `userpassword` attribute is being modified on the modify operation, then any password policy attributes that are to be force updated must be updated separate



from the userpassword modification operation. This ensures that all of the proper password policy changes that occur on an add or modify operation are performed.



---

## Appendix H. IBM Tivoli Directory Server 6.0 required attribute definitions

```
attributetypes=(1.3.18.0.2.4.285
NAME 'aclEntry'
DESC 'Holds the access controls for entries in an IBM eNetwork LDAP
directory'
EQUALITY 2.5.13.2
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.285
DBNAME('aclEntry' 'aclEntry')
ACCESS-CLASS restricted
LENGTH 32700)
```

```
attributetypes=(1.3.18.0.2.4.286
NAME 'aclPropagate'
DESC 'Indicates whether the ACL applies on entry or subtree.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.286
DBNAME('aclPropagate' 'aclPropagate')
ACCESS-CLASS restricted
LENGTH 5)
```

```
attributetypes=(1.3.18.0.2.4.287
NAME 'aclSource'
DESC 'Indicates whether the ACL applies on entry or subtree.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.287
DBNAME('aclSource' 'aclSource')
ACCESS-CLASS system
LENGTH 1000)
```

```
attributetypes=(2.5.4.1
NAME ('aliasedObjectName' 'aliasedentryname')
DESC 'Represents the pointed to entry that is specified within an
alias entry.'
EQUALITY 2.5.13.1
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(2.5.4.1
DBNAME('aliasedObject' 'aliasedObject')
ACCESS-CLASS normal
LENGTH 1000
EQUALITY)
```

```
attributetypes=(1.3.6.1.4.1.1466.101.120.6
NAME 'altServer'
DESC 'The values of this attribute are URLs of other servers which
may be contacted when this server becomes unavailable.'
EQUALITY 1.3.6.1.4.1.1466.109.114.1
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
USAGE dSAOperation)
IBMAttributetypes=(1.3.6.1.4.1.1466.101.120.6
DBNAME('altServer' 'altServer')
ACCESS-CLASS normal
LENGTH 2048)
```

```
attributetypes=(2.5.21.5
NAME 'attributeTypes'
DESC 'This attribute is typically located in the subschema entry
```

```

and is used to store all attributes known to the server and
objectClasses.'
EQUALITY 2.5.13.30
SYNTAX 1.3.6.1.4.1.1466.115.121.1.3
USAGE directoryOperation)
IBMAttributetypes=(2.5.21.5
DBNAME('attributeTypes' 'attributeTypes')
ACCESS-CLASS system
LENGTH 30
EQUALITY)

attributetypes=(2.5.4.15
NAME 'businessCategory'
DESC 'This attribute describes the kind of business performed by an
organization.'
EQUALITY 2.5.13.2
SUBSTR 2.5.13.4
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
USAGE userApplications)
IBMAttributetypes=(2.5.4.15
DBNAME('businessCategory' 'businessCategory')
ACCESS-CLASS normal
LENGTH 128
EQUALITY
SUBSTR)

attributetypes=(2.16.840.1.113730.3.1.5
NAME 'changeNumber'
DESC 'Contains the change number of the entry as assigned by the
supplier server.'
EQUALITY 2.5.13.14
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
NO-USER-MODIFICATION
USAGE userApplications)
IBMAttributetypes=(2.16.840.1.113730.3.1.5
DBNAME('changeNumber' 'changeNumber')
ACCESS-CLASS normal
LENGTH 11
EQUALITY APPROX)

attributetypes=(2.16.840.1.113730.3.1.8
NAME 'changes'
DESC 'Defines changes made to a directory server. These changes are
in LDIF format.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.5
SINGLE-VALUE
NO-USER-MODIFICATION
USAGE userApplications)
IBMAttributetypes=(2.16.840.1.113730.3.1.8
DBNAME('changes' 'changes')
ACCESS-CLASS sensitive)

attributetypes=(2.16.840.1.113730.3.1.77
NAME 'changeTime'
DESC 'Time last changed.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.24
SINGLE-VALUE
NO-USER-MODIFICATION
USAGE userApplications)
IBMAttributetypes=(2.16.840.1.113730.3.1.77
DBNAME('changeTime' 'changeTime')
ACCESS-CLASS normal
LENGTH 30)

attributetypes=(2.16.840.1.113730.3.1.7
NAME 'changeType'

```

```

DESC 'Describes the type of change performed on an entry. Accepted
values include: add, delete, modify, modrdn.'
EQUALITY 2.5.13.2
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
NO-USER-MODIFICATION
USAGE userApplications)
IBMAttributetypes=(2.16.840.1.113730.3.1.7
DBNAME('changeType' 'changeType')
ACCESS-CLASS normal
LENGTH 250
EQUALITY)

attributetypes=(2.5.4.3
NAME ('cn' 'commonName')
DESC 'This is the X.500 commonName attribute, which contains a name of an object.
If the object corresponds to a person, it is typically the persons
full name.'
SUP 2.5.4.41
EQUALITY 2.5.13.2
ORDERING 2.5.13.3
SUBSTR 2.5.13.4
USAGE userApplications)
IBMAttributetypes=(2.5.4.3
DBNAME('cn' 'cn')
ACCESS-CLASS normal
LENGTH 256
EQUALITY
ORDERING
SUBSTR
APPROX)

attributetypes=(2.5.18.1
NAME 'createTimestamp'
DESC 'Contains the time that the directory entry was created.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.24
SINGLE-VALUE
NO-USER-MODIFICATION
USAGE directoryOperation)
IBMAttributetypes=(2.5.18.1
DBNAME('ldap_entry' 'create_Timestamp')
ACCESS-CLASS system
LENGTH 26)

attributetypes=(2.5.18.3
NAME 'creatorsName'
DESC 'Contains the creator of a directory entry.'
EQUALITY 2.5.13.1
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
SINGLE-VALUE
NO-USER-MODIFICATION
USAGE directoryOperation)
IBMAttributetypes=(2.5.18.3
DBNAME('ldap_entry' 'creator')
ACCESS-CLASS system
LENGTH 1000
EQUALITY)

attributetypes=(2.16.840.1.113730.3.1.10
NAME 'deleteOldRdn'
DESC 'a flag which indicates if the old RDN should be retained as
an attribute of the entry'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
SINGLE-VALUE
NO-USER-MODIFICATION
USAGE userApplications)
IBMAttributetypes=(2.16.840.1.113730.3.1.10

```

```
DBNAME('deleteOldRdn' 'deleteOldRdn')
ACCESS-CLASS normal
LENGTH 5)
```

```
attributetypes=(2.5.4.13
NAME 'description'
DESC 'Attribute common
to CIM and LDAP schema to provide lengthy description of a
directory object entry.'
EQUALITY 2.5.13.2
SUBSTR 2.5.13.4
SYNTAX
1.3.6.1.4.1.1466.115.121.1.15
USAGE userApplications)
IBMAttributetypes=(2.5.4.13
DBNAME('description' 'description')
ACCESS-CLASS normal
LENGTH 1024
EQUALITY
SUBSTR)
```

```
attributetypes=(2.5.21.2
NAME 'ditContentRules'
DESC 'Refer to RFC 2252.'
EQUALITY 2.5.13.30
SYNTAX 1.3.6.1.4.1.1466.115.121.1.16
USAGE directoryOperation)
IBMAttributetypes=(2.5.21.2
DBNAME('ditContentRules' 'ditContentRules')
ACCESS-CLASS system
LENGTH 256
EQUALITY)
```

```
attributetypes=(2.5.21.1
NAME 'ditStructureRules'
DESC 'Refer to RFC 2252.'
EQUALITY 2.5.13.29
SYNTAX 1.3.6.1.4.1.1466.115.121.1.17
USAGE directoryOperation)
IBMAttributetypes=(2.5.21.1
DBNAME('ditStructureRules' 'ditStructureRules')
ACCESS-CLASS system
LENGTH 256
EQUALITY)
```

```
attributetypes=(2.5.4.49
NAME ('dn' 'distinguishedName')
DESC 'This attribute type is not used as the name of the object itself,
but it is instead a base type from which attributes with DN syntax
inherit. It is unlikely that values of this type itself will occur
in an entry.'
EQUALITY 2.5.13.1
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
USAGE userApplications)
IBMAttributetypes=(2.5.4.49
DBNAME('dn' 'dn')
ACCESS-CLASS normal
LENGTH 1000
EQUALITY)
```

```
attributetypes=(1.3.18.0.2.4.288
NAME 'entryOwner'
DESC 'Indicates the distinguished name noted as the owner of the
entry'
EQUALITY 2.5.13.2
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
USAGE directoryOperation)
```

```
IBMAttributetypes=(1.3.18.0.2.4.288
DBNAME('entryOwner' 'entryOwner')
ACCESS-CLASS restricted
LENGTH 1000)
```

```
attributetypes=(2.5.18.9
NAME 'hasSubordinates'
DESC 'Indicates whether any subordinate entries exist below the
entry holding this attribute.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
SINGLE-VALUE
NO-USER-MODIFICATION
USAGE directoryOperation)
IBMAttributetypes=(2.5.18.9
DBNAME('hasSubordinates' 'hasSubordinates')
ACCESS-CLASS system
LENGTH 5)
```

```
attributetypes=(1.3.18.0.2.4.2244
NAME 'ibm-allGroups'
DESC 'All groups to which an entry belongs. An entry may be a member
directly via member, uniqueMember or memberURL attributes, or
indirectly via ibm-memberGroup attributes. Read-only operational
attribute (not allowed in filter).'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
NO-USER-MODIFICATION
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2244
DBNAME('allGroups' 'allGroups')
ACCESS-CLASS normal
LENGTH 1000)
```

```
attributetypes=(1.3.18.0.2.4.2243
NAME 'ibm-allMembers'
DESC 'All members of a group. An entry may be a member directly via
member, uniqueMember or memberURL attributes, or indirectly via
ibm-memberGroup attributes. Read-only operational attribute (not
allowed in filter).'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
NO-USER-MODIFICATION
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2243
DBNAME('ibmallMembers' 'ibmallMembers')
ACCESS-CLASS normal
LENGTH 1000)
```

```
attributetypes=(1.3.18.0.2.4.1077
NAME 'ibm-audit'
DESC 'TRUE or FALSE. Enable or disable the audit service. Default
is FALSE.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.1077
DBNAME('audit' 'audit')
ACCESS-CLASS critical
LENGTH 16)
```

```
attributetypes=(1.3.18.0.2.4.1073
NAME 'ibm-auditAdd'
DESC 'TRUE or FALSE. Indicate whether to log the Add operation.
Default is FALSE.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.1073
DBNAME('auditAdd' 'auditAdd')
```

ACCESS-CLASS critical  
LENGTH 16 )

attributetypes=( 1.3.18.0.2.4.1070  
NAME 'ibm-auditBind'  
DESC 'TRUE or FALSE. Indicate whether to log the Bind operation.  
Default is FALSE.'  
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7  
SINGLE-VALUE  
USAGE directoryOperation )  
IBMAttributetypes=( 1.3.18.0.2.4.1070  
DBNAME( 'auditBind' 'auditBind' )  
ACCESS-CLASS critical  
LENGTH 16 )

attributetypes=( 1.3.18.0.2.4.1071  
NAME 'ibm-auditDelete'  
DESC 'TRUE or FALSE. Indicate whether to log the Delete operation.  
Default is FALSE.'  
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7  
SINGLE-VALUE  
USAGE directoryOperation )  
IBMAttributetypes=( 1.3.18.0.2.4.1071  
DBNAME( 'auditDelete' 'auditDelete' )  
ACCESS-CLASS critical  
LENGTH 16 )

attributetypes=( 1.3.18.0.2.4.1069  
NAME 'ibm-auditExtOpEvent'  
DESC 'TRUE or FALSE. Indicate whether to log LDAP v3 Event  
Notification extended operations. Default is FALSE.'  
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7  
SINGLE-VALUE  
USAGE directoryOperation )  
IBMAttributetypes=( 1.3.18.0.2.4.1069  
DBNAME( 'auditExtOpEvent' 'auditExtOpEvent' )  
ACCESS-CLASS critical  
LENGTH 16 )

attributetypes=( 1.3.18.0.2.4.1078  
NAME 'ibm-auditFailedOpOnly'  
DESC 'TRUE or FALSE. Indicate whether to only log failed operations.  
Default is FALSE.'  
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7  
SINGLE-VALUE  
USAGE directoryOperation )  
IBMAttributetypes=( 1.3.18.0.2.4.1078  
DBNAME( 'auditFailedOpOnly' 'auditFailedOpOnly' )  
ACCESS-CLASS  
critical LENGTH 16 )

attributetypes=( 1.3.18.0.2.4.1079  
NAME 'ibm-auditLog'  
DESC 'Specifies the pathname for the audit log.'  
EQUALITY 2.5.13.5 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15  
SINGLE-VALUE  
USAGE directoryOperation )  
IBMAttributetypes=( 1.3.18.0.2.4.1079  
DBNAME( 'auditLog' 'auditLog' )  
ACCESS-CLASS critical  
LENGTH 1024 )

attributetypes=( 1.3.18.0.2.4.1072  
NAME 'ibm-auditModify'  
DESC 'TRUE or FALSE. Indicate whether to log the Modify operation.  
Default is FALSE.'  
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7



```

SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.1072
DBNAME('auditModify' 'auditModify')
ACCESS-CLASS critical
LENGTH 16)

attributetypes=(1.3.18.0.2.4.1075
NAME 'ibm-auditModifyDN'
DESC 'TRUE or FALSE. Indicate whether to log the ModifyRDN
operation. Default is FALSE.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.1075
DBNAME('auditModifyDN' 'auditModifyDN')
ACCESS-CLASS critical
LENGTH 16)

attributetypes=(1.3.18.0.2.4.1074
NAME 'ibm-auditSearch'
DESC 'TRUE or FALSE. Indicate whether to log the Search operation.
Default is FALSE.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.1074
DBNAME('auditSearch' 'auditSearch')
ACCESS-CLASS critical
LENGTH 16)

attributetypes=(1.3.18.0.2.4.1076
NAME 'ibm-auditUnbind'
DESC 'TRUE or FALSE. Indicate whether to log the Unbind operation.
Default is FALSE.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.1076
DBNAME('auditUnbind' 'auditUnbind')
ACCESS-CLASS critical
LENGTH 16)

attributetypes=(1.3.18.0.2.4.2483
NAME 'ibm-capabilitiesubentry'
DESC 'Names the ibm-capabilitiesubentry object listing the
capabilities of the naming context containing this object.'
EQUALITY 2.5.13.1
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 SINGLE-VALUE
NO-USER-MODIFICATION
USAGE dSAOperation)
IBMAttributetypes=(1.3.18.0.2.4.2483
DBNAME('ibmcapsubentry' 'ibmcapsubentry')
ACCESS-CLASS system
LENGTH 1000)

attributetypes=(1.3.18.0.2.4.2444
NAME 'ibm-effectiveAcl'
DESC 'An operational attribute that contains the accumulated filter
based effective access for entries in an IBM LDAP directory.'
EQUALITY 2.5.13.2
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 NO-USER-MODIFICATION
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2444
DBNAME('effectiveAcl' 'effectiveAcl')
ACCESS-CLASS restricted
LENGTH 32700)

```

```

attributetypes=(1.3.18.0.2.4.2331
NAME 'ibm-effectiveReplicationModel'
DESC 'Advertises in the Root DSE the OID of the replication model in
use by the server'
EQUALITY 1.3.6.1.4.1.1466.109.114.1
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
SINGLE-VALUE NO-USER-MODIFICATION
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2331
DBNAME('effectiveReplicat' 'effectiveReplicat')
ACCESS-CLASS system
LENGTH 240)

```

```

attributetypes=(1.3.18.0.2.4.2482
NAME 'ibm-enabledCapabilities'
DESC 'Lists capabilities that are enabled for use on this server.'
EQUALITY 2.5.13.2
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
NO-USER-MODIFICATION
USAGE dSAOperation)
IBMAttributetypes=(1.3.18.0.2.4.2482
DBNAME('ibmenabledcap' 'ibmenabledcap')
ACCESS-CLASS system
LENGTH 100)

```

```

attributetypes=(1.3.18.0.2.4.2325
NAME 'ibm-entryChecksum'
DESC 'A checksum of the user attributes for the entry containing
this attribute.'
EQUALITY 1.3.6.1.4.1.1466.109.114.1
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
SINGLE-VALUE
NO-USER-MODIFICATION
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2325
DBNAME('entryChecksum' 'entryChecksum')
ACCESS-CLASS system
LENGTH 100)

```

```

attributetypes=(1.3.18.0.2.4.2326
NAME 'ibm-entryChecksumOp'
DESC 'A checksum of the replicated operational attributes for the
entry containing this attribute.'
EQUALITY 1.3.6.1.4.1.1466.109.114.1
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
SINGLE-VALUE NO-USER-MODIFICATION
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2326
DBNAME('entryChecksumOp' 'entryChecksumOp')
ACCESS-CLASS system
LENGTH 100)

```

```

attributetypes=(1.3.18.0.2.4.1780
NAME 'ibm-entryUuid'
DESC 'Uniquely identifies a directory entry throughout its life.'
EQUALITY 2.5.13.2
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
NO-USER-MODIFICATION
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.1780
DBNAME('ibmEntryUuid' 'ibmEntryUuid')
ACCESS-CLASS system
LENGTH 36
EQUALITY)

```

```

attributetypes=(1.3.18.0.2.4.2443
NAME 'ibm-filterAcIEntry'
DESC 'Contains filter based access controls for entries in an IBM
LDAP directory.'
EQUALITY 2.5.13.2
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2443
DBNAME('filterAcIEntry' 'filterAcIEntry')
ACCESS-CLASS restricted
LENGTH 32700)

attributetypes=(1.3.18.0.2.4.2445
NAME 'ibm-filterAcIInherit'
DESC 'Indicates whether filter based ACLs should accumulate up the
ancestor tree.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2445
DBNAME('filterAcIInherit' 'filterAcIInherit')
ACCESS-CLASS restricted
LENGTH 5)

attributetypes=(1.3.18.0.2.4.2330
NAME 'ibm-replicationChangeLDIF'
DESC 'Provides LDIF representation of the last failing operation'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.5
SINGLE-VALUE
NO-USER-MODIFICATION
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2330
DBNAME('replicationChange' 'replicationChange')
ACCESS-CLASS system)

attributetypes=(1.3.18.0.2.4.2498
NAME 'ibm-replicationIsQuiesced'
DESC 'Indicates whether the replicated subtree containing this
attribute is quiesced on this server.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7 S
INGLE-VALUE
NO-USER-MODIFICATION
USAGE dSAOperation)
IBMAttributetypes=(1.3.18.0.2.4.2498
DBNAME('replIsQuiesced' 'replIsQuiesced')
ACCESS-CLASS system
LENGTH 5)

attributetypes=(1.3.18.0.2.4.2338
NAME 'ibm-replicationLastActivationTime'
DESC 'Indicates the last time the replication thread was activated'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.24
SINGLE-VALUE
NO-USER-MODIFICATION
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2338
DBNAME('replicationLastAc' 'replicationLastAc')
ACCESS-CLASS system
LENGTH 32)

attributetypes=(1.3.18.0.2.4.2334
NAME 'ibm-replicationLastChangeId'
DESC 'Indicates last change ID successfully replicated for a
replication agreement'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
NO-USER-MODIFICATION

```

```

USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2334
DBNAME('replicationLastCh' 'replicationLastCh')
ACCESS-CLASS system
LENGTH 11)

attributetypes=(1.3.18.0.2.4.2335
NAME 'ibm-replicationLastFinishTime'
DESC 'Indicates the last time the replication thread completed
sending all of the pending entries.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.24
SINGLE-VALUE
NO-USER-MODIFICATION
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2335
DBNAME('replicationLastFi' 'replicationLastFi')
ACCESS-CLASS system
LENGTH 30)

attributetypes=(1.3.18.0.2.4.2448
NAME 'ibm-replicationLastGlobalChangeId'
DESC 'Indicates the ID of the last global (applies to the entire
DIT, such as schema) change successfully replicated.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE
NO-USER-MODIFICATION
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2448
DBNAME('replicationLastGl' 'replicationLastGl')
ACCESS-CLASS normal
LENGTH 11)

attributetypes=(1.3.18.0.2.4.2340
NAME 'ibm-replicationLastResult'
DESC 'Result of last attempted replication in the form:
<time><change id><resultcode> <entry-dn> '
EQUALITY 2.5.13.2
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE NO-USER-MODIFICATION
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2340
DBNAME('replicationLastRe' 'replicationLastRe')
ACCESS-CLASS system
LENGTH 2048)

attributetypes=(1.3.18.0.2.4.2332
NAME 'ibm-replicationLastResultAdditional'
DESC 'Provides any additional error information returned by the
consuming server in the message component of the LDAP result'
EQUALITY 2.5.13.2
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
NO-USER-MODIFICATION
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2332
BNAME('replicationLastAd' 'replicationLastAd')
ACCESS-CLASS system
LENGTH 1024)

attributetypes=(1.3.18.0.2.4.2339
NAME 'ibm-replicationNextTime'
DESC 'Indicates next scheduled time for replication'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.24
SINGLE-VALUE
NO-USER-MODIFICATION
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2339
DBNAME('replicationNextTi' 'replicationNextTi')
ACCESS-CLASS system

```

```

LENGTH 30)

attributetypes=(1.3.18.0.2.4.2333
NAME 'ibm-replicationPendingChangeCount'
DESC 'Indicates the total number of pending unreplicated changes for
this replication agreement'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
NO-USER-MODIFICATION
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2333
DBNAME('replicationPendig' 'replicationPendig')
ACCESS-CLASS system
LENGTH 12)

attributetypes=(1.3.18.0.2.4.2337
NAME 'ibm-replicationPendingChanges'
DESC 'Unreplicated change in the form
<change id><operation> <dn>
where operation is ADD, DELETE, MODIFY, MODIFYDN'
EQUALITY 2.5.13.2
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
NO-USER-MODIFICATION
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2337
DBNAME('replicationPendch' 'replicationPendch')
ACCESS-CLASS system
LENGTH 1100)

attributetypes=(1.3.18.0.2.4.2336
NAME 'ibm-replicationState'
DESC 'Indicates the state of the replication thread:
active,ready,waiting,suspended, or full; if full, the value will
indicate the amount of progress'
EQUALITY 1.3.6.1.4.1.1466.109.114.1
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
SINGLE-VALUE
NO-USER-MODIFICATION
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2336
DBNAME('replicationState' 'replicationState')
ACCESS-CLASS system
LENGTH 240)

attributetypes=(1.3.18.0.2.4.2495
NAME 'ibm-replicationThisServerIsMaster'
DESC 'Indicates whether the server returning this attribute is a
master server for the subtree containing this entry.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
SINGLE-VALUE NO-USER-MODIFICATION
USAGE dSAOperation)
IBMAttributetypes=(1.3.18.0.2.4.2495
DBNAME('replThisSvrMast' 'replThisSvrMast')
ACCESS-CLASS system
LENGTH 5)

attributetypes=(1.3.18.0.2.4.2328
NAME 'ibm-serverId'
DESC 'Advertises in the Root DSE the ibm-slapdServerId configuration
setting'
EQUALITY 1.3.6.1.4.1.1466.109.114.1
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
SINGLE-VALUE
NO-USER-MODIFICATION
USAGE dSAOperation)
IBMAttributetypes=(1.3.18.0.2.4.2328
DBNAME('serverId' 'serverId')

```

```

ACCESS-CLASS system
LENGTH 240)

attributetypes=(1.3.18.0.2.4.2374
NAME 'ibm-slapdACLCache'
DESC 'Controls whether or not the server caches ACL information'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2374
DBNAME('ACLCache' 'ACLCache')
ACCESS-CLASS normal
LENGTH 5)

attributetypes=(1.3.18.0.2.4.2373
NAME 'ibm-slapdACLCacheSize'
DESC 'Maximum number of entries to keep in the ACL Cache'
EQUALITY 2.5.13.14
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 S
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2373
DBNAME('slapdACLCacheSize' 'slapdACLCacheSize')
ACCESS-CLASS normal
LENGTH 11)

attributetypes=(1.3.18.0.2.4.2428
NAME 'ibm-slapdAdminDN'
DESC 'Bind DN for ibmslapd administrator, e.g.: cn=root'
EQUALITY 2.5.13.1
ORDERING 1.3.18.0.2.4.405
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2428
DBNAME('slapdAdminDN' 'slapdAdminDN')
ACCESS-CLASS critical
LENGTH 1000
EQUALITY ORDERING)

attributetypes=(1.3.18.0.2.4.2425
NAME 'ibm-slapdAdminPW'
DESC 'Bind password for ibmslapd administrator.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.5
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2425
DBNAME('slapdAdminPW' 'slapdAdminPW')
ACCESS-CLASS critical)

attributetypes=(1.3.18.0.2.4.2366
NAME 'ibm-slapdAuthIntegration'
DESC 'Specifies integration of LDAP administrator access with local
OS users. Legal values are : 0 - do not map local OS users to LDAP
administrator, 1 - map local OS users with proper authority to LDAP
administrator. This is supported only on i5/OS.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2366
DBNAME('slapdAuthIntegrat' 'slapdAuthIntegrat')
ACCESS-CLASS system
LENGTH 11)

attributetypes=(1.3.18.0.2.4.2432
NAME 'ibm-slapdCLIErrors'

```

```

DESC 'File path or device on ibmslapd host machine to which DB2 CLI
error messages will be written. On Windows, forward slashes are
allowed, and a leading slash not preceded by a drive letter is
assumed to be rooted at the install directory (i.e.: /tmp/cli.errors
= D:\Program Files\IBM\ldap\tmp\cli.errors).'
EQUALITY 2.5.13.5
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2432
DBNAME('slapdCLIErrors' 'slapdCLIErrors')
ACCESS-CLASS normal
LENGTH 1024)

attributetypes=(1.3.18.0.2.4.3116
NAME 'ibm-slapdCryptoSync'
DESC 'A key stash file consistency marker string.
It is queried by the server atstart up as part of
a verification process to ensure that the key stash
files match any data that has been two-way encrypted.'
EQUALITY 2.5.13.17
SYNTAX 1.3.6.1.4.1.1466.115.121.1.5
SINGLE-VALUE
NO-USER-MODIFICATION
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.3116
DBNAME('CryptoSync' 'CryptoSync')
ACCESS-CLASS system)

attributetypes=(1.3.18.0.2.4.2369
NAME 'ibm-slapdDB2CP'
DESC 'Specifies the Code Page of the directory database. 1208 is
the code page for UTF-8 databases.'
EQUALITY 2.5.13.5
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2369
DBNAME('slapdDB2CP' 'slapdDB2CP')
ACCESS-CLASS normal
LENGTH 11)

attributetypes=(1.3.18.0.2.4.2431
NAME 'ibm-slapdDBAlias'
DESC 'The DB2 database alias.'
EQUALITY 2.5.13.5
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 S
INGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2431
DBNAME('slapdDBAlias' 'slapdDBAlias')
ACCESS-CLASS normal L
LENGTH 8)

attributetypes=(1.3.18.0.2.4.2417
NAME 'ibm-slapdDbConnections'
DESC 'Specify the number of DB2 connections the server will dedicate
to the DB2 backend. The value must be between 5 & 50 (inclusive).
The ODBCCONS environment variable overrides this value. If
ibm-slapdDbConnections (or ODBCCONS) is less than 5 or greater than
50, the server will use 5 or 50 respectively. Additional connections
may be created for replication and change log.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2417
DBNAME('DbConnections' 'DbConnections')

```

```

ACCESS-CLASS critical
LENGTH 2)

attributetypes=(1.3.18.0.2.4.2418
NAME 'ibm-slapdDbInstance'
DESC 'The DB2 database instance for this backend.'
EQUALITY 2.5.13.5
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2418
DBNAME('slapdDbInstance' 'slapdDbInstance')
ACCESS-CLASS critical
LENGTH 8)

attributetypes=(1.3.18.0.2.4.2382
NAME 'ibm-slapdDbLocation'
DESC 'The file system path where the backend database is located. On
UNIX or Linux this is usually the home directory of the DB2INSTANCE owner
(e.g.: /home/ldapdb2). On windows its just a drive specifier (e.g.: D:)'
EQUALITY 2.5.13.5
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2382
DBNAME('slapdDbLocation' 'slapdDbLocation')
ACCESS-CLASS critical
LENGTH 1024)

attributetypes=(1.3.18.0.2.4.2426
NAME 'ibm-slapdDbName'
DESC 'The DB2 database name for this backend.'
EQUALITY 2.5.13.5
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2426
DBNAME('slapdDbName' 'slapdDbName')
ACCESS-CLASS critical
LENGTH 8)

attributetypes=(1.3.18.0.2.4.2422
NAME 'ibm-slapdDbUserID'
DESC 'The user name with which to connect to the DB2 database for
this backend.'
EQUALITY 2.5.13.5
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2422
DBNAME('slapdDbUserID' 'slapdDbUserID')
ACCESS-CLASS critical
LENGTH 8)

attributetypes=(1.3.18.0.2.4.2423
NAME 'ibm-slapdDbUserPW'
DESC 'The userpassword with which to connect to the DB2 database
for this backend.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.5
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2423
DBNAME('slapdDbUserPW' 'slapdDbUserPW')
ACCESS-CLASS critical)

attributetypes=(OID TBD
NAME 'ibm-slapdDerefAliases'
DESC 'Maximum alias dereferencing level on search requests, regardless of

```



any derefAliases that may have been specified on the client requests. Allowed values are "never", "find", "search" and "always".'

```
EQUALITY 2.5.13.2
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.3054
DBNAME('DerefAliases' 'DerefAliases')
ACCESS-CLASS critical
LENGTH 6)
```

```
attributetypes=(1.3.18.0.2.4.2449
NAME 'ibm-slapdDN' DESC 'This attribute is used to sort search
results by the entry DN (LDAP_ENTRY.DN column in the LDAPDB2
database).'
```

```
EQUALITY 2.5.13.1
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
SINGLE-VALUE NO-USER-MODIFICATION
USAGE dSAOperation)
IBMAttributetypes=(1.3.18.0.2.4.2449
DBNAME('LDAP_ENTRY' 'DN')
ACCESS-CLASS system
LENGTH 1000)
```

```
attributetypes=(1.3.18.0.2.4.2481
NAME 'ibm-supportedCapabilities'
DESC 'Lists capabilities supported, but necessarily enabled, by this
server.'
```

```
QUALITY 2.5.13.2
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
NO-USER-MODIFICATION
USAGE dSAOperation)
IBMAttributetypes=(1.3.18.0.2.4.2481
DBNAME('ibmsupportedCap' 'ibmsupportedCap')
ACCESS-CLASS system
LENGTH 100)
```

```
attributetypes=(1.3.18.0.2.4.2421
NAME 'ibm-slapdEnableEventNotification'
DESC 'If set to FALSE, the server will reject all extended
operation requests to register for event notification with the
extended result LDAP_UNWILLING_TO_PERFORM.'
```

```
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2421 D
BNAME('enableEvtNotify' 'enableEvtNotify')
ACCESS-CLASS critical
LENGTH 5)
```

```
attributetypes=(1.3.18.0.2.4.2372
NAME 'ibm-slapdEntryCacheSize'
DESC 'Maximum number of entries to keep in the entry cache'
```

```
EQUALITY 2.5.13.14
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2372
DBNAME('slapdRDBMCacheSiz' 'slapdRDBMCacheSiz')
ACCESS-CLASS normal
LENGTH 11)
```

```
attributetypes=(1.3.18.0.2.4.2424
NAME 'ibm-slapdErrorLog'
DESC 'File path or device on the ibmslapd host machine
to which error messages will be written. On Windows, forward
slashes are allowed, and a leading slash not preceded by a drive
letter is assumed to be rooted at the install directory (i.e.:
```

```

/tmp/slapd.errors = D:\Program Files\IBM\ldap\tmp\slapd.errors).'
EQUALITY 2.5.13.5
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2424
DBNAME('slapdErrorLog' 'slapdErrorLog')
ACCESS-CLASS critical
LENGTH 1024)

attributetypes=(1.3.18.0.2.4.2371
NAME 'ibm-slapdFilterCacheBypassLimit'
DESC 'Search filters that match more than this number of entries
will not be added to the Search Filter cache. Because the list of
entry ids that matched the filter are included in this cache, this
setting helps to limit memory use. A value of 0 indicates no
limit.'
EQUALITY 2.5.13.14
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2371
DBNAME('slapdRDBMCacheByP' 'slapdRDBMCacheByP')
ACCESS-CLASS normal
LENGTH 11)

attributetypes=(1.3.18.0.2.4.2370
NAME 'ibm-slapdFilterCacheSize'
DESC 'Specifies the maximum number of entries to keep in the Search
Filter Cache.'
EQUALITY 2.5.13.14
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2370
DBNAME('slapdFilterCacheS' 'slapdFilterCacheS')
ACCESS-CLASS normal
LENGTH 11)

attributetypes=(1.3.18.0.2.4.2378
NAME 'ibm-slapdIdleTimeOut'
DESC 'Reserved for future use.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2378
DBNAME('SlapdIdleTimeOut' 'SlapdIdleTimeOut')
ACCESS-CLASS critical
LENGTH 11)

attributetypes=(1.3.18.0.2.4.2364
NAME 'ibm-slapdIncludeSchema'
DESC 'File path on the ibmslapd host machine containing schema
definitions used by the LDCF backend. Standard values are:
/etc/V3.system.at /etc/V3.system.oc
/etc/V3.ibm.at /etc/V3.ibm.oc /etc/V3.user.at /etc/V3.user.oc
/etc/V3.ldapsyntaxes /etc/V3.matchingrules /etc/V3.modifiedschema
On Windows, forward slashes are allowed, and a leading slash not
preceded by a drive letter is assumed to be rooted at the install
directory (i.e.: /etc/V3.system.at =
D:\Program Files\IBM\ldap\etc\V3.system.at).'
EQUALITY 2.5.13.5
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2364
DBNAME('slapdIncludeSchema' 'slapdIncludeSchema')
ACCESS-CLASS critical
LENGTH 1024)

```

```

attributetypes=(1.3.18.0.2.4.2365
NAME 'ibm-slapdIpAddress'
DESC 'Specifies IP addresses the server will listen on. These can
be IPv4 or IPv6 addresses. If the attribute is not specified, the
server uses all IP addresses assigned to the host machine. This is
supported on i5/OS only.'
EQUALITY 1.3.6.1.4.1.1466.109.114.1
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2365
DBNAME('slapdIpAddress' 'slapdIpAddress')
ACCESS-CLASS system
LENGTH 32)

```

```

attributetypes=(1.3.18.0.2.4.2420
NAME 'ibm-slapdKrbAdminDN'
DESC 'Specifies the kerberos ID of the LDAP administrator (e.g.
ibm-kn=name@realm). Used when kerberos authentication is used to
authenticate the administrator when logged onto the Web Admin
interface. This is specified instead of adminDN and adminPW.'
EQUALITY 2.5.13.5
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2420
DBNAME('slapdKrbAdminDN' 'slapdKrbAdminDN')
ACCESS-CLASS critical
LENGTH 512)

```

```

attributetypes=(1.3.18.0.2.4.2394
NAME 'ibm-slapdKrbEnable'
DESC 'Must be one of { TRUE | FALSE }. Specifies whether the
server supports kerberos authentication.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2394
DBNAME('slapdKrbEnable' 'slapdKrbEnable')
ACCESS-CLASS critical
LENGTH 5)

```

```

attributetypes=(1.3.18.0.2.4.2419
NAME 'ibm-slapdKrbIdentityMap'
DESC 'If set to TRUE, when a client is authenticated with a
kerberos ID, the server will search for a local user with matching
kerberos credentials, and add that user DN to the connections
bind credentials. This allows ACLs based on LDAP user DNS to still
be usable with kerberos authentication.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2419
DBNAME('KrbIdentityMap' 'KrbIdentityMap')
ACCESS-CLASS critical
LENGTH 5)

```

```

attributetypes=(1.3.18.0.2.4.2416
NAME 'ibm-slapdKrbKeyTab'
DESC 'Specifies the LDAP servers keytab file. This file contains the
LDAP servers private key, as associated with its kerberos account.
This file should be protected (like the servers SSL key database
file).
On Windows, forward slashes are allowed, and a leading slash not
preceded by a drive letter (D:) is assumed to be rooted at the
install directory (i.e.: /tmp/slapd.errors =
D:\Program Files\IBM\ldap\tmp\slapd.errors).'
EQUALITY 2.5.13.5
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15

```

```

SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2416
DBNAME('slapdKrbKeyTab' 'slapdKrbKeyTab')
ACCESS-CLASS critical
LENGTH 1024)

attributetypes=(1.3.18.0.2.4.2400
NAME 'ibm-slapdKrbRealm'
DESC 'Specifies the LDAP servers kerberos realm. Used to publish
the ldapservername attribute in the root DSE. Note that an LDAP
server can serve as the repository of account information for
multiple KDCs (and realms), but the LDAP server, as a kerberos
server, can only be a member of a single realm.'
EQUALITY 2.5.13.2
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2400
DBNAME('slapdKrbRealm' 'slapdKrbRealm')
ACCESS-CLASS critical
LENGTH 256)

attributetypes=(1.3.18.0.2.4.2415
NAME 'ibm-slapdLdapCrIHost'
DESC 'Specify the hostname of the LDAP server that contains the
Certificate Revocation Lists (CRLs) for validating client x.509v3
certificates. This parameter is needed when
ibm-slapdSslAuth=serverclientauth AND the client certificates
have been issued for CRL validation'
EQUALITY 2.5.13.2
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2415
DBNAME('LdapCrIHost' 'LdapCrIHost')
ACCESS-CLASS critical
LENGTH 256)

attributetypes=(1.3.18.0.2.4.2407
NAME 'ibm-slapdLdapCrIPassword'
DESC 'Specify the password that server-side SSL will use to bind to
the LDAP server that contains the Certificate Revocation Lists
(CRLs) for validating client x.509v3 certificates. This parameter
may be needed when ibm-slapdSslAuth=serverclientauth AND the client
certificates have been issued for CRL validation. Note: If the
LDAP server holding the CRLs permits unauthenticated
access to the CRLs (i.e. anonymous access), then
ibm-slapdLdapCrIPassword is not required.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.5
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2407
DBNAME('CrIPassword' 'CrIPassword')
ACCESS-CLASS critical)

attributetypes=(1.3.18.0.2.4.2404
NAME 'ibm-slapdLdapCrIPort'
DESC 'Specify the LDAP ibm-slapdPort used by the LDAP server that
contains the Certificate Revocation Lists (CRLs) for validating
client x.509v3 certificates. This parameter is needed when
ibm-slapdSslAuth=serverclientauth AND the client certificates have
been issued for CRL validation. (IP ports are unsigned, 16-bit
integers in the range 1 - 65535)'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE directoryOperation)

```

```
BMAtributetypes=(1.3.18.0.2.4.2404
DBNAME('LdapCrIPort' 'LdapCrIPort')
ACCESS-CLASS critical
LENGTH 11)
```

```
attributetypes=(1.3.18.0.2.4.2403
NAME 'ibm-slapdLdapCrIUser'
DESC 'Specify the bindDN that server-side SSL will use to bind to
the LDAP server that contains the Certificate Revocation Lists
(CRLs) for validating client x.509v3 certificates. This parameter
may be needed when ibm-slapdSslAuth=serverclientauth AND the client
certificates have been issued for CRL validation.
```

Note:

If the LDAP server holding the CRLs permits unauthenticated access to the CRLs (i.e. anonymous access), then ibm-slapdLdapCrIUser is not required.'

```
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
SINGLE-VALUE
```

```
USAGE directoryOperation)
```

```
IBMAtributetypes=(1.3.18.0.2.4.2403
DBNAME('LdapCrIUser' 'LdapCrIUser')
ACCESS-CLASS critical
LENGTH 1000)
```

```
attributetypes=(1.3.18.0.2.4.2409
NAME 'ibm-slapdMasterDN'
DESC 'Bind DN used by a replication supplier server. The value has
to match the replicaBindDN in the credentials object associated
with the replication agreement defined between the servers.
When kerberos is used to authenticate to the replica,
ibm-slapdMasterDN must specify the DN representation of the
kerberos ID (e.g. ibm-kn=freddy@realm1). When kerberos is used,
MasterServerPW is ignored.'
```

```
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
SINGLE-VALUE
```

```
USAGE directoryOperation)
```

```
IBMAtributetypes=(1.3.18.0.2.4.2409
DBNAME('MasterDN' 'MasterDN')
ACCESS-CLASS critical
LENGTH 1000)
```

```
attributetypes=(1.3.18.0.2.4.2411
NAME 'ibm-slapdMasterPW'
DESC 'Bind password used by a replication supplier. The value has to
match the replicaBindPW in the credentials object associated with
the replication agreement defined between the servers. When kerberos
is used, MasterServerPW is ignored.'
```

```
SYNTAX 1.3.6.1.4.1.1466.115.121.1.5
SINGLE-VALUE
```

```
USAGE directoryOperation)
```

```
IBMAtributetypes=(1.3.18.0.2.4.2411
DBNAME('MasterPW' 'MasterPW')
ACCESS-CLASS critical)
```

```
attributetypes=(1.3.18.0.2.4.2401
NAME 'ibm-slapdMasterReferral'
DESC 'URL of a master replica server (e.g.:
ldaps://master.us.ibm.com:636)'
```

```
EQUALITY 2.5.13.2
```

```
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
```

```
SINGLE-VALUE USAGE directoryOperation)
```

```
IBMAtributetypes=(1.3.18.0.2.4.2401
DBNAME('MasterReferral' 'MasterReferral')
ACCESS-CLASS critical
LENGTH 256)
```

```
attributetypes=(1.3.18.0.2.4.2412
```

```

NAME 'ibm-slapdMaxEventsPerConnection'
DESC 'Maximum number of event notifications which can be registered
per connection. Minimum = 0 (unlimited) Maximum = 2,147,483,647'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE
directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2412
DBNAME('EventsPerCon' 'EventsPerCon')
ACCESS-CLASS critical
LENGTH 11)

attributetypes=(1.3.18.0.2.4.2405
NAME 'ibm-slapdMaxEventsTotal'
DESC 'Maximum total number of event notifications which can be
registered for all connections. Minimum = 0 (unlimited) Maximum =
2,147,483,647'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2405
DBNAME('MaxEventsTotal' 'MaxEventsTotal')
ACCESS-CLASS critical
LENGTH 11)

attributetypes=(1.3.18.0.2.4.2439
NAME 'ibm-slapdMaxNumOfTransactions'
DESC 'Maximum number of transactions active at one time. 0 = unlimited'
EQUALITY 2.5.13.29
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2439
DBNAME('MaxNumOfTrans' 'MaxNumOfTrans')
ACCESS-CLASS critical
LENGTH 11
EQUALITY ORDERING SUBSTR APPROX)

attributetypes=(1.3.18.0.2.4.2385
NAME 'ibm-slapdMaxOpPerTransaction'
DESC 'Maximum number of operations per transaction. 0 = unlimited'
EQUALITY 2.5.13.29
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2385
DBNAME('MaxOpPerTrans' 'MaxOpPerTrans')
ACCESS-CLASS critical
LENGTH 11
EQUALITY ORDERING APPROX)

attributetypes=(1.3.18.0.2.4.2386
NAME 'ibm-slapdMaxTimeLimitOfTransactions'
DESC 'The maximum timeout value of a pending transaction in
seconds. 0 = unlimited'
EQUALITY 2.5.13.29
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2386
DBNAME('MaxTimeOfTrans' 'MaxTimeOfTrans')
ACCESS-CLASS critical
LENGTH 11
EQUALITY ORDERING APPROX)

attributetypes=(1.3.18.0.2.4.2500
NAME 'ibm-slapdMigrationInfo'
DESC 'Information used to control migration of a component.'

```

```

EQUALITY 2.5.13.2
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2500
DBNAME('slapdMigrationInf' 'slapdMigrationInf')
ACCESS-CLASS critical
LENGTH 2048)

attributetypes=(1.3.18.0.2.4.2376
NAME 'ibm-slapdPagedResAllowNonAdmin'
DESC 'Whether or not the server should allow non-Administrator
bind for paged results requests on a search request. If the value
read from the ibmslapd.conf file is TRUE, the server will process
any client request, including those submitted by a user binding
anonymously. If the value read from the ibmslapd.conf file is
FALSE, the server will process only those client requests submitted
by a user with Administrator authority. If a client requests paged
results with a criticality of TRUE or FALSE for a search operation,
does not have Administrator authority, and the value read from the
ibmslapd.conf file for this attribute is FALSE, the server will
return to the client with return code insufficientAccessRights - no
searching or paging will be performed. '
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2376
DBNAME('SlapdPagedNonAdmn' 'SlapdPagedNonAdmn')
ACCESS-CLASS critical
LENGTH 5)

attributetypes=(1.3.18.0.2.4.2380
NAME 'ibm-slapdPagedResLmt'
DESC 'Maximum number of outstanding paged results search requests
allowed active simultaneously. Range = 0.... If a client requests
a paged results operation, and a maximum number of outstanding paged
results are currently active, then the server will return to the
client with return code of busy - no searching or paging will be
performed.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2380
DBNAME('SlapdPagedResLmt' 'SlapdPagedResLmt')
ACCESS-CLASS critical
LENGTH 11)
attributetypes=(1.3.18.0.2.4.2406
NAME 'ibm-slapdPlugin'
DESC 'A plugin is a dynamically loaded library which extends the
capabilities of the server. An ibm-slapdPlugin attribute specifies
to the server how to load and initialize a plugin library. The
syntax is: keyword filename init_function [args...]. The syntax
will be slightly different for each platform due to library
naming conventions.'
EQUALITY 2.5.13.5
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2406
DBNAME('slapdPlugin' 'slapdPlugin')
ACCESS-CLASS critical
LENGTH 2000)

attributetypes=(1.3.18.0.2.4.2408
NAME 'ibm-slapdPort'
DESC 'TCP/IP ibm-slapdPort used for non-SSL connections.
Can not have the same value as ibm-slapdSecurePort. (IP ports are
unsigned, 16-bit integers in the range 1 - 65535)'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27

```

```

SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2408
DBNAME('slapdPort' 'slapdPort')
ACCESS-CLASS critical
LENGTH 5)

attributetypes=(1.3.18.0.2.4.2402
NAME 'ibm-slapdPwEncryption'
DESC 'Must be one of { none | AES128 | AES192 | AES256 |crypt | sha }.
Specify the encoding mechanism for the user passwords before they are
stored in the directory. Defaults to none if unspecified. If the
value is set other than none, SASL digest-md5 bind will fail.'
EQUALITY 2.5.13.2
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2402
DBNAME('PwEncryption' 'PwEncryption')
ACCESS-CLASS critical
LENGTH 5)

attributetypes=(1.3.18.0.2.4.2413
NAME 'ibm-slapdReadOnly'
DESC 'Must be one of { TRUE | FALSE }. Specifies whether
the backend can be written to. Defaults to FALSE if unspecified. If
setto TRUE, the server will return LDAP_UNWILLING_TO_PERFORM (0x35)
in response to any client request which would change data in the
readOnly database.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2413
DBNAME('ReadOnly' 'ReadOnly')
ACCESS-CLASS critical
LENGTH 5)

attributetypes=(1.3.18.0.2.4.2487
NAME 'ibm-slapdReferral'
DESC 'Specify the referral LDAP URL to pass back when the local
suffixes do not match the request. Used for superior referral
(i.e. ibm-slapdSuffix is not within the servers naming context).'
EQUALITY 2.5.13.5
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2487
DBNAME('Referral' 'Referral')
ACCESS-CLASS critical
LENGTH 32700)

attributetypes=(1.3.18.0.2.4.2437
NAME 'ibm-slapdSchemaAdditions'
DESC 'File path on the ibmslapd host machine containing additional
schema definitions used by the LDCF backend. Standard values are:
/etc/V3.modifiedschema On Windows, forward slashes are allowed,
and a leading slash not preceded by a drive letter is assumed to be
rooted at the install directory (i.e.: /etc/V3.system.at=
D:\Program Files\IBM\ldap\etc\V3.system.at).'
EQUALITY 2.5.13.5
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2437
DBNAME('slapdSchemaAdditi' 'slapdSchemaAdditi')
ACCESS-CLASS normal
LENGTH 1024)

attributetypes=(1.3.18.0.2.4.2363

```



```

NAME 'ibm-slapdSchemaCheck'
DESC 'Must be one of { V2 | V3 | V3_lenient}. Specifies schema
checking mechanism for add/modify operation. V2 = perform LDAP v2
checking. V3 = perform LDAP v3 checking. V3_lenient = not ALL
parent object classes are required. Only the immediate object class
is needed when adding entries.'
EQUALITY 2.5.13.2
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2363
DBNAME('SchemaCheck' 'SchemaCheck')
ACCESS-CLASS critical
LENGTH 10)

attributetypes=(1.3.18.0.2.4.2398
NAME 'ibm-slapdSecurePort'
DESC 'TCP/IP port used for SSL connections. Can not have the same
value as ibm-slapdPort. (IP ports are unsigned, 16-bit integers in
the range 1 - 65535)'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2398
DBNAME('SecurePort' 'SecurePort')
ACCESS-CLASS critical
LENGTH 5)

attributetypes=(1.3.18.0.2.4.2399
NAME 'ibm-slapdSecurity'
DESC 'Must be one of { none | SSL | SSLOnly }. Specifies types of
connections accepted by the server. none - server listens on
non-ssl port only. ssl - server listens on both ssl and non-ssl
ports. sslonly - server listens on ssl port only.'
EQUALITY 2.5.13.2
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2399
DBNAME('Security' 'Security')
ACCESS-CLASS critical
LENGTH 7)

attributetypes=(1.3.18.0.2.4.2397
NAME 'ibm-slapdSetenv'
DESC 'Server executes putenv() for all values of ibm-slapdSetenv
at startup to modify its own runtime environment. Shell variables
(%PATH% or %LANG) will not be expanded. The only current use for
this attribute is to set DB2CODEPAGE=1208, which is required if
using UCS-2 (Unicode) databases.'
EQUALITY 2.5.13.5
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2397
DBNAME('slapdSetenv' 'slapdSetenv')
ACCESS-CLASS critical
LENGTH 2000)

attributetypes=(1.3.18.0.2.4.2396
NAME 'ibm-slapdSizeLimit'
DESC 'Maximum number of entries to return from search, regardless of
any sizelimit that may have been specified on the client search
request. Range = 0... If a client has passed a limit, then the
smaller value of the client value and the value read from
ibmslapd.conf will be used. If a client has not passed a limit and
has bound as admin DN, then the limit will be considered unlimited.
If the client has not passed a limit and has not bound as admin DN,

```

then the limit will be that which was read from ibmslapd.conf file.  
0 = unlimited.'

```
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2396
DBNAME('SizeLimit' 'SizeLimit')
ACCESS-CLASS critical
LENGTH 11)
```

```
attributetypes=(1.3.18.0.2.4.2381
NAME 'ibm-slapdSortKeyLimit'
DESC 'Maximum number of sort conditions (keys) that can be specified
on a single search request. Range = 0... If a client has passed a
search request with more sort keys than the limit allows, and the
sorted search control criticality is FALSE, then the server will
honor the value read from ibmslapd.conf and ignore any sort keys
encountered after the limit has been reached - searching and
sorting will be performed. If a client has passed a search request
with more keys than the limit allows, and the sorted search control
criticality is TRUE, then the server will return to the client with
return code of adminLimitExceeded - no searching or sorting
will be performed.'
```

```
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2381
DBNAME('SlapdSortKeyLimit' 'SlapdSortKeyLimit')
ACCESS-CLASS critical
LENGTH 11)
```

```
attributetypes=(1.3.18.0.2.4.2377
NAME 'ibm-slapdSortSrchAllowNonAdmin'
DESC 'Whether or not the server should allow non-Administrator bind
for sort on a search request. If the value read from the
ibmslapd.conf file is TRUE, the server will process any client
request, including those submitted by a user binding anonymously.
If the value read from the ibmslapd.conf file is FALSE, the server
will process only those client requests submitted by a user with
Administrator authority. If a client requests sort with a
criticality of TRUE for a search operation, does not have
Administrator authority, and the value read from the ibmslapd.conf
file for this attribute is FALSE, the server will return to the
client with return code insufficientAccessRights - no searching or
sorting will be performed.'
```

```
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2377
BNAME('SlapdSortNonAdmin' 'SlapdSortNonAdmin')
ACCESS-CLASS critical
LENGTH 5)
```

```
attributetypes=(1.3.18.0.2.4.2395
NAME 'ibm-slapdSslAuth'
DESC 'Must be one of { serverauth | serverclientauth}. Specify
authentication type for ssl connection. serverauth - supports
server authentication at the client. serverclientauth - supports
both server and client authentication.'
```

```
EQUALITY 2.5.13.2
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2395
DBNAME('slapdSslAuth' 'slapdSslAuth')
ACCESS-CLASS critical
LENGTH 16)
```

```

attributetypes=(1.3.18.0.2.4.2389
NAME 'ibm-slapdSslCertificate'
DESC 'Specify the label that identifies the servers Personal
Certificate in the key database file. This label is specified
when the servers private key and certificate are created with the
ikmgui application. If ibm-slapdSslCertificate is not defined, the
default private key, as defined in the key database file, is used by
the LDAP server for SSL connections.'
EQUALITY 2.5.13.5
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2389
DBNAME('SslCertificate' 'SslCertificate')
ACCESS-CLASS critical
LENGTH 128)

```

```

attributetypes=(1.3.18.0.2.4.2429
NAME 'ibm-slapdSslCipherSpec'
ESC 'SSL Cipher Spec Value must be set to DES-56, RC2-40-MD5,
RC4-128-MD5, RC4-128-SHA, RC4-40-MD5,TripleDES-168, or AES. It
identifies the allowable encryption/decryption methods for
establishing a SSL connection between LDAP clients and the server.'
EQUALITY 1.3.6.1.4.1.1466.109.114.1
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2429
DBNAME('slapdSslCipherSpe' 'slapdSslCipherSpe')
ACCESS-CLASS normal
LENGTH 30)

```

```

attributetypes=(1.3.18.0.2.4.2362
NAME 'ibm-slapdSslCipherSpecs'
DESC 'This attribute is deprecated in favor of
ibm-slapdSslCipherSpec. Specifies a decimal number which identifies
the allowable encryption/decryption methods for establishing a SSL
connection between LDAP client(s) and the server. This number
represents the availability of the encryption/decryption methods
supported by the LDAP server. The pre-defined Cipher values and
their descriptions are: SLAPD_SSL_TRIPLE_DES_SHA_US 0x0A Triple DES
encryption with a 168-bit key and a SHA-1 MAC LAPD_SSL_DES_SHA_US
0x09DES encryption with a 56-bit key and a SHA-1 MAC
SLAPD_SSL_RC4_SHA_US 0x05 RC4 encryption with a 128-bit key and a
SHA-1 MAC SLAPD_SSL_RC4_MD5_US 0x04 RC4 encryption with a 128-bit
key and a MD5 MAC SLAPD_SSL_RC4_MD5_EXPORT 0x03 RC4 encryption
with a 40-bit key and a MD5 MAC SLAPD_SSL_RC2_MD5_EXPORT 0x06 RC2
encryption with a 40-bit key and a MD5 MAC'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2362
DBNAME('SslCipherSpecs' 'SslCipherSpecs')
ACCESS-CLASS critical
LENGTH 11)

```

```

attributetypes=(1.3.18.0.2.4.2375
NAME 'ibm-slapdSSLKeyDatabase'
DESC 'File path to the LDAP servers SSL key database file. This key
database file is used for handling SSL connections from LDAP
clients, as well as for creating secure SSL connections to replica
LDAP servers. On Windows, forward slashes are allowed, and a
leading slash not preceded by a drive specifier (D:) is assumed to
be rooted at the install directory (i.e.: /etc/key.kdb = D:\Program
Files\IBM\ldap\etc\key.kdb).'
EQUALITY 2.5.13.5
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE directoryOperation)

```

```
IBMAttributetypes=(1.3.18.0.2.4.2375
DBNAME('slapdSSLKeyDataba' 'slapdSSLKeyDataba')
ACCESS-CLASS critical
LENGTH 1024)
```

```
attributetypes=(1.3.18.0.2.4.2438
NAME 'ibm-slapdSSLKeyDatabasePW'
DESC 'Specify the password associated with the LDAP servers SSL key
database file, as specified on the ibm-slapdSslKeyDatabase
parameter. If the LDAP servers key database file has an associated
password stash file, then the ibm-slapdSslKeyDatabasePW parameter
can be omitted, or set to ibm-slapdSslKeyDatabasePW = none.
```

Note:

The password stash file must be located in the same directory as the key database file and it must have the same file name as the key database file, but with an extension of .sth, instead of .kdb'

```
SYNTAX 1.3.6.1.4.1.1466.115.121.1.5
SINGLE-VALUE USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2438
DBNAME('slapdSSLKeyDPW' 'slapdSSLKeyDPW')
ACCESS-CLASS normal)
```

```
attributetypes=(1.3.18.0.2.4.2392
NAME 'ibm-slapdSslKeyRingFile'
DESC 'file path to the LDAP servers SSL key database file. This key
database file is used for handling SSL connections from LDAP
clients, as well as for creating secure SSL connections to replica
LDAP servers. On Windows, forward slashes are allowed, and a
leading slash not preceded by a drive specifier (D:) is assumed to
be rooted at the install directory (i.e.: /etc/key.kdb =
D:\Program Files\IBM\ldap\etc\key.kdb).'
```

EQUALITY 2.5.13.5

```
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2392
DBNAME('SslKeyRingFile' 'SslKeyRingFile')
ACCESS-CLASS critical
LENGTH 1024)
```

```
attributetypes=(1.3.18.0.2.4.2390
NAME 'ibm-slapdSslKeyRingFilePW'
DESC 'Specify the password associated with the LDAP servers SSL key
database file, as specified on the ibm-slapdSslKeyRingFile
parameter. If the LDAP servers key database file has an associated
password stash file, then the ibm-slapdSslKeyRingFilePW parameter
can be omitted, or set to ibm-slapdSslKeyRingFilePW = none.
```

Note:

The password stash file must be located in the same directory as the key database file and it must have the same file name as the key database file, but with an extension of .sth, instead of .kdb.'

```
SYNTAX 1.3.6.1.4.1.1466.115.121.1.5
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2390
DBNAME('SslKeyRingFilePW' 'SslKeyRingFilePW')
ACCESS-CLASS critical)
```

```
attributetypes=(1.3.18.0.2.4.2388
NAME 'ibm-slapdSuffix'
DESC 'Specifies a naming context to be stored in this backend.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2388
DBNAME('slapdSuffix' 'slapdSuffix')
```

```
ACCESS-CLASS critical
LENGTH 1000)
```

```
attributetypes=(1.3.18.0.2.4.2480
NAME 'ibm-slapdSupportedWebAdmVersion'
DESC 'This attribute defines the earliest version of the web
administration console that supports configuration of this server.'
EQUALITY 2.5.13.2
ORDERING 2.5.13.3
SUBSTR 2.5.13.4
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2480
DBNAME('slapdSupWebAdmVer' 'slapdSupWebAdmVer')
ACCESS-CLASS normal
LENGTH 256)
```

```
attributetypes=(1.3.18.0.2.4.2393
NAME 'ibm-slapdSysLogLevel'
DESC 'Must be one of { l | m | h }. Level at which debugging and
operation statistics are logged in ibmslapd.log file. h - high
(verbose), m - medium, l - low (terse).'
EQUALITY 2.5.13.2
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2393
DBNAME('SysLogLevel' 'SysLogLevel')
ACCESS-CLASS critical
LENGTH 1)
```

```
attributetypes=(1.3.18.0.2.4.2391
NAME 'ibm-slapdTimeLimit'
DESC 'Maximum number of number of seconds to spend on search
request, regardless of any timelimit that may have been specified
on the client request. Range = 0... If a client has passed a
limit, then the smaller value of the client value and the value
read from ibmslapd.conf will be used. If a client has not passed a
limit and has bound as admin DN, then the limit will be considered
unlimited. If the client has not passed a limit and has not bound as
admin DN, then the limit will be that which was read from
ibmslapd.conf file. 0 = unlimited.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2391
DBNAME('TimeLimit' 'TimeLimit')
ACCESS-CLASS critical
LENGTH 11)
```

```
attributetypes=(ibm-slapdStartupTraceEnabled-oid
NAME 'ibm-slapdTraceEnabled'
DESC 'Must be one of { TRUE | FALSE }. Specifies whether trace information is to be
collected at server startup'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(ibm-slapdStartupTraceEnabled-oid
ACCESS-CLASS normal
LENGTH 5)
```

```
attributetypes=(ibm-slapdTraceMessageLevel-oid
NAME 'ibm-slapdTraceMessageLevel'
DESC 'any value that would be acceptable after the command line -h option, sets
Debug message level'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
```

```

SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(ibm-slapdTraceMessageLevel-oid
ACCESS-CLASS normal
LENGTH 16)

attributetypes=(ibm-slapdTraceMessageLog-oid
NAME 'ibm-slapdTraceMessageLog'
DESC 'File path or device on ibmslapd host machine to which
LDAP C API and Debug macro messages will be written.
On Windows, forward slashes are allowed, and a leading
slash not preceded by a drive letter is assumed to be rooted at
the install directory
(i.e., /tmp/tracemsg.log = C:\Program Files\IBM\ldap\tmp\tracemsg.log).'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(ibm-slapdTraceMessageLog-oid
ACCESS-CLASS normal
LENGTH 1024)

attributetypes=(1.3.18.0.2.4.2384
NAME 'ibm-slapdTransactionEnable'
DESC 'If FALSE, globally disables transaction support; the server
will reject all StartTransaction requests with the response
LDAP_UNWILLING_TO_PERFORM.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2384
DBNAME('TransactionEnable' 'TransactionEnable')
ACCESS-CLASS critical
LENGTH 5)

attributetypes=(1.3.18.0.2.4.2499
NAME 'ibm-slapdUseProcessIdPW'
DESC 'If set to true the server will use the user login ID
associated with the ibmslapd process to connect to the database. If
set to false the server will use the ibm-slapdDbUserID and
ibm-slapdDbUserPW values to connect to the database.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2499
DBNAME('useprocidpw' 'useprocidpw')
ACCESS-CLASS normal
LENGTH 5)

attributetypes=(1.3.18.0.2.4.2436
NAME 'ibm-slapdVersion'
DESC 'IBM Slapd version Number'
EQUALITY 2.5.13.5
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2436
DBNAME('slapdVersion' 'slapdVersion')
ACCESS-CLASS normal
LENGTH 1024)

attributetypes=(1.3.18.0.2.4.2327
NAME 'ibm-supportedReplicationModels'
DESC 'Advertises in the Root DSE the OIDs of replication models
supported by the server'
EQUALITY 1.3.6.1.4.1.1466.109.114.1
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
NO-USER-MODIFICATION

```

```

USAGE dSAOperation)
IBMAattributetypes=(1.3.18.0.2.4.2327
DBNAME('supportedReplicat' 'supportedReplicat')
ACCESS-CLASS system
LENGTH 240)

attributetypes=(1.3.18.0.2.4.470
NAME 'IBMAAttributeTypes'
DESC ' '
EQUALITY 2.5.13.2
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE
USAGE directoryOperation)
IBMAattributetypes=(1.3.18.0.2.4.470
DBNAME('IBMAAttributeTypes' 'IBMAAttributeTypes')
ACCESS-CLASS normal
LENGTH 256)

attributetypes=(1.3.6.1.4.1.1466.101.120.16
NAME 'ldapSyntaxes'
DESC 'Servers MAY use this attribute to list the syntaxes which are
implemented. Each value corresponds to one syntax.'
EQUALITY 2.5.13.30
SYNTAX 1.3.6.1.4.1.1466.115.121.1.54
USAGE directoryOperation)
IBMAattributetypes=(1.3.6.1.4.1.1466.101.120.16
DBNAME('ldapSyntaxes' 'ldapSyntaxes')
ACCESS-CLASS system
LENGTH 256 EQUALITY)

attributetypes=(2.5.21.4
NAME 'matchingRules'
DESC 'This attribute is typically located in the subschema entry.'
EQUALITY 2.5.13.30
SYNTAX 1.3.6.1.4.1.1466.115.121.1.30
USAGE directoryOperation)
IBMAattributetypes=(2.5.21.4
DBNAME('matchingRules' 'matchingRules')
ACCESS-CLASS system
LENGTH 256
EQUALITY)

attributetypes=(2.5.21.8
NAME 'matchingRuleUse'
DESC 'This attribute is typically located in the subschema entry.'
EQUALITY 2.5.13.30
SYNTAX 1.3.6.1.4.1.1466.115.121.1.31
USAGE directoryOperation)
IBMAattributetypes=(2.5.21.8
DBNAME('matchingRuleUse' 'matchingRuleUse')
ACCESS-CLASS system
LENGTH 256
EQUALITY)

attributetypes=(2.5.4.31
NAME 'member'
DESC 'Identifies the distinguished names for each member of the group.'
SUP 2.5.4.49
EQUALITY 2.5.13.1
USAGE userApplications)
IBMAattributetypes=(2.5.4.31
DBNAME('member' 'member')
ACCESS-CLASS normal
LENGTH 1000
EQUALITY)

attributetypes=(2.5.18.4
NAME 'modifiersName'

```

```

DESC 'Contains the last modifier of a directory entry.'
EQUALITY 2.5.13.1 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
SINGLE-VALUE
NO-USER-MODIFICATION
USAGE directoryOperation)
IBMAttributetypes=(2.5.18.4
DBNAME('ldap_entry' 'modifier')
ACCESS-CLASS system
LENGTH 1000
EQUALITY)

attributetypes=(2.5.18.2
NAME 'modifyTimestamp'
DESC 'Contains the time of the last modification of the directory
entry.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.24
SINGLE-VALUE
NO-USER-MODIFICATION
USAGE directoryOperation)
IBMAttributetypes=(2.5.18.2
DBNAME('ldap_entry' 'modify_Timestamp')
ACCESS-CLASS system
LENGTH 26)

attributetypes=(2.5.4.41
NAME 'name' DESC 'The name attribute type
is the attribute supertype from which string attribute types
typically used for naming may be formed. It is unlikely that values
of this type itself will occur in an entry.'
EQUALITY 1.3.6.1.4.1.1466.109.114.2
SUBSTR 2.5.13.4
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
USAGE userApplications)
IBMAttributetypes=(2.5.4.41
DBNAME('name' 'name')
ACCESS-CLASS normal
LENGTH 32700
EQUALITY
SUBSTR)

attributetypes=(2.5.21.7
NAME 'nameForms'
DESC 'This attribute is typically located in the subschema entry.'
EQUALITY 2.5.13.30
SYNTAX 1.3.6.1.4.1.1466.115.121.1.35
USAGE directoryOperation)
IBMAttributetypes=(2.5.21.7
DBNAME('nameForms' 'nameForms')
ACCESS-CLASS normal
LENGTH 256
EQUALITY)

attributetypes=(1.3.6.1.4.1.1466.101.120.5
NAME 'namingContexts'
DESC 'The values of this attribute correspond to naming contexts
which this server masters or shadows.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
USAGE dSAOperation)
IBMAttributetypes=(1.3.6.1.4.1.1466.101.120.5
DBNAME('namingContexts' 'namingContexts')
ACCESS-CLASS normal
LENGTH 1000)

attributetypes=(2.16.840.1.113730.3.1.11
NAME 'newSuperior'
DESC 'Specifies the name of the entry that will become the
immediate superior of the existing entry, when processing a modDN

```



```

operation.'
EQUALITY 2.5.13.1
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
SINGLE-VALUE
NO-USER-MODIFICATION
USAGE userApplications)
IBMAttributetypes=(2.16.840.1.113730.3.1.11
DBNAME('newSuperior' 'newSuperior')
ACCESS-CLASS normal
LENGTH 1000
EQUALITY APPROX)

attributetypes=(1.3.1.1.4.1.453.16.2.103
NAME 'numSubordinates'
DESC 'Counts the number of children of this entry.'
EQUALITY 2.5.13.14
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
NO-USER-MODIFICATION
USAGE directoryOperation)
IBMAttributetypes=(1.3.1.1.4.1.453.16.2.103
DBNAME('numSubordinates' 'numSubordinates')
ACCESS-CLASS system
LENGTH 11

attributetypes=(2.5.4.10
NAME ('o' 'organizationName' 'organization')
DESC 'This attribute contains the name of an organization (organizationName).'
SUP 2.5.4.41
EQUALITY 1.3.6.1.4.1.1466.109.114.2
SUBSTR 2.5.13.4
USAGE userApplications)
IBMAttributetypes=(2.5.4.10
DBNAME('o' 'o')
ACCESS-CLASS normal
LENGTH 128)

attributetypes=(2.5.4.0
NAME 'objectClass'
DESC 'The values of the objectClass attribute describe the kind of
object which an entry represents.'
EQUALITY 2.5.13.0
SYNTAX 1.3.6.1.4.1.1466.115.121.1.38
USAGE userApplications)
IBMAttributetypes=(2.5.4.0
DBNAME('objectClass' 'objectClass')
ACCESS-CLASS normal
LENGTH 128
EQUALITY)

attributetypes=(2.5.21.6
NAME 'objectClasses'
DESC 'This attribute is typically located in the subschema entry.'
EQUALITY 2.5.13.30
SYNTAX 1.3.6.1.4.1.1466.115.121.1.37
USAGE directoryOperation)
IBMAttributetypes=(2.5.21.6
DBNAME('objectClasses' 'objectClasses')
ACCESS-CLASS system
LENGTH 256
EQUALITY)

attributetypes=(1.3.18.0.2.4.289
NAME 'ownerPropagate'
DESC 'Indicates whether the entryOwner applies on entry or subtree.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
USAGE directoryOperation)

```

```

IBMAttributetypes=(1.3.18.0.2.4.289
DBNAME('ownerPropagate' 'ownerPropagate')
ACCESS-CLASS restricted
LENGTH 5)

attributetypes=(2.5.4.11
NAME ('ou' 'organizationalUnit' 'organizationalUnitName')
DESC 'This attribute contains the name of an organization (organizationName).'
SUP 2.5.4.41
EQUALITY 1.3.6.1.4.1.1466.109.114.2
SUBSTR 2.5.13.4
USAGE userApplications)
IBMAttributetypes=(2.5.4.11
DBNAME('ou' 'ou')
ACCESS-CLASS normal
LENGTH 128)

attributetypes=(2.5.4.32
NAME 'owner'
DESC 'Identifies the distinguished name (DN) of the person responsible
for the entry.'
SUP 2.5.4.49
EQUALITY 2.5.13.1
USAGE userApplications)
IBMAttributetypes=(2.5.4.32
DBNAME('owner' 'owner')
ACCESS-CLASS normal
LENGTH 1000)

attributetypes=(1.3.18.0.2.4.290
NAME 'ownerSource'
DESC 'Indicates the distinguished name of the entry whose entryOwner
value is being applied to the entry.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.290
DBNAME('ownerSource' 'ownerSource')
ACCESS-CLASS system
LENGTH 1000)

attributetypes=(1.3.6.1.4.1.42.2.27.8.1.17
NAME 'pwdAccountLockedTime'
DESC 'Specifies the time that the users account was locked'
EQUALITY 2.5.13.27
ORDERING 2.5.13.28
SYNTAX 1.3.6.1.4.1.1466.115.121.1.24
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.6.1.4.1.42.2.27.8.1.17
DBNAME('pwdAccLockTime' 'pwdAccLockTime')
ACCESS-CLASS critical
LENGTH 30)

attributetypes=(1.3.6.1.4.1.42.2.27.8.1.16
NAME 'pwdChangedTime'
DESC 'Specifies the last time the entrys password was changed'
EQUALITY 2.5.13.27
ORDERING 2.5.13.28
SYNTAX 1.3.6.1.4.1.1466.115.121.1.24
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.6.1.4.1.42.2.27.8.1.16
DBNAME('pwdChangedTime' 'pwdChangedTime')
ACCESS-CLASS critical
LENGTH 30)

```

```

attributetypes=(1.3.6.1.4.1.42.2.27.8.1.18
NAME 'pwdExpirationWarned'
DESC 'The time the user was first warned about the coming expiration
of the password'
EQUALITY 2.5.13.27
ORDERING 2.5.13.28
SYNTAX 1.3.6.1.4.1.1466.115.121.1.24
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.6.1.4.1.42.2.27.8.1.18
DBNAME('pwdExpireWarned' 'pwdExpireWarned')
ACCESS-CLASS critical
LENGTH 30)

```

```

attributetypes=(1.3.6.1.4.1.42.2.27.8.1.19
NAME 'pwdFailureTime'
DESC 'The timestamps of the last consecutive authentication
failures'
EQUALITY 2.5.13.27
ORDERING 2.5.13.28
SYNTAX 1.3.6.1.4.1.1466.115.121.1.24
USAGE directoryOperation)
IBMAttributetypes=(1.3.6.1.4.1.42.2.27.8.1.19
DBNAME('pwdFailureTime' 'pwdFailureTime')
ACCESS-CLASS critical
LENGTH 30)

```

```

attributetypes=(1.3.6.1.4.1.42.2.27.8.1.21
NAME 'pwdGraceUseTime'
DESC 'The timestamps of the grace login once the password has
expired'
EQUALITY 2.5.13.27
SYNTAX 1.3.6.1.4.1.1466.115.121.1.24
USAGE directoryOperation)
IBMAttributetypes=(1.3.6.1.4.1.42.2.27.8.1.21
DBNAME('pwdGraceUseTime' 'pwdGraceUseTime')
ACCESS-CLASS critical
LENGTH 30)

```

```

attributetypes=(1.3.6.1.4.1.42.2.27.8.1.20
NAME 'pwdHistory'
DESC 'The history of users passwords'
EQUALITY 2.5.13.5
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
USAGE directoryOperation)
IBMAttributetypes=(1.3.6.1.4.1.42.2.27.8.1.20
DBNAME('pwdHistory' 'pwdHistory')
ACCESS-CLASS critical
LENGTH 1024)

```

```

attributetypes=(1.3.6.1.4.1.42.2.27.8.1.22
NAME 'pwdReset'
DESC 'Indicates that the password has been reset.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.6.1.4.1.42.2.27.8.1.22
DBNAME('pwdReset' 'pwdReset')
ACCESS-CLASS critical
LENGTH 5)

```

```

attributetypes=(1.3.18.0.2.4.299
NAME 'replicaBindDN'
DESC 'Distinguished name to use on LDAP bind to the remote replica'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.299

```

```

DBNAME('replicaBindDN' 'replicaBindDN')
ACCESS-CLASS critical
LENGTH 1000)

attributetypes=(1.3.18.0.2.4.302
NAME 'replicaBindMethod'
DESC 'LDAP bind type to use on LDAP bind to replica.'
EQUALITY 2.5.13.2
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.302
DBNAME('replicaBindMethod' 'replicaBindMethod')
ACCESS-CLASS normal
LENGTH 100)

attributetypes=(1.3.18.0.2.4.300
NAME ('replicaCredentials' 'replicaBindCredentials')
DESC 'Credentials to use on LDAP bind to the remote replica'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.5
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.300
DBNAME('replicaCred' 'replicaCred')
ACCESS-CLASS critical)

attributetypes=(1.3.18.0.2.4.298
NAME 'replicaHost'
DESC 'Hostname of the remote replica'
EQUALITY 2.5.13.2
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.298
DBNAME('replicaHost' 'replicaHost')
ACCESS-CLASS normal
LENGTH 100)

attributetypes=(1.3.18.0.2.4.301
NAME 'replicaPort'
DESC 'TCP/IP port that the replica server is listening on.'
EQUALITY 2.5.13.2
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.301
DBNAME('replicaPort' 'replicaPort')
ACCESS-CLASS normal
LENGTH 10)

attributetypes=(1.3.18.0.2.4.304
NAME 'replicaUpdateTimeInterval'
DESC 'Specifies the time between replica update transmissions from
master to slave replica.'
EQUALITY 2.5.13.2
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.304
DBNAME('replicaUpdateInt' 'replicaUpdateInt')
ACCESS-CLASS normal
LENGTH 20)

attributetypes=(1.3.18.0.2.4.303
NAME 'replicaUseSSL'
DESC 'Signifies whether replication flows should be protected using
SSL communications.'
EQUALITY 2.5.13.2

```

```

SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.303
DBNAME('replicaUseSSL' 'replicaUseSSL')
ACCESS-CLASS normal
LENGTH 10)

attributetypes=(2.16.840.1.113730.3.1.34
NAME 'ref'
DESC 'standard Attribute'
EQUALITY 2.5.13.5
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
USAGE userApplications)
IBMAttributetypes=(2.16.840.1.113730.3.1.34
DBNAME('ref' 'ref')
ACCESS-CLASS normal
LENGTH 100)

attributetypes=(2.5.4.34
NAME 'seeAlso'
DESC 'Identifies another directory server entry that may contain information
related to this entry.'
SUP 2.5.4.49
EQUALITY 2.5.13.1
USAGE userApplications)
IBMAttributetypes=(2.5.4.34
DBNAME('seeAlso' 'seeAlso')
ACCESS-CLASS normal
LENGTH 1000)

attributetypes=(2.5.18.10
NAME 'subschemaSubentry'
DESC 'The value of this attribute is the name of a subschema entry
in which the server makes available attributes specifying the
schema.'
EQUALITY 2.5.13.1
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 SINGLE-VALUE
NO-USER-MODIFICATION
USAGE directoryOperation)
IBMAttributetypes=(2.5.18.10
DBNAME('subschemaSubent' 'subschemaSubent')
ACCESS-CLASS system
LENGTH 1000
EQUALITY)

attributetypes=(1.3.18.0.2.4.819
NAME 'subtreeSpecification'
DESC 'Identifies a collection of entries that are located at the
vertices of a single subtree.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
NO-USER-MODIFICATION
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.819
DBNAME('subtreeSpec' 'subtreeSpec')
ACCESS-CLASS system
LENGTH 2024)

attributetypes=(1.3.6.1.4.1.1466.101.120.7
NAME 'supportedExtension'
DESC 'The values of this attribute are OBJECT IDENTIFIERS
identifying the supported extended operations which the server
supports.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.38
USAGE dSAOperation)
IBMAttributetypes=(1.3.6.1.4.1.1466.101.120.7

```

```
DBNAME('supportedExtensio' 'supportedExtensio')
ACCESS-CLASS normal
LENGTH 256)
```

```
attributetypes=(1.3.6.1.4.1.1466.101.120.15
NAME 'supportedLDAPVersion'
DESC 'The values of this attribute are the versions of the LDAP
protocol which the server implements.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
USAGE dSAOperation)
IBMAttributetypes=(1.3.6.1.4.1.1466.101.120.15
DBNAME('supportedLDAPVers' 'supportedLDAPVers')
ACCESS-CLASS normal
LENGTH 11)
```

```
attributetypes=(1.3.6.1.4.1.1466.101.120.14
NAME 'supportedSASLMechanisms'
DESC 'The values of this attribute are the names of supported SASL
mechanisms which the server supports.'
EQUALITY 2.5.13.2
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
USAGE dSAOperation)
IBMAttributetypes=(1.3.6.1.4.1.1466.101.120.14
DBNAME('supportedSASLMech' 'supportedSASLMech')
ACCESS-CLASS normal LENGTH 2048)
```

```
attributetypes=(2.16.840.1.113730.3.1.6
NAME 'targetDN'
DESC 'Defines the distinguished name of an entry that was added,
modified, or deleted on a supplier server. In the case of a modrdn
operation, the targetDn contains the distinguished name of the
entry before it was modified.'
EQUALITY 2.5.13.1
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
SINGLE-VALUE
NO-USER-MODIFICATION
USAGE userApplications)
IBMAttributetypes=(2.16.840.1.113730.3.1.6
DBNAME('targetDN' 'targetDN')
ACCESS-CLASS normal
LENGTH 1000
EQUALITY APPROX)
```

---

## Appendix I. 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. 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 441. 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. 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 J. Filtered ACLs and non-filtered ACLs – sample LDIF file

To have a complete understanding of the ACL models, an administrator can best learn through hands on trial. Create sample data with sample ACLs for your directory and check the effective ACLs of each of the entries to ensure that the ACL scheme is correct for the desired access.

Included is a sample LDIF file that contains combinations of filtered ACLs and non-filtered ACLs. This sample LDIF file can be loaded onto a directory server.

In this sample LDIF file, there is one suffix entry, two user entries and 17 additional entries spread over 5 levels of the directory tree. Each entry has a two-digit designation. The first digit identifies the level where the entry is in the directory tree. The entries are also numbered on each level, incrementally, from left to right. This numbering format is reflected in the second digit.

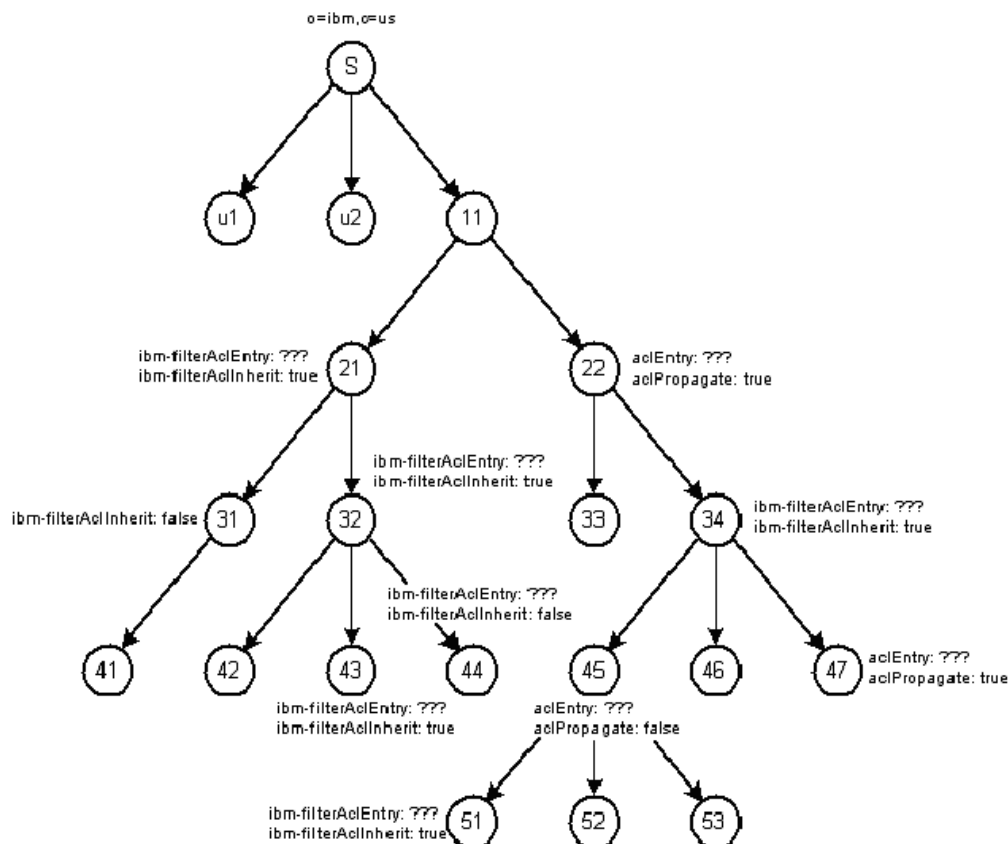


Figure 21. Filtered ACLs and non-filtered ACLs

LDIF File:

version: 1

dn: o=IBM, c=US  
objectclass: organization

```

objectclass: top
o: IBM

dn: cn=User1, o=IBM, c=US
cn: User1
sn: User
objectclass: person
objectclass: top
userPassword: User1

dn: o=Level11, o=IBM, c=US
o: Level11
objectclass: organization
objectclass: top

dn: o=Level21, o=Level11, o=IBM, c=US
o: Level21
objectclass: organization
objectclass: top
ibm-filterAclEntry: access-id:CN=USER1,O=IBM,C=US:(o=Level32):normal:rWSC:
sensitive:rsc:critical:rsc

dn: o=Level31, o=Level21, o=Level11, o=IBM, c=US
o: Level31
objectclass: organization
objectclass: top
ibm-filterAclInherit: FALSE

dn: o=Level41, o=Level31, o=Level21, o=Level11, o=IBM, c=US
o: Level41
objectclass: organization
objectclass: top

dn: o=Level32, o=Level21, o=Level11, o=IBM, c=US
o: Level32
objectclass: organization
objectclass: top
ibm-filterAclEntry: access-id:CN=USER1,O=IBM,C=US:(o=Level42):normal:rWSC:
sensitive:rsc:critical:rsc
ibm-filterAclEntry: access-id:CN=USER1,O=IBM,C=US:(o=Level43):normal:rWSC:
sensitive:rsc:critical:rsc
ibm-filterAclEntry: access-id:CN=USER2,O=IBM,C=US:(o=Level44):normal:rWSC:
sensitive:rsc:critical:rsc

dn: o=Level42, o=Level32, o=Level21, o=Level11, o=IBM, c=US
o: Level42
objectclass: organization
objectclass: top

dn: o=Level43, o=Level32, o=Level21, o=Level11, o=IBM, c=US
o: Level43
objectclass: organization
objectclass: top
ibm-filterAclEntry: access-id:CN=USER1,O=IBM,C=US:(o=Level43):normal:rWSC:
sensitive:rsc:critical:rsc

dn: o=Level44, o=Level32, o=Level21, o=Level11, o=IBM, c=US
o: Level44
objectclass: organization
objectclass: top
ibm-filterAclEntry: access-id:CN=USER1,O=IBM,C=US:(o=Level44):normal:rWSC:
sensitive:rsc:critical:rsc
ibm-filterAclInherit: FALSE

dn: cn=User2, o=IBM, c=US
cn: User2
sn: User

```

```

objectclass: person
objectclass: top
userPassword: User2

dn: o=Level22, o=Level11, o=IBM, c=US
o: Level22
objectclass: organization
objectclass: top
aclentry: access-id:CN=USER2,O=IBM,C=US:normal:rsc:at.sn:deny:c:sensitive:
c:critical:c

dn: o=Level33, o=Level22, o=Level11, o=IBM, c=US
o: Level33
objectclass: organization
objectclass: top

dn: o=Level34, o=Level22, o=Level11, o=IBM, c=US
o: Level34
objectclass: organization
objectclass: top
ibm-filterAclEntry: access-id:CN=USER2,O=IBM,C=US:(o=Level34):normal:rwc:
sensitive:rsc:critical:rsc
ibm-filterAclEntry: access-id:CN=USER2,O=IBM,C=US:(o=Level51):normal:rwc:
sensitive:rwc:critical:rsc
ibm-filterAclEntry: access-id:CN=USER1,O=IBM,C=US:(o=Level53):normal:rwc:
sensitive:rsc:critical:rsc
ibm-filterAclEntry: access-id:CN=USER2,O=IBM,C=US:(o=Level46):normal:rwc:
sensitive:rsc:critical:rsc

dn: o=Level45, o=Level34, o=Level22, o=Level11, o=IBM, c=US
o: Level45
objectclass: organization
objectclass: top
aclentry: access-id:CN=USER2,O=IBM,C=US:normal:rwc:sensitive:rsc:critical
:rsc
aclpropagate: FALSE

dn: o=Level51, o=Level45, o=Level34, o=Level22, o=Level11, o=IBM, c=US
o: Level51
objectclass: organization
objectclass: top
ibm-filterAclEntry: access-id:CN=USER2,O=IBM,C=US:(o=Level51):normal:rwc:
sensitive:rsc:critical:rsc

dn: o=Level52, o=Level45, o=Level34, o=Level22, o=Level11, o=IBM, c=US
o: Level52
objectclass: organization
objectclass: top

dn: o=Level53, o=Level45, o=Level34, o=Level22, o=Level11, o=IBM, c=US
o: Level53
objectclass: organization
objectclass: top

dn: o=Level46, o=Level34, o=Level22, o=Level11, o=IBM, c=US
o: Level46
objectclass: organization
objectclass: top

dn: o=Level47, o=Level34, o=Level22, o=Level11, o=IBM, c=US
o: Level47
objectclass: organization
objectclass: top
aclentry: access-id:CN=USER2,O=IBM,C=US:normal:rwc:sensitive:rsc:critical
:rsc

```

The following is a sample search output with comments about how the ACL was calculated for that entry:

```
>idsldapsearch -D <admin DN> -w <admin PW> -b o=ibm,c=us objectclass=*
 ibm-effectiveACL ibm-filterAcEntry
 ibm-filterACLInherit acEntry acPropagate
```

```
o=IBM,c=US
acPropagate=TRUE
acEntry=group:CN=ANYBODY:system:rsc:normal:rsc:restricted:rsc
ibm-effectiveACL=group:CN=ANYBODY:restricted:rsc:normal:rsc:system:rsc
```

The effective ACL for this entry is the default ACL because the following are true:

- There is no explicit non-filtered ACL defined on this entry.
- There are no propagating non-filtered ACLs defined higher in the directory tree.
- None of the defined filtered ACLs apply to this entry.

```
cn=User1,o=IBM,c=US
acPropagate=TRUE
acEntry=group:CN=ANYBODY:system:rsc:normal:rsc:restricted:rsc
ibm-effectiveACL=group:CN=ANYBODY:restricted:rsc:normal:rsc:system:rsc
```

The effective ACL for this entry is the default ACL because the following are true:

- There is no explicit non-filtered ACL defined on this entry.
- There are no propagating non-filtered ACLs defined higher in the directory tree.
- None of the defined filtered ACLs apply to this entry.

```
o=Level11,o=IBM,c=US
acPropagate=TRUE
acEntry=group:CN=ANYBODY:system:rsc:normal:rsc:restricted:rsc
ibm-effectiveACL=group:CN=ANYBODY:restricted:rsc:normal:rsc:system:rsc
```

The effective ACL for this entry is the default ACL because the following are true:

- There is no explicit non-filtered ACL defined on this entry.
- There are no propagating non-filtered ACLs defined higher in the directory tree.
- None of the defined filtered ACLs apply to this entry.

```
o=Level21,o=Level11,o=IBM,c=US
ibm-filterACLInherit=TRUE
ibm-filterAcEntry=access-id:CN=USER1,o=IBM,C=US:(o=Level32):normal:rws:
sensitive:rsc:critical:rsc
ibm-effectiveACL=group:CN=ANYBODY:restricted:rsc:system:rsc:normal:rsc
```

This entry has a filtered ACL defined in it that does not apply to the entry. The filtered ACL defined in this entry only applies to an entry that has o=Level32. The effective ACL for this entry is the default ACL because the following are true:

- There is no explicit non-filtered ACL defined on this entry.
- There are no propagating non-filtered ACLs defined higher in the directory tree.
- None of the defined filtered ACLs apply to this entry.

```
o=Level31,o=Level21,o=Level11,o=IBM,c=US
ibm-filterACLInherit=FALSE
ibm-effectiveACL=group:CN=ANYBODY:restricted:rsc:system:rsc:normal:rsc
```

This entry has an `ibm-filterACLInherit=FALSE` defined on it. This attribute acts as a ceiling and stops the accumulation of filtered ACLs. In this case, there are no filtered ACLs defined below this entry. The effective ACL for this entry is the default ACL because the following are true:

- The `ibm-filterACLInherit` definition causes this entry to be in filter ACL mode, and therefore excludes non-filter ACL definitions.

- None of the defined filtered ACLs apply to this entry.

```
o=Level141,o=Level131,o=Level121,o=Level111,o=IBM,c=US
ibm-effectiveACL=group:CN=ANYBODY:restricted:rsc:system:rsc:normal:rsc
```

The effective ACL for this entry is the default ACL because the following are true:

- There is no explicit non-filtered ACL defined on this entry.
- There are no propagating non-filtered ACLs defined higher in the directory tree.
- None of the defined filtered ACLs apply to this entry.

```
o=Level132,o=Level121,o=Level111,o=IBM,c=US
ibm-filterACLInherit=TRUE
ibm-filterAclEntry=access-id:CN=USER2,0=IBM,C=US:(o=Level144):normal:rsc:
sensitive:rsc:critical:rsc
ibm-filterAclEntry=access-id:CN=USER1,0=IBM,C=US:(o=Level143):normal:rsc:
sensitive:rsc:critical:rsc
ibm-filterAclEntry=access-id:CN=USER1,0=IBM,C=US:(o=Level142):normal:rsc:
sensitive:rsc:critical:rsc
ibm-effectiveACL=access-id:CN=USER1,0=IBM,C=US:normal:rsc:sensitive:rsc:
critical:rsc
```

The attribute `ibm-filterACLInherit=TRUE` means that this entry does not act as a ceiling for any filtered ACLs.

The three `ibm-filterAclEntry` attributes provide an example of how a filtered ACL can be defined on one entry and apply to another entry. In this case the three filtered ACLs apply to the three children of this entry but not to this entry. The effective ACL was calculated by an accumulation of all the filtered ACLs which applied to this entry. There was only one filtered ACL that applied to this entry, which is the filtered ACL defined on the `o=Level121,o=Level111,o=IBM,c=US` entry. No other filtered ACLs apply to this entry, so the effective ACL is taken directly from the filtered ACL defined on the `o=Level121,o=Level111,o=IBM,c=US` entry.

```
o=Level142,o=Level132,o=Level121,o=Level111,o=IBM,c=US
ibm-effectiveACL=access-id:CN=USER1,0=IBM,C=US:normal:rsc:sensitive:rsc:
critical:rsc
```

The filtered ACL defined on the `o=Level132,o=Level121,o=Level111,o=IBM,c=US` entry is used to calculate the effective ACL for this entry.

```
o=Level143,o=Level132,o=Level121,o=Level111,o=IBM,c=US
ibm-filterACLInherit=TRUE
ibm-filterAclEntry=access-id:CN=USER1,0=IBM,C=US:(o=Level143):normal:rsc:
sensitive:rsc:critical:rsc
ibm-effectiveACL=access-id:CN=USER1,0=IBM,C=US:normal:rsc:sensitive:rsc:
critical:rsc
```

This entry is a simple example of how filtered ACLs accumulate. The filtered ACL defined on the `o=Level132,o=Level121,o=Level111,o=IBM,c=US` entry is combined with the filtered ACL defined on the

`o=Level143,o=Level132,o=Level121,o=Level111,o=IBM,c=US` entry to give read, write, search and compare access to all three classes of attributes for user 1.

```
o=Level144,o=Level132,o=Level121,o=Level111,o=IBM,c=US
ibm-filterACLInherit=FALSE
ibm-filterAclEntry=access-id:CN=USER1,0=IBM,C=US:(o=Level144):normal:rsc:
sensitive:rsc:critical:rsc
ibm-effectiveACL=access-id:CN=USER1,0=IBM,C=US:normal:rsc:sensitive:rsc:
critical:rsc
```

This entry is a simple example of how the `ibm-filterACLInherit` attribute can be used to stop the accumulation of filtered ACLs. The filtered ACL defined on the `o=Level132,o=Level121,o=Level111,o=IBM,c=US` entry does not apply to this entry

because `ibm-filterACLInherit=FALSE`. Only the filtered ACL defined on the `o=Level44,o=Level32,o=Level21,o=Level11,o=IBM,c=US` entry applies to give access to user 1. If the `ibm-filterACLInherit` value is changed to `TRUE`, the effective ACL gives access to both user 2 and user 1, and looks like the following:

```
ibm-effectiveACL=access-id:CN=USER2,0=IBM,C=US:normal:rws: sensitive:rsc:
critical:rsc
ibm-effectiveACL=access-id:CN=USER1,0=IBM,C=US:normal:rws: sensitive:rsc:
critical:rsc
cn=User2,o=IBM,c=US
aclPropagate=TRUE
aclEntry=group:CN=ANYBODY:system:rsc:normal:rsc:restricted:rsc
ibm-effectiveACL=group:CN=ANYBODY:restricted:rsc:normal:rsc:system:rsc
```

The effective ACL for this entry is the default ACL because the following are true:

- There is no explicit non-filtered ACL defined on this entry.
- There are no propagating non-filtered ACLs defined higher in the directory tree.
- None of the defined filtered ACLs apply to this entry.

```
o=Level22,o=Level11,o=IBM,c=US
aclPropagate=TRUE
aclEntry=access-id:CN=USER2,0=IBM,C=US:sensitive:c:at.sn:deny:c:normal:
rsc:critical:c
ibm-effectiveACL=access-id:CN=USER2,0=IBM,C=US:critical:c:normal:rsc:
at.sn:deny:c:sensitive:c
```

This is an example of non-filtered ACLs. The effective ACL for this entry is the ACL defined in the entry.

**Note:** The value returned in the effective ACL is the server's normalized value.

```
o=Level33,o=Level22,o=Level11,o=IBM,c=US
aclPropagate=TRUE
aclEntry=access-id:CN=USER2,0=IBM,C=US:sensitive:c:at.sn:deny:c:normal:
rsc:critical:c
ibm-effectiveACL=access-id:CN=USER2,0=IBM,C=US:critical:c:normal:rsc:
at.sn:deny:c:sensitive:c
```

This is an example of the non-filtered ACL defined on the `o=Level22,o=Level11,o=IBM,c=US` entry propagating down to the `o=Level33,o=Level22,o=Level11,o=IBM,c=US` entry. This propagation occurs because the `aclPropagate` attribute was set to `TRUE` in the `o=Level22,o=Level11,o=IBM,c=US` entry.

```
o=Level34,o=Level22,o=Level11,o=IBM,c=US
ibm-filterACLInherit=TRUE
ibm-filterAclEntry=access-id:CN=USER2,0=IBM,C=US:(o=Level46):normal:rws:
sensitive:rsc:critical:rsc
ibm-filterAclEntry=access-id:CN=USER1,0=IBM,C=US:(o=Level53):normal:rws:
sensitive:rsc:critical:rsc
ibm-filterAclEntry=access-id:CN=USER2,0=IBM,C=US:(o=Level51):normal:rws:
sensitive:rws:critical:rsc
ibm-filterAclEntry=access-id:CN=USER2,0=IBM,C=US:(o=Level34):normal:rws:
sensitive:rsc:critical:rsc
ibm-effectiveACL=access-id:CN=USER2,0=IBM,C=US:normal:rws: sensitive:rsc:
critical:rsc
```

This entry has 4 filtered ACLs defined in it. One of the filtered ACLs applies to this entry. The effective ACL is a result of this filtered ACL.

**Note:** The non-filter ACL defined on the o=Level22,o=Level11,o=IBM,c=US entry did not propagate to this entry. The non-filtered ACL did not propagate to this entry because filtered ACLs are defined on this entry, and only one kind of ACL can exist on a given entry.

```
o=Level145,o=Level134,o=Level122,o=Level11,o=IBM,c=US
aclPropagate=FALSE
aclEntry=access-id:CN=USER2,0=IBM,C=US:sensitive:rsc:normal:rwc:critical:
 rsc
ibm-effectiveACL=access-id:CN=USER2,0=IBM,C=US:critical:rsc:normal:rwc:
 sensitive:rsc
```

This entry has an explicit non-filtered ACL defined, and the effective ACL is taken from the explicitly defined ACL. Because aclPropagate is FALSE, the defined non-filtered ACL does not propagate down the tree.

```
o=Level151,o=Level145,o=Level134,o=Level122,o=Level11,o=IBM,c=US
ibm-filterACLInherit=TRUE
ibm-filterACLEntry=access-id:CN=USER2,0=IBM,C=US:(o=Level151):normal:rwc:
 sensitive:rsc:critical:rsc
ibm-effectiveACL=access-id:CN=USER2,0=IBM,C=US:normal:rwc:sensitive:rwc:
 critical:rsc
```

This entry is an example of how filtered ACLs can accumulate even past a non-filtered ACL entry. The effective ACL for the entry is a combination of the filtered ACL defined on the o=Level134,o=Level122,o=Level11,o=IBM,c=US entry and the o=Level151,o=Level145,o=Level134,o=Level122,o=Level11,o=IBM,c=US entry.

```
o=Level152,o=Level145,o=Level134,o=Level122,o=Level11,o=IBM,c=US
ibm-effectiveACL=group:CN=ANYBODY:restricted:rsc:system:rsc:normal:rsc
```

The effective ACL for this entry is the default ACL. Because the entry does not have any explicit ACL attributes to set the mode to either filtered or not filtered, you must look up the directory tree for the ACL source. The Level145 entry has non-filtered ACLs, but has aclPropagate set to FALSE, so it is not the ACL source. Then, we go to the next ancestor in the directory tree, the Level 34 entry. The Level 34 entry is of the filter ACL type. The Level 34 entry is the ACL source for the entry. Since there are no filtered ACLs in the tree that apply to the entry, the default ACL is applied.

```
o=Level153,o=Level145,o=Level134,o=Level122,o=Level11,o=IBM,c=US
ibm-effectiveACL=access-id:CN=USER1,0=IBM,C=US:normal:rwc:sensitive:rsc:
 critical:rsc
```

The effective ACL for this entry is the filtered ACL defined in the o=Level134,o=Level122,o=Level11,o=IBM,c=US entry.

```
o=Level146,o=Level134,o=Level122,o=Level11,o=IBM,c=US
ibm-effectiveACL=access-id:CN=USER2,0=IBM,C=US:normal:rwc:sensitive:rsc:
 critical:rsc
```

The effective ACL for this entry is the propagated non-filtered ACL defined on the o=Level134,o=Level122,o=Level11,o=IBM,c=US entry.

```
o=Level147,o=Level134,o=Level122,o=Level11,o=IBM,c=US
aclPropagate=TRUE
aclEntry=access-id:CN=USER2,0=IBM,C=US:sensitive:rsc:normal:rwc:critical:
 rsc
ibm-effectiveACL=access-id:CN=USER2,0=IBM,C=US:critical:rsc:normal:rwc:
 sensitive:rsc
```

This entry has an explicit non-filtered ACL defined, so the effective ACL is taken from the explicitly defined ACL.





---

## Appendix K. IBM Tivoli Directory Server 6.0 configuration schema object classes and attributes

These are the configuration object classes and attributes that are included in the IBM Tivoli Directory Server Version 6.0. They can be found in the **V3config.oc** and **V3.config.at** files in the **etc** directory. They define the objects that can appear in the **ibmslapd.conf** file.

---

### Configuration object classes

These are the schema object classes that are shipped with the IBM Tivoli Directory Server Version 6.0

```
File generated at 8:27:12 AM on 8/18/2004 from IBM LDAP schema version 1.5
```

```
objectclasses=(1.3.18.0.2.6.489
NAME 'ibm-slapdAdmin'
DESC 'Global configuration settings for IBM Admin Daemon'
SUP (ibm-slapdConfigEntry $ top)
STRUCTURAL
MUST (cn $ ibm-slapdPort)
MAY (ibm-slapdSecurePort))
```

```
objectclasses=(1.3.18.0.2.6.556
NAME 'ibm-slapdAdminGroupMember'
DESC 'A User belonging to the IBM Directory Server Administration Group.'
SUP (top $ ibm-slapdConfigEntry)
STRUCTURAL
MUST (ibm-slapdAdminDN $ ibm-slapdAdminPW)
MAY (ibm-slapdKrbAdminDN $ ibm-slapdDigestAdminUser))
```

```
objectclasses=(1.3.18.0.2.6.490
NAME 'ibm-slapdConfigBackend'
DESC 'Config backend configuration for IBM Directory'
SUP (top $ ibm-slapdConfigEntry)
STRUCTURAL
MUST (cn $ ibm-slapdPlugin $ ibm-slapdSuffix)
MAY (ibm-slapdReadOnly))
```

```
objectclasses=(1.3.18.0.2.6.486
NAME 'ibm-slapdConfigEntry'
DESC 'ibm slapd config entry'
SUP 'top'
ABSTRACT
MUST (cn)
MAY (ibm-slapdInvalidLine))
```

```
objectclasses=(1.3.18.0.2.6.560
NAME 'ibm-slapdConnectionManagement'
DESC 'Global connection settings for IBM Directory Server.'
SUP (ibm-slapdConfigEntry $ top)
STRUCTURAL
MUST (cn)
MAY (ibm-slapdAllowAnon $ ibm-slapdAllReapingThreshold
$ ibm-slapdAnonReapingThreshold $ ibm-slapdBoundReapingThreshold
$ ibm-slapdESizeThreshold $ ibm-slapdEThreadActivate
$ ibm-slapdEThreadEnable $ ibm-slapdETimeThreshold
$ ibm-slapdWriteTimeout $ ibm-slapdIdleTimeOut))
```

```
objectclasses=(1.3.18.0.2.6.493
NAME 'ibm-slapdCRL'
```

```

DESC 'Certificate revocation list settings for IBM Directory.'
SUP (top $ ibm-slapdConfigEntry)
STRUCTURAL
MUST (cn $ ibm-slapdLdapCr1Host $ ibm-slapdLdapCr1Port)
MAY (ibm-slapdLdapCr1Password $ ibm-slapdLdapCr1User))

objectclasses=(1.3.18.0.2.6.575
NAME 'ibm-slapdDigest'
DESC 'Global configuration entries for the DIGEST-MD5 SASL
bind mechanism for IBM Directory.'
SUP 'ibm-slapdConfigEntry'
STRUCTURAL
MAY (ibm-slapdDigestAdminUser $ ibm-slapdDigestAttr
$ ibm-slapdDigestRealm))

objectclasses=(1.3.18.0.2.6.500
NAME 'ibm-slapdEventNotification'
DESC 'Global event notification settings for IBM Directory.'
SUP (top $ ibm-slapdConfigEntry)
STRUCTURAL
MUST (cn $ ibm-slapdEnableEventNotification)
MAY (ibm-slapdMaxEventsPerConnection $ ibm-slapdMaxEventsTotal))

objectclasses=(1.3.18.0.2.6.501
NAME 'ibm-slapdFrontEnd'
DESC 'Global front-end settings which the server will load at startup.'
SUP (top $ ibm-slapdConfigEntry)
STRUCTURAL
MUST (cn)
MAY (ibm-slapdPlugin $ ibm-slapdSetenv
$ ibm-slapdIdleTimeOut $ ibm-slapdACLCache
$ ibm-slapdACLCacheSize $ ibm-slapdFilterCacheSize
$ ibm-slapdFilterCacheBypassLimit $ ibm-slapdEntryCacheSize
$ ibm-slapdDB2CP))

objectclasses=(1.3.18.0.2.6.494
NAME 'ibm-slapdKerberos'
DESC 'Global kerberos authentication settings for IBM Directory.'
SUP (top $ ibm-slapdConfigEntry)
STRUCTURAL
MUST (cn $ ibm-slapdKrbAdminDN $ ibm-slapdKrbEnable
$ ibm-slapdKrbIdentityMap $ ibm-slapdKrbKeyTab
$ ibm-slapdKrbRealm))

objectclasses=(1.3.18.0.2.6.495
NAME 'ibm-slapdLdcfBackend'
DESC 'LDCF backend configuration for IBM Directory.'
SUP (top $ ibm-slapdConfigEntry)
STRUCTURAL
MUST (cn)
MAY (ibm-slapdSuffix $ ibm-slapdPlugin
$ ibm-slapdReadOnly))

objectclasses=(1.3.18.0.2.6.588
NAME 'ibm-slapdLogConfig'
DESC 'Log management configuration.'
SUP (top $ ibm-slapdConfigEntry)
AUXILIARY
MAY (ibm-slapdLogMaxArchives $ ibm-slapdLogOptions
$ ibm-slapdLogSizeThreshold $ ibm-slapdLogArchivePath
$ ibm-slapdLog))

objectclasses=(1.3.18.0.2.6.526
NAME 'ibm-slapdPendingMigration'
DESC 'Indicates that a server component requires migration.'
SUP 'top'
AUXILIARY

```

```

MAY (ibm-slapdMigrationInfo))

objectclasses=(1.3.18.0.2.6.586
NAME 'ibm-slapdProxyBackend'
DESC 'Information related to loading the proxy plug-in.'
SUP (top $ ibm-slapdConfigEntry)
STRUCTURAL
MUST (ibm-slapdPlugin $ cn)
MAY (ibm-slapdSuffix))

objectclasses=(1.3.18.0.2.6.585
NAME 'ibm-slapdProxyBackendServer'
DESC 'Contains information necessary for a proxy server to connect
to a backend server'
SUP (ibm-slapdConfigEntry $ top)
STRUCTURAL
MUST (cn $ ibm-slapdProxyBindMethod $ ibm-slapdProxyTargetURL)
MAY (ibm-slapdProxyConnectionPoolSize $ ibm-slapdProxyDigestUserName
$ ibm-slapdSslCertificate $ ibm-slapdSSLKeyDatabase
$ ibm-slapdSSLKeyDatabasePW $ ibm-slapdSslKeyRingFilePW
$ ibm-slapdProxyDn $ ibm-slapdProxyPw
$ ibm-slapdProxyDigestRealm $ ibm-slapdReferral
$ ibm-slapdSuffix))

objectclasses=(1.3.18.0.2.6.594
NAME 'ibm-slapdProxyBackendSplit'
DESC 'Contains specific indexes of a split partition and defines
which server holds them.'
SUP 'top'
STRUCTURAL
MUST (ibm-slapdProxyPartitionIndex $ ibm-slapdProxyBackendServerDn))

objectclasses=(1.3.18.0.2.6.593
NAME 'ibm-slapdProxyBackendSplitContainer'
DESC 'Objectclass containing attributes that describe a split
container held by 1 or more servers'
SUP (ibm-slapdConfigEntry $ top)
STRUCTURAL
MUST (cn $ ibm-slapdProxyNumPartitions $ ibm-slapdProxyPartitionBase))

objectclasses=(1.3.18.0.2.6.592
NAME 'ibm-slapdPwdPolicyAdmin'
DESC 'Defines the global configuration for the IBM Administrative
Password Policy for IBM Directory Server'
SUP 'top'
STRUCTURAL
MUST (ibm-slapdConfigPwdPolicyOn)
MAY (pwdLockout $ pwdLockoutDuration
$ pwdAccountLockedTime $ pwdMaxFailure
$ pwdFailureCountInterval $ passwordMinAlphaChars
$ passwordMinOtherChars $ passwordMinDiffChars
$ passwordMaxRepeatedChars $ pwdMinLength))

objectclasses=(1.3.18.0.2.6.497
NAME 'ibm-slapdRdbmBackend'
DESC 'DB2 database backend configuration for IBM Directory.'
SUP (top $ ibm-slapdConfigEntry)
STRUCTURAL
MUST (cn $ ibm-slapdDbName $ ibm-slapdDbInstance
$ ibm-slapdDbUserID $ ibm-slapdDbUserPW)
MAY (ibm-slapdPlugin $ ibm-slapdSuffix
$ ibm-slapdReadOnly $ ibm-slapdChangeLogMaxEntries
$ ibm-slapdPagedResAllowNonAdmin $ ibm-slapdPagedResLmt
$ ibm-slapdSortKeyLimit $ ibm-slapdSortSrchAllowNonAdmin
$ ibm-slapdDbConnections $ ibm-slapdDbLocation
$ ibm-slapdDB2CP $ ibm-slapdReplDbConns
$ ibm-slapdCLIErrors $ ibm-slapdBulkloadErrors

```

```

$ ibm-slapdDBAlias $ ibm-slapdUseProcessIdPW
$ ibm-slapdChangeLogMaxAge $ ibm-slapdCachedAttributeSize
$ ibm-slapdCachedAttribute $ ibm-slapdLanguageTagsEnabled))

objectclasses=(1.3.18.0.2.6.485
NAME 'ibm-slapdReferral'
DESC 'Global superior referrals for IBM Directory.'
SUP (top $ ibm-slapdConfigEntry)
STRUCTURAL
MUST (cn $ ibm-slapdReferral))

objectclasses=(1.3.18.0.2.6.496
NAME 'ibm-slapdReplication'
DESC 'Contains the default bind credentials and master server referral URL.
This is used when the server contains one or more replication contexts that
are replicated to it by other servers. This server may be acting as
one of several masters or as a read only replica. If the MasterDN is
specified without the Master PW attribute, kerberos authentication is used.'
SUP (top $ ibm-slapdConfigEntry)
STRUCTURAL
MUST (cn)
MAY (ibm-slapdMasterDN $ ibm-slapdMasterPW
$ ibm-slapdMasterReferral))

objectclasses=(1.3.18.0.2.6.596
NAME 'ibm-slapdReplicationConfiguration'
DESC 'Used to configure replication for a supplier'
SUP 'top'
STRUCTURAL
MUST (cn)
MAY (description $ ibm-slapdMaxPendingChangesDisplayed
$ ibm-slapdReplContextCacheSize $ ibm-slapdReplMaxErrors
$ ibm-slapdReplConflictMaxEntrySize $ ibm-replicationOnHold))

objectclasses=(1.3.18.0.2.6.499
NAME 'ibm-slapdSchema'
DESC 'Global schema settings for IBM Directory.
Multiple schemas are not currently supported, but if they were then there
would be one ibm-slapdSchema entry per schema.'
SUP (top $ ibm-slapdConfigEntry)
STRUCTURAL
MUST (cn $ ibm-slapdSchemaCheck $ ibm-slapdIncludeSchema)
MAY (ibm-slapdSchemaAdditions))

objectclasses=(1.3.18.0.2.6.492
NAME 'ibm-slapdSSL' DESC 'Global SSL connection settings for IBM Directory.'
SUP (top $ ibm-slapdConfigEntry)
STRUCTURAL
MUST (cn $ ibm-slapdSecurity $ ibm-slapdSecurePort
$ ibm-slapdSslAuth)
MAY (ibm-slapdSslCertificate $ ibm-slapdSslCipherSpec
$ ibm-slapdSslCipherSpecs $ ibm-slapdSSLKeyDatabase
$ ibm-slapdSSLKeyDatabasePW $ ibm-slapdSslKeyRingFilePW
$ ibm-slapdSslFIPsModeEnabled $ ibm-slapdSslFIPsProcessingMode))

objectclasses=(1.3.18.0.2.6.488
NAME 'ibm-slapdSupplier'
DESC 'Contains bind credentials used by a replication supplier server to
update the specified subtree on this consumer server. Use of this object class
overrides the default bind credentials specified in an ibm-slapdReplication
object.'
SUP (top $ ibm-slapdConfigEntry)
STRUCTURAL
MUST (cn $ ibm-slapdReplicaSubtree $ ibm-slapdMasterDN)
MAY (ibm-slapdMasterPW))

objectclasses=(1.3.18.0.2.6.498

```

```

NAME 'ibm-slapdTop'
DESC 'Global configuration settings for IBM Directory Server.'
SUP (top $ ibm-slapdConfigEntry)
STRUCTURAL
MUST (cn $ ibm-slapdAdminDN $ ibm-slapdAdminPW $ ibm-slapdPort
$ ibm-slapdPwEncryption $ ibm-slapdSizeLimit
$ ibm-slapdTimeLimit $ ibm-slapdDerefAliases
$ ibm-slapdCryptoSync)
MAY (ibm-slapdServerId $ ibm-slapdVersion
$ ibm-slapdMaxPendingChangesDisplayed $ ibm-slapdSupportedWebAdmVersion
$ ibm-slapdStartupTraceEnabled $ ibm-slapdTraceMessageLevel
$ ibm-slapdTraceMessageLog $ ibm-slapdAdminGroupEnabled
$ ibm-slapdIpAddress $ ibm-slapdServerBackend))

objectclasses=(1.3.18.0.2.6.491
NAME 'ibm-slapdTransaction'
DESC 'Global transaction support settings for IBM Directory.'
SUP (top $ ibm-slapdConfigEntry)
STRUCTURAL
MUST (cn $ ibm-slapdMaxNumOfTransactions $ ibm-slapdMaxOpPerTransaction
$ ibm-slapdMaxTimeLimitOfTransactions $ ibm-slapdTransactionEnable))

objectclasses=(1.3.18.0.2.6.589
NAME 'ids-instance'
DESC 'An entry for a ibm directory server instance in the ibm directory
server instance repository.'
SUP 'top'
STRUCTURAL
MUST (ids-instanceVersion $ ids-instanceLocation)
MAY (ids-instanceDesc)

```

---

## Configuration attributes

These are the configuration attributes that are shipped with the IBM Tivoli Directory Server Version 6.0. For descriptive names to go with the syntax OIDs, see the **V3.ldapsyntaxes** file in the **etc** directory.

# File generated at 8:26:38 AM on 8/18/2004 from IBM LDAP schema version 1.5

```

attributetypes=(1.3.18.0.2.4.3056
NAME 'ibm-auditExtOp'
DESC 'TRUE or FALSE Indicate whether to log the Extended operation.
Default is FALSE.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.3056
DBNAME('auditExOp' 'auditExOp')
ACCESS-CLASS normal
LENGTH 5)

```

```

attributetypes=(1.3.18.0.2.4.3055
NAME 'ibm-auditVersion'
DESC 'Specifies which version of auditing to use.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.3055
DBNAME('auditVersion' 'auditVersion')
ACCESS-CLASS normal
LENGTH 11)

```

```

attributetypes=(1.3.18.0.2.4.2485
NAME 'ibm-slapdACLAccess'
DESC 'If set to true anyone that can read an entry can also read the
entry's ACL attributes. If set to false only the entry owner or the

```

```

administrator can read ACL attributes.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
SINGLE-VALUE
USAGE userApplications)
IBMAttributetypes=(1.3.18.0.2.4.2485
DBNAME('slapdACLAccess' 'slapdACLAccess')
ACCESS-CLASS normal
LENGTH 5)

attributetypes=(1.3.18.0.2.4.2374
NAME 'ibm-slapdACLCache'
DESC 'Controls whether or not the server caches ACL information'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2374
DBNAME('ACLCache' 'ACLCache')
ACCESS-CLASS normal
LENGTH 5)

attributetypes=(1.3.18.0.2.4.2373
NAME 'ibm-slapdACLCacheSize'
DESC 'Maximum number of entries to keep in the ACL Cache'
EQUALITY 2.5.13.14
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2373
DBNAME('slapdACLCacheSize' 'slapdACLCacheSize')
ACCESS-CLASS normal
LENGTH 11)

attributetypes=(1.3.18.0.2.4.2428
NAME 'ibm-slapdAdminDN'
DESC 'Bind DN for ibmslapd administrator, e.g.: cn=root'
EQUALITY 2.5.13.1
ORDERING 1.3.18.0.2.4.405
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2428
DBNAME('slapdAdminDN' 'slapdAdminDN')
ACCESS-CLASS critical
LENGTH 1000
EQUALITY
ORDERING)

attributetypes=(1.3.18.0.2.4.3013
NAME 'ibm-slapdAdminGroupEnabled'
DESC 'Must be one of { TRUE | FALSE }.
Specifies whether the Administrative Group is currently enabled.
Defaults to FALSE if unspecified. If set to TRUE, the
server will allow users in the administrative group to login.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.3013
DBNAME('AdmGroupEnabled' 'AdmGroupEnabled')
ACCESS-CLASS critical
LENGTH 5)

attributetypes=(1.3.18.0.2.4.2425
NAME 'ibm-slapdAdminPW'
DESC 'Bind password for ibmslapd administrator.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.5
SINGLE-VALUE
USAGE directoryOperation)

```

```

IBMAttributetypes=(1.3.18.0.2.4.2425
 DBNAME('slapdAdminPW' 'slapdAdminPW')
 ACCESS-CLASS critical)

attributetypes=(1.3.18.0.2.4.3021
 NAME 'ibm-slapdAllowAnon'
 DESC 'Specifies if anonymous binds are allowed.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
 SINGLE-VALUE
 USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.3021
 DBNAME('slapdAllowAnon' 'slapdAllowAnon')
 ACCESS-CLASS normal
 LENGTH 5)

attributetypes=(1.3.18.0.2.4.3024
 NAME 'ibm-slapdAllReapingThreshold'
 DESC 'Specifies a number of connections to maintain in the server
 before connection management is activated.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
 SINGLE-VALUE
 USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.3024
 DBNAME('slapdAllReapingTh' 'slapdAllReapingTh')
 ACCESS-CLASS normal
 LENGTH 11)

attributetypes=(1.3.18.0.2.4.3022
 NAME 'ibm-slapdAnonReapingThreshold'
 DESC 'Specifies a number of connections to maintain in the server
 before connection management of
 anonymous connections is activated.'
 EQUALITY 2.5.13.14
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
 SINGLE-VALUE
 USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.3022
 DBNAME('slapdAnonReapingT' 'slapdAnonReapingT')
 ACCESS-CLASS normal
 LENGTH 11)

attributetypes=(1.3.18.0.2.4.2366
 NAME 'ibm-slapdAuthIntegration'
 DESC 'Specifies integration of LDAP administrator access with
 local OS users. Legal values are:
 0 - do not map local OS users to LDAP administrator,
 1 - map local OS users with proper authority to LDAP administrator.
This is supported only on i5/OS.'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
 SINGLE-VALUE
 USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2366
 DBNAME('slapdAuthIntegrat' 'slapdAuthIntegrat')
 ACCESS-CLASS system
 LENGTH 11)

attributetypes=(1.3.18.0.2.4.3023
 NAME 'ibm-slapdBoundReapingThreshold'
 DESC 'Specifies a number of connections to maintain in the server
 before connection management of anonymous and bound
 connections is activated.'
 EQUALITY 2.5.13.14
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
 SINGLE-VALUE
 USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.3023
 DBNAME('slapdBoundReaping' 'slapdBoundReaping')

```

```

ACCESS-CLASS normal
LENGTH 11)

attributetypes=(1.3.18.0.2.4.2368
NAME 'ibm-slapdBulkloadErrors'
DESC 'File path or device on ibmslapd host machine to
which bulkload error messages will be written.
On Windows, forward slashes are allowed, and a leading
slash not preceded by a drive letter is assumed to
be rooted at the install directory
(i.e.: /tmp/bulkload.errors = D:\Program Files\IBM\ldap\tmp\bulkload.errors).'
EQUALITY 2.5.13.5
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
USAGE userApplications)
IBMAttributetypes=(1.3.18.0.2.4.2368
DBNAME('slapdBulkloadErro' 'slapdBulkloadErro')
ACCESS-CLASS normal
LENGTH 1024)

attributetypes=(1.3.18.0.2.4.3069
NAME 'ibm-slapdCachedAttribute'
DESC 'Contains the names of the attributes to be cached in the
attribute cache, one attribute name per value.'
EQUALITY 1.3.6.1.4.1.1466.109.114.2
ORDERING 2.5.13.3
SUBSTR 2.5.13.4
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.3069
DBNAME('slapdCachedAttr' 'slapdCachedAttr')
ACCESS-CLASS normal
LENGTH 256)

attributetypes=(1.3.18.0.2.4.3068
NAME 'ibm-slapdCachedAttributeSize'
DESC 'Amount of memory, in bytes, that can be used by the
attribute cache. A value of 0 indicates not use an
attribute cache.'
EQUALITY 2.5.13.14
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.3068
DBNAME('slapdAttrCacheSz' 'slapdAttrCacheSz')
ACCESS-CLASS normal
LENGTH 11)

attributetypes=(1.3.18.0.2.4.3012
NAME 'ibm-slapdChangeLogMaxAge'
DESC 'Specifies the maximum age, in hours, of changelog entries
allowed in the associated backend. Each changelog backend has
its own ibm-slapdChangeLogMaxAge attribute. If the attribute is
undefined or out of range (negative), it defaults to 0.
Min: 0 (unlimited) Max: 2,147,483,647 (32-bit, signed integer)'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE userApplications)
IBMAttributetypes=(1.3.18.0.2.4.3012
DBNAME('chgLogMaxAge' 'chgLogMaxAge')
ACCESS-CLASS critical
LENGTH 11)

attributetypes=(1.3.18.0.2.4.2427
NAME 'ibm-slapdChangeLogMaxEntries'
DESC 'Specifies the maximum number of changelog entries allowed
in the associated backend. Each changelog backend has
its own ibm-slapdChangeLogMaxEntries attribute. If the

```



```

attribute is undefined or out of range (negative), it defaults to 0.
Min: 0 (unlimited) Max: 2,147,483,647 (32-bit, signed integer)'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE userApplications)
IBMAttributetypes=(1.3.18.0.2.4.2427
DBNAME('chgLogMaxEntries' 'chgLogMaxEntries')
ACCESS-CLASS critical
LENGTH 11)

attributetypes=(1.3.18.0.2.4.2432
NAME 'ibm-slapdCLIErrors'
DESC 'File path or device on ibmslapd host machine to which
DB2 CLI error messages will be written. On Windows,
forward slashes are allowed, and a leading slash not
preceded by a drive letter is assumed to be rooted at
the install directory
(i.e.: /tmp/cli.errors = D:\Program Files\IBM\ldap\tmp\cli.errors).'
EQUALITY 2.5.13.5
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2432
DBNAME('slapdCLIErrors' 'slapdCLIErrors')
ACCESS-CLASS normal
LENGTH 1024)

attributetypes=(1.3.18.0.2.4.3140
NAME 'ibm-slapdConfigPwdPolicyOn'
DESC 'TRUE or FALSE. Indicates if the IBM Administrative
Password Policy is ON'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.3140
DBNAME('ConfigPwdPolicyOn' 'ConfigPwdPolicyOn')
ACCESS-CLASS critical
LENGTH 5)

attributetypes=(1.3.18.0.2.4.3116
NAME 'ibm-slapdCryptoSync'
DESC 'A key stash file consistency marker string.
It is queried by the server at start up as part of
a verification process to ensure that the key stash
files match any data that has been two-way encrypted.'
EQUALITY 2.5.13.17
SYNTAX 1.3.6.1.4.1.1466.115.121.1.5
SINGLE-VALUE
NO-USER-MODIFICATION
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.3116
DBNAME('CryptoSync' 'CryptoSync')
ACCESS-CLASS system)

attributetypes=(1.3.18.0.2.4.2369
NAME 'ibm-slapdDB2CP'
DESC 'Specifies the Code Page of the directory database.
1208 is the code page for UTF-8 databases.'
EQUALITY 2.5.13.5
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2369
DBNAME('slapdDB2CP' 'slapdDB2CP')
ACCESS-CLASS normal
LENGTH 11)

```

```

attributetypes=(1.3.18.0.2.4.2431
NAME 'ibm-slapdDBAlias'
DESC 'The DB2 database alias.'
EQUALITY 2.5.13.5
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2431
DBNAME('slapdDBAlias' 'slapdDBAlias')
ACCESS-CLASS normal
LENGTH 8)

attributetypes=(1.3.18.0.2.4.2417
NAME 'ibm-slapdDbConnections'
DESC 'The number of DB2 connections the server will
dedicate to the DB2 backend. The value must
be 5 or greater. Additional connections may be created
for replication and change log.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2417
DBNAME('DbConnections' 'DbConnections')
ACCESS-CLASS critical
LENGTH 2)

attributetypes=(1.3.18.0.2.4.2418
NAME 'ibm-slapdDbInstance'
DESC 'The DB2 database instance for this backend.'
EQUALITY 2.5.13.5
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2418
DBNAME('slapdDbInstance' 'slapdDbInstance')
ACCESS-CLASS critical
LENGTH 8)

attributetypes=(1.3.18.0.2.4.2382
NAME 'ibm-slapdDbLocation'
DESC 'The file system path where the backend database
is located. On Unix this is usually the home directory of the
DB2INSTANCE owner (e.g.: /home/ldapdb2). On Windows it is
just a drive specifier (e.g.: D:)'
EQUALITY 2.5.13.5
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2382
DBNAME('slapdDbLocation' 'slapdDbLocation')
ACCESS-CLASS critical
LENGTH 1024)

attributetypes=(1.3.18.0.2.4.2426
NAME 'ibm-slapdDbName'
DESC 'The DB2 database name for this backend.'
EQUALITY 2.5.13.5
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2426
DBNAME('slapdDbName' 'slapdDbName')
ACCESS-CLASS critical
LENGTH 8)

attributetypes=(1.3.18.0.2.4.2422
NAME 'ibm-slapdDbUserID'
DESC 'The user name with which to connect to the DB2

```

```

database for this backend.'
EQUALITY 2.5.13.5
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2422
DBNAME('slapDbUserID' 'slapDbUserID')
ACCESS-CLASS critical
LENGTH 8)

attributetypes=(1.3.18.0.2.4.2423
NAME 'ibm-slapDbUserPW'
DESC 'The user password with which to connect to
the DB2 database for this backend.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.5
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2423
DBNAME('slapDbUserPW' 'slapDbUserPW')
ACCESS-CLASS critical)

attributetypes=(1.3.18.0.2.4.3054
NAME 'ibm-slapdDerefAliases'
DESC 'Maximum alias dereferencing level on search requests,
regardless of any derefAliases that may have been specified
on the client requests. Allowed values are never, find,
search and always.'
EQUALITY 2.5.13.2
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.3054
DBNAME('slapDerefAliases' 'slapDerefAliases')
ACCESS-CLASS normal
LENGTH 6)

attributetypes=(1.3.18.0.2.4.3032
NAME 'ibm-slapdDigestAdminUser'
DESC 'Specifies the Digest MD5 User Name of the LDAP
administrator or administrative group member.
Used when MD5 Digest authentication is used to
authenticate an administrator.'
EQUALITY 2.5.13.5
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.3032
DBNAME('DigestAdminUser' 'DigestAdminUser')
ACCESS-CLASS critical
LENGTH 512)

attributetypes=(1.3.18.0.2.4.3082
NAME 'ibm-slapdDigestAttr'
DESC 'Overrides the default DIGEST-MD5 username attribute.
The name of the attribute to use for DIGEST-MD5 SASL
bind username lookup. If the value is not specified,
the server uses uid.'
EQUALITY 2.5.13.0
SYNTAX 1.3.6.1.4.1.1466.115.121.1.38
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.3082
DBNAME('slapdDigestAttr' 'slapdDigestAttr')
ACCESS-CLASS critical
LENGTH 128)

attributetypes=(1.3.18.0.2.4.3083
NAME 'ibm-slapdDigestRealm'

```

```

DESC 'Overrides the default DIGEST-MD5 realm.
A string which can enable users to know which username
and password to use, in case they might have
different ones for different servers. Conceptually, it is
the name of a collection of accounts that might include
the users account. This string should contain at least
the name of the host performing the authentication and
might additionally indicate the collection of users who
might have access. An example might be
registered_users@gotham.news.example.com. If the attribute
is not specified, the server uses the fully qualified hostname
of the server.'
EQUALITY 2.5.13.5
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.3083
DBNAME('slapdDigestRealm' 'slapdDigestRealm')
ACCESS-CLASS critical
LENGTH 1024)

attributetypes=(1.3.18.0.2.4.3123
NAME 'ibm-slapdDistributedDynamicGroups'
DESC 'Switch that determines whether the proxy allows
for dynamic group evaluation (e.g. ibm-allmembers).'

```

```

SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2424
 DBNAME('slapdErrorLog' 'slapdErrorLog')
ACCESS-CLASS critical
LENGTH 1024)

attributetypes=(1.3.18.0.2.4.3028
NAME 'ibm-slapdESizeThreshold'
DESC 'Specifies the number of work items on the work queue
before the Emergency thread is activated.'
EQUALITY 2.5.13.14
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.3028
DBNAME('slapdESizeThresho' 'slapdESizeThresho')
ACCESS-CLASS normal
LENGTH 11)

attributetypes=(1.3.18.0.2.4.3030
NAME 'ibm-slapdEThreadActivate'
DESC 'Specifies which conditions will activate the Emergency Thread.
Must be set to one of the following values:
S - size only, T - time only, SOT - size or time,
SAT - size and time.'
EQUALITY 2.5.13.5
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.3030
DBNAME('slapdEThreadActiv' 'slapdEThreadActiv')
ACCESS-CLASS normal
LENGTH 1024)

attributetypes=(1.3.18.0.2.4.3031
NAME 'ibm-slapdEThreadEnable'
DESC 'Specifies if the Emergency Thread can be activated.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.3031
DBNAME('slapdEThreadEnabl' 'slapdEThreadEnabl')
ACCESS-CLASS normal
LENGTH 5)

attributetypes=(1.3.18.0.2.4.3029
NAME 'ibm-slapdETimeThreshold'
DESC 'Specifies the amount of time in minutes between items
removed from the work queue before the Emergency thread is activated.'
EQUALITY 2.5.13.14
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.3029
DBNAME('slapdETimeThresho' 'slapdETimeThresho')
ACCESS-CLASS normal
LENGTH 11)

attributetypes=(1.3.18.0.2.4.2371
NAME 'ibm-slapdFilterCacheBypassLimit'
DESC 'Search filters that match more than this number of
entries will not be added to the Search Filter cache.
Because the list of entry ids that matched the filter
are included in this cache, this setting helps to limit
memory use. A value of 0 indicates no limit.'

```

```

EQUALITY 2.5.13.14
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2371
DBNAME('slapdRDBMCacheByP' 'slapdRDBMCacheByP')
ACCESS-CLASS normal
LENGTH 11)

attributetypes=(1.3.18.0.2.4.2370
NAME 'ibm-slapdFilterCacheSize'
DESC 'Specifies the maximum number of entries to keep in the
Search Filter Cache.'
EQUALITY 2.5.13.14
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2370
DBNAME('slapdFilterCacheS' 'slapdFilterCacheS')
ACCESS-CLASS normal
LENGTH 11)

attributetypes=(1.3.18.0.2.4.2378
NAME 'ibm-slapdIdleTimeOut'
DESC 'Reserved for future use.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2378
DBNAME('SlapdIdleTimeOut' 'SlapdIdleTimeOut')
ACCESS-CLASS critical
LENGTH 11)

attributetypes=(1.3.18.0.2.4.2364
NAME 'ibm-slapdIncludeSchema'
DESC 'File path on ibmslapd host machine containing
schema definitions used by the LDCF backend. Standard values are:
/etc/V3.system.at /etc/V3.system.oc /etc/V3.ibm.at
/etc/V3.ibm.oc /etc/V3.user.at /etc/V3.user.oc
/etc/V3.ldapsyntaxes /etc/V3.matchingrules /etc/V3.modifiedschema
On Windows, forward slashes are allowed, and a leading slash not
preceded by a drive letter is assumed to be rooted at the
install directory
(i.e.: /etc/V3.system.at = D:\Program Files\IBM\ldap\etc\V3.system.at).'
EQUALITY 2.5.13.5
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2364
DBNAME('slapdIncludeSchema' 'slapdIncludeSchema') A
ACCESS-CLASS critical
LENGTH 1024)

attributetypes=(1.3.18.0.2.4.2430
NAME 'ibm-slapdInvalidLine'
DESC 'This attribute will be prepended to the beginning of
any configuration attribute for which the value is invalid.
This allows invalid configuration settings to be identified
with a simple search for ibm-slapdInvalidLine=*'
EQUALITY 2.5.13.2
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
USAGE userApplications)
IBMAttributetypes=(1.3.18.0.2.4.2430
DBNAME('slapdInvalidLine' 'slapdInvalidLine')
ACCESS-CLASS normal
LENGTH 1024)

attributetypes=(1.3.18.0.2.4.2365
NAME 'ibm-slapdIpAddress' DESC 'Specifies IP addresses the

```

```

server will listen on. These can be IPv4 or IPv6 addresses.
If the attribute is not specified, the server uses all
IP addresses assigned to the host machine.'
EQUALITY 1.3.6.1.4.1.1466.109.114.1
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2365
DBNAME('slapdIpAddress' 'slapdIpAddress')
ACCESS-CLASS system
LENGTH 32)

```

```

attributetypes=(1.3.18.0.2.4.2420
NAME 'ibm-slapdKrbAdminDN'
DESC 'Specifies the kerberos ID of the LDAP administrator
(e.g. ibm-kn=name@realm). Used when kerberos authentication
is used to authenticate the administrator when logged onto
the Web Admin interface. This is specified instead
of adminDN and adminPW.'
EQUALITY 2.5.13.5
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2420
DBNAME('slapdKrbAdminDN' 'slapdKrbAdminDN')
ACCESS-CLASS critical
LENGTH 512)

```

```

attributetypes=(1.3.18.0.2.4.2394
NAME 'ibm-slapdKrbEnable'
DESC 'Must be one of { TRUE | FALSE }. Specifies whether the
server supports kerberos authentication.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2394
DBNAME('slapdKrbEnable' 'slapdKrbEnable')
ACCESS-CLASS critical
LENGTH 5)

```

```

attributetypes=(1.3.18.0.2.4.2419
NAME 'ibm-slapdKrbIdentityMap'
DESC 'If set to TRUE, when a client is authenticated
with a kerberos ID, the server will search for a local
user with matching kerberos credentials, and add that
users DN to the connections bind credentials.
This allows ACLs based on LDAP user DN's to still be
usable with kerberos authentication.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2419
DBNAME('KrbIdentityMap' 'KrbIdentityMap')
ACCESS-CLASS critical
LENGTH 5)

```

```

attributetypes=(1.3.18.0.2.4.2416
NAME 'ibm-slapdKrbKeyTab'
DESC 'Specifies the LDAP servers keytab file. This
file contains the LDAP servers private key, as associated
with its kerberos account. This file should be protected
(like the servers SSL key database file).
On Windows, forward slashes are allowed, and a leading
slash not preceded by a drive letter (D:) is assumed
to be rooted at the install directory
(i.e.: /tmp/slapd.errors = D:\Program Files\IBM\ldap\tmp\slapd.errors).'
EQUALITY 2.5.13.5
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15

```

```

SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2416
DBNAME('slapdKrbKeyTab' 'slapdKrbKeyTab')
ACCESS-CLASS critical
LENGTH 1024)

attributetypes=(1.3.18.0.2.4.2400
NAME 'ibm-slapdKrbRealm'
DESC 'Specifies the LDAP servers kerberos realm.
Used to publish the ldap servicename attribute in the root DSE.
Note that an LDAP server can serve as the repository of
account information for multiple KDCs (and realms),
but the LDAP server, as a kerberos server, can only
be a member of a single realm.'
EQUALITY 2.5.13.2
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2400
DBNAME('slapdKrbRealm' 'slapdKrbRealm')
ACCESS-CLASS critical
LENGTH 256)

attributetypes=(1.3.18.0.2.4.3074
NAME 'ibm-slapdLanguageTagsEnabled'
DESC 'Specifies whether or not the directory server
will allow Language Tags as part of an attribute description.
Possible values include TRUE and FALSE.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.3074
DBNAME('slapdLanguageTags' 'slapdLanguageTags')
ACCESS-CLASS critical
LENGTH 5)

attributetypes=(1.3.18.0.2.4.2415
NAME 'ibm-slapdLdapCr1Host'
DESC 'Specify the hostname of the LDAP server that
contains the Certificate Revocation Lists (CRLs) for
validating client x.509v3 certificates. This parameter
is needed when ibm-slapdSslAuth=serverclientauth AND the client
certificates have been issued for CRL validation'
EQUALITY 2.5.13.2
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2415
DBNAME('LdapCr1Host' 'LdapCr1Host')
ACCESS-CLASS critical
LENGTH 256)

attributetypes=(1.3.18.0.2.4.2407
NAME 'ibm-slapdLdapCr1Password'
DESC 'Specify the password that server-side SSL
will use to bind to the LDAP server that contains the Certificate
Revocation Lists (CRLs) for validating client x.509v3 certificates.
This parameter may be needed when
ibm-slapdSslAuth=serverclientauth AND the client certificates
have been issued for CRL validation.
Note: If the LDAP server holding the CRLs permits unauthenticated
access to the CRLs (i.e. anonymous access), then
ibm-slapdLdapCr1Password is not required.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.5 S
INGLE-VALUE
USAGE directoryOperation)

```



```

IBMAttributetypes=(1.3.18.0.2.4.2407
DBNAME('Cr1Password' 'Cr1Password')
ACCESS-CLASS critical)

attributetypes=(1.3.18.0.2.4.2404
NAME 'ibm-slapdLdapCr1Port'
DESC 'Specify the LDAP ibm-slapdPort used by the LDAP
server that contains the Certificate Revocation Lists
(CRLs) for validating client x.509v3 certificates.
This parameter is needed when ibm-slapdSslAuth=serverclientauth
AND the client certificates have been issued for CRL validation.
(IP ports are unsigned, 16-bit integers in the range 1 - 65535)'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2404
DBNAME('LdapCr1Port' 'LdapCr1Port')
ACCESS-CLASS critical
LENGTH 11)

attributetypes=(1.3.18.0.2.4.2403
NAME 'ibm-slapdLdapCr1User'
DESC 'Specify the bindDN that server-side SSL will use
to bind to the LDAP server that contains the Certificate Revocation
Lists (CRLs) for validating client x.509v3 certificates.
This parameter may be needed when ibm-slapdSslAuth=serverclientaut
h AND the client certificates have been issued for CRL validation.
Note: If the LDAP server holding the CRLs permits
unauthenticated access to the CRLs (i.e. anonymous access),
then ibm-slapdLdapCr1User is not required.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2403
DBNAME('LdapCr1User' 'LdapCr1User')
ACCESS-CLASS critical
LENGTH 1000)

attributetypes=(1.3.18.0.2.4.3128
NAME 'ibm-slapdLog'
DESC 'Log path and file name. On Windows, forward slashes
are allowed, and a leading slash not preceded by a drive letter
is assumed to be rooted at the install directory
(i.e.: /tmp/bulkload.errors = D:\Program Files\IBM\ldap\tmp\bulkload.errors).'
EQUALITY 2.5.13.5
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE userApplications)
IBMAttributetypes=(1.3.18.0.2.4.3128
DBNAME('ibmlog' 'ibmlog') ACCESS-CLASS critical
LENGTH 1024)

attributetypes=(1.3.18.0.2.4.3134
NAME 'ibm-slapdLogArchivePath'
DESC 'Path for archived files. On Windows, forward slashes
are allowed, and a leading slash not preceded by a drive letter
is assumed to be rooted at the install directory
(i.e.: /tmp = D:\Program Files\IBM\ldap\tmp).'
EQUALITY 2.5.13.5
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE userApplications)
IBMAttributetypes=(1.3.18.0.2.4.3134
DBNAME('logArchivePath' 'logArchivePath')
ACCESS-CLASS critical
LENGTH 1024)

attributetypes=(1.3.18.0.2.4.3129

```

```

NAME 'ibm-slapdLogMaxArchives'
DESC 'The maximum number of archived logs where 0 means no
archive file will be kept and -1 means an unlimited number of
archive files will be kept.'
EQUALITY 2.5.13.14
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE userApplications)
IBMAttributetypes=(1.3.18.0.2.4.3129
DBNAME('logMaxArchives' 'logMaxArchives')
ACCESS-CLASS critical
LENGTH 11)

attributetypes=(1.3.18.0.2.4.3130
NAME 'ibm-slapdLogOptions'
DESC 'Any log options that the log uses, for example, log level or mask.'
EQUALITY 2.5.13.2
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE userApplications)
IBMAttributetypes=(1.3.18.0.2.4.3130
DBNAME('logOptions' 'logOptions')
ACCESS-CLASS critical
LENGTH 30)

attributetypes=(1.3.18.0.2.4.3131
NAME 'ibm-slapdLogSizeThreshold'
DESC 'When this size threshold, in MB, is exceeded the file
will be archived where 0 means no threshold and thus no archiving.'
EQUALITY 2.5.13.14
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE userApplications)
IBMAttributetypes=(1.3.18.0.2.4.3131
DBNAME('logSizeThreshold' 'logSizeThreshold')
ACCESS-CLASS critical
LENGTH 11)

attributetypes=(1.3.18.0.2.4.2409
NAME 'ibm-slapdMasterDN'
DESC 'Bind DN used by a replication supplier server.
The value has to match the replicaBindDN in the credentials
object associated with the replication agreement. When
kerberos is used to authenticate to the replica,
ibm-slapdMasterDN must specify the DN representation of
the kerberos ID (e.g. ibm-kn=freddy@realm1).
When kerberos is used, MasterServerPW is ignored.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2409
DBNAME('MasterDN' 'MasterDN')
ACCESS-CLASS critical
LENGTH 1000)

attributetypes=(1.3.18.0.2.4.2411
NAME 'ibm-slapdMasterPW'
DESC 'Bind password used by replication supplier server. The value has
to match the replicaBindPW in the credentials
object associated with the replication agreement.
When kerberos is used, MasterServerPW is ignored.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.5
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2411
DBNAME('MasterPW' 'MasterPW')
ACCESS-CLASS critical)

```

```

attributetypes=(1.3.18.0.2.4.2401
NAME 'ibm-slapdMasterReferral'
DESC 'URL of master replica server (e.g.: ldaps://master.us.ibm.com:636)'
EQUALITY 2.5.13.2
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2401
DBNAME('MasterReferral' 'MasterReferral')
ACCESS-CLASS critical
LENGTH 256)

```

```

attributetypes=(1.3.18.0.2.4.2412
NAME 'ibm-slapdMaxEventsPerConnection'
DESC 'Maximum number of event notifications which can be registered
per connection.
Minimum = 0 (unlimited) Maximum = 2,147,483,647'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2412
DBNAME('EventsPerCon' 'EventsPerCon')
ACCESS-CLASS critical
LENGTH 11)

```

```

attributetypes=(1.3.18.0.2.4.2405
NAME 'ibm-slapdMaxEventsTotal'
DESC 'Maximum total number of event notifications which can be registered
for all connections.
Minimum = 0 (unlimited) Maximum = 2,147,483,647'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2405
DBNAME('MaxEventsTotal' 'MaxEventsTotal')
ACCESS-CLASS critical
LENGTH 11)

```

```

attributetypes=(1.3.18.0.2.4.2439
NAME 'ibm-slapdMaxNumOfTransactions'
DESC 'Maximum number of transactions active at one time.
0 = unlimited'
EQUALITY 2.5.13.29
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2439
DBNAME('MaxNumOfTrans' 'MaxNumOfTrans')
ACCESS-CLASS critical
LENGTH 11
EQUALITY
ORDERING
SUBSTR
APPROX)

```

```

attributetypes=(1.3.18.0.2.4.2385
NAME 'ibm-slapdMaxOpPerTransaction'
DESC 'Maximum number of operations per transaction. 0 = unlimited'
EQUALITY 2.5.13.29
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2385
DBNAME('MaxOpPerTrans' 'MaxOpPerTrans')
ACCESS-CLASS critical
LENGTH 11

```

EQUALITY  
ORDERING  
APPROX )

attributetypes=( 1.3.18.0.2.4.2486  
NAME 'ibm-slapdMaxPendingChangesDisplayed'  
DESC 'Maximum number of pending replication updates or failed  
updates to be displayed for any given replication agreement  
on a supplier server. The value is dynamic'  
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27  
USAGE userApplications )  
IBMAttributetypes=( 1.3.18.0.2.4.2486  
DBNAME( 'slapdMaxPendingCh' 'slapdMaxPendingCh' )  
ACCESS-CLASS normal  
LENGTH 11 )

attributetypes=( 1.3.18.0.2.4.2386  
NAME 'ibm-slapdMaxTimeLimitOfTransactions'  
DESC 'The maximum timeout value of a pending transaction in seconds.  
0 = unlimited'  
EQUALITY 2.5.13.29  
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27  
SINGLE-VALUE  
USAGE directoryOperation )  
IBMAttributetypes=( 1.3.18.0.2.4.2386  
DBNAME( 'MaxTimeOfTrans' 'MaxTimeOfTrans' )  
ACCESS-CLASS critical  
LENGTH 11  
EQUALITY  
ORDERING  
APPROX )

attributetypes=( 1.3.18.0.2.4.2500  
NAME 'ibm-slapdMigrationInfo'  
DESC 'Information used to control migration of a component.'  
EQUALITY 2.5.13.2  
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15  
USAGE directoryOperation )  
IBMAttributetypes=( 1.3.18.0.2.4.2500  
DBNAME( 'slapdMigrationInf' 'slapdMigrationInf' )  
ACCESS-CLASS critical  
LENGTH 2048 )

attributetypes=( 1.3.18.0.2.4.2376  
NAME 'ibm-slapdPagedResAllowNonAdmin'  
DESC 'Whether or not the server should allow non-Administrator  
bind for paged results requests on a search request.  
If the value read from the ibmslapd.conf file is TRUE, the  
server will process any client request, including those  
submitted by a user binding anonymously. If the value read  
from the ibmslapd.conf file is FALSE, the server will  
process only those client requests submitted by a user  
with Administrator authority. If a client requests paged  
results with a criticality of TRUE or FALSE for a search  
operation, does not have Administrator authority, and the  
value read from the ibmslapd.conf file for this attribute is FALSE,  
the server will return to the client with return code  
insufficientAccessRights - no searching or paging will be performed. '

SYNTAX 1.3.6.1.4.1.1466.115.121.1.7  
SINGLE-VALUE  
USAGE directoryOperation )  
IBMAttributetypes=( 1.3.18.0.2.4.2376  
DBNAME( 'SlapdPagedNonAdmn' 'SlapdPagedNonAdmn' )  
ACCESS-CLASS critical  
LENGTH 5 )

attributetypes=( 1.3.18.0.2.4.2380

```

NAME 'ibm-slapdPagedResLmt'
DESC 'Maximum number of outstanding paged results search
requests allowed active simultaneously. Range = 0...
If a client requests a paged results operation, and a
maximum number of outstanding paged results are currently
active, then the server will return to the client with
return code of busy - no searching or paging will be performed.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributeTypes=(1.3.18.0.2.4.2380
DBNAME('SlapdPagedResLmt' 'SlapdPagedResLmt')
ACCESS-CLASS critical L
ENGTH 11)

attributetypes=(1.3.18.0.2.4.2406
NAME 'ibm-slapdPlugin'
DESC 'A plugin is a dynamically loaded library which extends
the capabilities of the server. An ibm-slapdPlugin
attribute specifies to the server how to load and initialize
a plugin library. The syntax is:
keyword filename init_function [args...] The syntax will be
slightly different for each platform due to library
naming conventions.'
EQUALITY 2.5.13.5
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
USAGE directoryOperation)
IBMAttributeTypes=(1.3.18.0.2.4.2406 DBNAME
('slapdPlugin' 'slapdPlugin')
ACCESS-CLASS critical
LENGTH 2000)

attributetypes=(1.3.18.0.2.4.2408
NAME 'ibm-slapdPort'
DESC 'TCP/IP port used for non-SSL connections. Can not
have the same value as ibm-slapdSecurePort.
(IP ports are unsigned, 16-bit integers in the range 1 - 65535)'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributeTypes=(1.3.18.0.2.4.2408
DBNAME('slapdPort' 'slapdPort')
ACCESS-CLASS critical
LENGTH 5)

attributetypes=(1.3.18.0.2.4.3146
NAME 'ibm-slapdProxyBackendServerDn'
DESC 'Reference to a configuration file entry describing
a proxy backend server.'
EQUALITY 2.5.13.2
ORDERING 2.5.13.3
SUBSTR 2.5.13.4
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE distributedOperation)
IBMAttributeTypes=(1.3.18.0.2.4.3146
DBNAME('slapdProxyBSDn' 'slapdProxyBSDn')
ACCESS-CLASS critical
LENGTH 2048)

attributetypes=(1.3.18.0.2.4.3117
NAME 'ibm-slapdProxyBindMethod'
DESC 'The method used to bind to backend servers. Must be one of
simple/digest/Kerberos.'
EQUALITY 2.5.13.2
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE

```

```

USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.3117
 DBNAME('slapdProxyBindMet' 'slapdProxyBindMet')
ACCESS-CLASS critical
LENGTH 50)

attributetypes=(1.3.18.0.2.4.3118
NAME 'ibm-slapdProxyConnectionPoolSize'
DESC 'The number of connections to be maintained by the proxy
server to an individual backend server.'
EQUALITY 2.5.13.14
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.3118
DBNAME('slapdProxyConnect' 'slapdProxyConnect')
ACCESS-CLASS critical
LENGTH 11)

attributetypes=(1.3.18.0.2.4.3135
NAME 'ibm-slapdProxyDigestRealm'
DESC 'Optional attribute to provide the realm of the digest
MD-5 bind when binding to a backend server'
EQUALITY 2.5.13.2
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.3135
DBNAME('slapdProxyDigestR' 'slapdProxyDigestR')
ACCESS-CLASS critical
LENGTH 2048)

attributetypes=(1.3.18.0.2.4.3119
NAME 'ibm-slapdProxyDigestUserName'
DESC 'The username to be used when DIGEST is selected as the
bind method to a backend server'
EQUALITY 2.5.13.2
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.3119
DBNAME('slapdProxyDigestU' 'slapdProxyDigestU')
ACCESS-CLASS critical
LENGTH 2048)

attributetypes=(1.3.18.0.2.4.3120
NAME 'ibm-slapdProxyDn'
DESC 'The DN the proxy server will use to bind to backend
server nodes.'
EQUALITY 2.5.13.2
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.3120
DBNAME('slapdProxyDn' 'slapdProxyDn')
ACCESS-CLASS critical
LENGTH 2048)

attributetypes=(1.3.18.0.2.4.3143
NAME 'ibm-slapdProxyNumPartitions'
DESC 'Specifies the number of servers a given container is
split between.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.3143
DBNAME('slapdProxyNumPart' 'slapdProxyNumPart')
ACCESS-CLASS critical

```

```

LENGTH 11)

attributetypes=(1.3.18.0.2.4.3144
NAME 'ibm-slapdProxyPartitionBase'
DESC 'Defines the base at which a container is to be split.
Entries below this DN will be split among any number of servers
defined with the same base.'
EQUALITY 2.5.13.2
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE distributedOperation)
IBMAttributetypes=(1.3.18.0.2.4.3144
DBNAME('slapdProxyPBase' 'slapdProxyPBase')
ACCESS-CLASS normal
LENGTH 2048)

attributetypes=(1.3.18.0.2.4.3145
NAME 'ibm-slapdProxyPartitionIndex'
DESC 'The unique index a given server is assigned in a
split container. The value here must be <= the corresponding
ibm-slapdProxyNumPartitionsValue. The first value begins at 1.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
USAGE userApplications)
IBMAttributetypes=(1.3.18.0.2.4.3145
DBNAME('slapdProxyPartiti' 'slapdProxyPartiti')
ACCESS-CLASS normal
LENGTH 11)

attributetypes=(1.3.18.0.2.4.3121
NAME 'ibm-slapdProxyPw'
DESC 'The password credentials the proxy server will when binding
to a backend server node.'
EQUALITY 2.5.13.2
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE USAGE userApplications)
IBMAttributetypes=(1.3.18.0.2.4.3121
DBNAME('slapdProxyPw' 'slapdProxyPw')
ACCESS-CLASS critical
LENGTH 2048)

attributetypes=(1.3.18.0.2.4.3122
NAME 'ibm-slapdProxyTargetURL'
DESC 'The URL of a backend server. This must be in the form
ldap:// or ldaps:// (to indicate SSL use ldaps).'
EQUALITY 2.5.13.2
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.3122
DBNAME('slapdProxyTargetU' 'slapdProxyTargetU')
ACCESS-CLASS critical
LENGTH 2048)

attributetypes=(1.3.18.0.2.4.2402
NAME 'ibm-slapdPwEncryption'
DESC 'Must be one of { none | aes128 | aes192 | aes256 | crypt | sha }.
Specify the encoding mechanism for the user
passwords before they are stored in the directory.
Defaults to none if unspecified. If the value is set
other than none, SASL cram-md5 bind will fail.'
EQUALITY 2.5.13.2
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2402
DBNAME('PwEncryption' 'PwEncryption')
ACCESS-CLASS critical

```

```

LENGTH 6)

attributetypes=(1.3.18.0.2.4.2413
NAME 'ibm-slapdReadOnly'
DESC 'Must be one of { TRUE | FALSE }. Specifies whether the
backend can be written to. Defaults to FALSE if unspecified.
If set to TRUE, the server will return LDAP_UNWILLING_TO_PERFORM (0x35)
in response to any client request which would change data in
the readOnly database.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2413
DBNAME('ReadOnly' 'ReadOnly')
ACCESS-CLASS critical
LENGTH 5)

attributetypes=(1.3.18.0.2.4.2487
NAME 'ibm-slapdReferral'
DESC 'Specify the referral LDAP URL to pass back
when the local suffixes do not match the request.
Used for superior referral (i.e. ibm-slapdSuffix
is not within the servers naming context).'
EQUALITY 2.5.13.5
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2487
DBNAME('Referral' 'Referral')
ACCESS-CLASS critical
LENGTH 32700)

attributetypes=(1.3.18.0.2.4.3142
NAME 'ibm-slapdReplConflictMaxEntrySize'
DESC 'Maximum number of bytes that an entry can contain
and still be resent to a target server as a result of
replication conflict resolution. This value is dynamic.'
EQUALITY 2.5.13.14
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE userApplications)
IBMAttributetypes=(1.3.18.0.2.4.3142
DBNAME('slapdReplConflict' 'slapdReplConflict')
ACCESS-CLASS normal
LENGTH 11)

attributetypes=(1.3.18.0.2.4.3153
NAME 'ibm-slapdReplContextCacheSize'
DESC 'Maximum size of replication context cache,
in bytes. The value is dynamic.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE userApplications)
IBMAttributetypes=(1.3.18.0.2.4.3153
DBNAME('slapdReplContextC' 'slapdReplContextC')
ACCESS-CLASS normal
LENGTH 11)

attributetypes=(1.3.18.0.2.4.2434
NAME 'ibm-slapdReplDbConns'
DESC 'Number of database connections for use by replication.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE userApplications)
IBMAttributetypes=(1.3.18.0.2.4.2434
DBNAME('slapdReplDbConns' 'slapdReplDbConns')
ACCESS-CLASS normal
LENGTH 11)

```



```

attributetypes=(1.3.18.0.2.4.2367
NAME 'ibm-slapdReplicaSubtree'
DESC 'A DN identifying the top of a replicated subtree.'
EQUALITY 2.5.13.1
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
USAGE userApplications)
IBMAttributetypes=(1.3.18.0.2.4.2367
DBNAME('slapdReplicaSubtr' 'slapdReplicaSubtr')
ACCESS-CLASS normal
LENGTH 1000)

attributetypes=(1.3.18.0.2.4.3152
NAME 'ibm-slapdReplMaxErrors'
DESC 'Limit to allowed errors per replication agreement, 0=unlimited.
The value is dynamic.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE userApplications)
IBMAttributetypes=(1.3.18.0.2.4.3152
DBNAME('slapdReplMaxError' 'slapdReplMaxError')
ACCESS-CLASS normal
LENGTH 11)

attributetypes=(1.3.18.0.2.4.2437
NAME 'ibm-slapdSchemaAdditions'
DESC 'File path on ibmslapd host machine containing additional
schema definitions used by the LDCF backend.
Standard values are: /etc/V3.modifiedschema On Windows,
forward slashes are allowed, and a leading slash not
preceded by a drive letter is assumed to be rooted at the install directory
(i.e.: /etc/V3.system.at = D:\Program Files\IBM\ldap\etc\V3.system.at).'
EQUALITY 2.5.13.5
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2437
DBNAME('slapdSchemaAdditi' 'slapdSchemaAdditi')
ACCESS-CLASS normal
LENGTH 1024)

attributetypes=(1.3.18.0.2.4.2363
NAME 'ibm-slapdSchemaCheck'
DESC 'Must be one of { V2 | V3 | V3_lenient }. Specifies
schema checking mechanism for add/modify operation.
V2 = perform LDAP v2 checking. V3 = perform strict LDAP v3 checking.
V3_lenient = not ALL parent object classes are required.
Only the immediate object class is needed when adding entries.'
EQUALITY 2.5.13.2
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2363
DBNAME('SchemaCheck' 'SchemaCheck')
ACCESS-CLASS critical
LENGTH 10)

attributetypes=(1.3.18.0.2.4.2398
NAME 'ibm-slapdSecurePort'
DESC 'TCP/IP port used for SSL connections. Can not have the
same value as ibm-slapdPort.
(IP ports are unsigned, 16-bit integers in the range 1 - 65535)'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2398
DBNAME('SecurePort' 'SecurePort')
ACCESS-CLASS critical

```

LENGTH 5 )

```
attributetypes=(1.3.18.0.2.4.2399
NAME 'ibm-slapdSecurity'
DESC 'Must be one of { none | SSL | SSLOnly }.
Specifies types of connections accepted by server.
none - server listens on non-ssl port only.
ssl - server listens on both ssl and non-ssl ports.
sslonly - server listens on ssl port only.'
EQUALITY 2.5.13.2
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2399
DBNAME('Security' 'Security')
ACCESS-CLASS critical
LENGTH 7)
```

```
attributetypes=(1.3.18.0.2.4.3111
NAME 'ibm-slapdServerBackend'
DESC 'Specifies whether this server loads a
database or proxy backend.'
EQUALITY 2.5.13.2
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.3111
DBNAME('slapdServerBacken' 'slapdServerBacken')
ACCESS-CLASS critical
LENGTH 1000)
```

```
attributetypes=(1.3.18.0.2.4.2433
NAME 'ibm-slapdServerId'
DESC 'Identifies the server for use in replication'
EQUALITY 1.3.6.1.4.1.1466.109.114.1
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
SINGLE-VALUE
USAGE userApplications)
IBMAttributetypes=(1.3.18.0.2.4.2433
DBNAME('slapdServerId' 'slapdServerId')
ACCESS-CLASS normal
LENGTH 240)
```

```
attributetypes=(1.3.18.0.2.4.2397
NAME 'ibm-slapdSetenv'
DESC 'Server executes putenv() for all values of
ibm-slapdSetenv at startup to modify its own runtime environment.
Shell variables (%PATH% or %LANG%) will not be expanded.
The only current use for this attribute is to set
DB2CODEPAGE=1208, which is required if using
UCS-2 (Unicode) databases.'
EQUALITY 2.5.13.5
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2397
DBNAME('slapdSetenv' 'slapdSetenv')
ACCESS-CLASS critical
LENGTH 2000)
```

```
attributetypes=(1.3.18.0.2.4.2396
NAME 'ibm-slapdSizeLimit'
DESC 'Maximum number of entries to return from search,
regardless of any sizelimit that may have
been specified on the client search request.
Range = 0... If a client has passed a limit, then
the smaller value of the client value and the value
read from ibmslapd.conf will be used.
If a client has not passed a limit and has bound as
```

admin DN, then the limit will be considered unlimited. If the client has not passed a limit and has not bound as admin DN, then the limit will be that which was read from ibmslapd.conf file.  
0 = unlimited.'

```
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2396
DBNAME('SizeLimit' 'SizeLimit')
ACCESS-CLASS critical
LENGTH 11)
```

```
attributetypes=(1.3.18.0.2.4.2381
NAME 'ibm-slapdSortKeyLimit'
DESC 'Maximum number of sort conditions (keys) that
can be specified on a single search request.
Range = 0.... If a client has passed a search request with more
sort keys than the limit allows, and the sorted search
control criticality is FALSE, then the server will honor
the value read from ibmslapd.conf and ignore any
sort keys encountered after the limit has been
reached - searching and sorting will be performed.
If a client has passed a search a request with more keys than the
limit allows, and the sorted search control criticality
is TRUE, then the server will return to the client with
return code of adminLimitExceeded - no searching or sorting
will be performed.'
```

```
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2381
DBNAME('SlapdSortKeyLimit' 'SlapdSortKeyLimit')
ACCESS-CLASS critical
LENGTH 11)
```

```
attributetypes=(1.3.18.0.2.4.2377
NAME 'ibm-slapdSortSrchAllowNonAdmin'
DESC 'Whether or not the server should allow
non-Administrator bind for sort on a search request.
If the value read from the ibmslapd.conf file is TRUE,
the server will process any client request, including
those submitted by a user binding anonymously.
If the value read from the ibmslapd.conf file is FALSE,
the server will process only those client requests submitted
by a user with Administrator authority. If a
client requests sort with a criticality of TRUE for a
search operation, does not have Administrator authority,
and the value read from the ibmslapd.conf file for
this attribute is FALSE, the server will return to the
client with return code insufficientAccessRights - no
searching or sorting will be performed.'
```

```
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2377
DBNAME('SlapdSortNonAdmin' 'SlapdSortNonAdmin')
ACCESS-CLASS critical
LENGTH 5)
```

```
attributetypes=(1.3.18.0.2.4.2395
NAME 'ibm-slapdSslAuth'
DESC 'Must be one of { serverauth | serverclientauth }. Specify authentication
type for ssl connection. serverauth - supports server authentication at
the client. serverclientauth - supports both server and client authentication.'
EQUALITY 2.5.13.2
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
```

```

USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2395
DBNAME('slapdSslAuth' 'slapdSslAuth')
ACCESS-CLASS critical
LENGTH 16)

attributetypes=(1.3.18.0.2.4.2389
NAME 'ibm-slapdSslCertificate'
DESC 'Specify the label that identifies the servers Personal Certificate in
the key database file. This label is specified when the servers private
key and certificate are created with the ikmgui application.
If ibm-slapdSslCertificate is not defined, the default private key, as
defined in the key database file, is used by the LDAP server for SSL connections.'
EQUALITY 2.5.13.5
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2389
DBNAME('SslCertificate' 'SslCertificate')
ACCESS-CLASS critical
LENGTH 128)

attributetypes=(1.3.18.0.2.4.2429
NAME 'ibm-slapdSslCipherSpec'
DESC 'SSL Cipher Spec Value must be set to DES-56, RC2-40-MD5, RC4-128-MD5,
RC4-128-SHA, RC4-40-MD5, TripleDES-168, or AES'
EQUALITY 1.3.6.1.4.1.1466.109.114.1
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2429
DBNAME('slapdSslCipherSpe' 'slapdSslCipherSpe')
ACCESS-CLASS normal
LENGTH 30)

attributetypes=(1.3.18.0.2.4.2362
NAME 'ibm-slapdSslCipherSpecs'
DESC 'This attribute is deprecated in favor of ibm-slapdSslCipherSpec.
Specifies a decimal number which identifies the allowable
encryption/decryption methods for establishing a SSL connection between
LDAP clients and server. This number represents the availability of
the encryption/decryption methods supported by the LDAP server.
The pre-defined Cipher values and their descriptions are:
SLAPD_SSL_TRIPLE_DES_SHA_US 0x0A Triple DES encryption with a 168-bit key
and a SHA-1 MAC
SLAPD_SSL_DES_SHA_US 0x09 DES encryption with a 56-bit key and a SHA-1 MAC
SLAPD_SSL_RC4_SHA_US 0x05 RC4 encryption with a 128-bit key and a SHA-1 MAC
SLAPD_SSL_RC4_MD5_US 0x04 RC4 encryption with a 128-bit key and a MD5 MAC
SLAPD_SSL_RC4_MD5_EXPORT 0x03 RC4 encryption with a 40-bit key and a MD5 MAC
SLAPD_SSL_RC2_MD5_EXPORT 0x06 RC2 encryption with a 40-bit key and a MD5 MAC'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2362
DBNAME('SslCipherSpecs' 'SslCipherSpecs')
ACCESS-CLASS critical
LENGTH 11)

attributetypes=(1.3.18.0.2.4.3088
NAME 'ibm-slapdSslFIPsModeEnabled'
DESC 'Specifies server will use ICC version of GSKit if TRUE,
BSAFE version if false.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
SINGLE-VALUE USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.3088
DBNAME('slapdSslFIPsModeE' 'slapdSslFIPsModeE')
ACCESS-CLASS critical
LENGTH 5)

```

```

attributetypes=(1.3.18.0.2.4.2375
NAME 'ibm-slapdSSLKeyDatabase'
DESC 'File path to the LDAP servers SSL key database file. This key database
file is used for handling SSL connections from LDAP clients, as well as for
creating secure SSL connections to replica LDAP servers. On Windows, forward
slashes are allowed, and a leading slash not preceded by a drive specifier (D:)
is assumed to be rooted at the install directory
(i.e.: /etc/key.kdb = D:\Program Files\IBM\ldap\etc\key.kdb).'
```

```

EQUALITY 2.5.13.5
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
```

```

USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2375
DBNAME('slapdSSLKeyDataba' 'slapdSSLKeyDataba')
ACCESS-CLASS critical
LENGTH 1024)
```

```

attributetypes=(1.3.18.0.2.4.2438
NAME 'ibm-slapdSSLKeyDatabasePW'
DESC 'Specify the password associated with the LDAP servers SSL key database file,
as specified on the ibm-slapdSslKeyDatabase parameter. If the LDAP servers key
database file has an associated password stash file, then the
ibm-slapdSslKeyDatabasePW parameter can be omitted, or set to
ibm-slapdSslKeyDatabasePW = none. Note that the password stash file must be
located in the same directory as the key database file and it must have the same
file name as the key database file, but with an extension of .sth, instead of .kdb'
```

```

SYNTAX 1.3.6.1.4.1.1466.115.121.1.5
SINGLE-VALUE
```

```

USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2438
DBNAME('slapdSSLKeyDPW' 'slapdSSLKeyDPW')
ACCESS-CLASS normal)
```

```

attributetypes=(1.3.18.0.2.4.2392
NAME 'ibm-slapdSslKeyRingFile'
DESC 'file path to the LDAP servers SSL key database file. This key database
file is used for handling SSL connections from LDAP clients, as well as for
creating secure SSL connections to replica LDAP servers. On Windows, forward
slashes are allowed, and a leading slash not preceded by a drive specifier (D:)
is assumed to be rooted at the install directory (
i.e.: /etc/key.kdb = D:\Program Files\IBM\ldap\etc\key.kdb).'
```

```

EQUALITY 2.5.13.5
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE USAGE directoryOperation)
```

```

IBMAttributetypes=(1.3.18.0.2.4.2392
DBNAME('SslKeyRingFile' 'SslKeyRingFile')
ACCESS-CLASS critical
LENGTH 1024)
```

```

attributetypes=(1.3.18.0.2.4.2390
NAME 'ibm-slapdSslKeyRingFilePW'
DESC 'Specify the password associated with the LDAP servers SSL key database file,
as specified on the ibm-slapdSslKeyRingFile parameter. If the LDAP servers key
database file has an associated password stash file, then the
ibm-slapdSslKeyRingFilePW parameter can be omitted, or set to
ibm-slapdSslKeyRingFilePW = none. Note that the password stash file must be
located in the same directory as the key database file and it must have the same
file name as the key database file, but with an extension of .sth, instead of .kdb.'
```

```

SYNTAX 1.3.6.1.4.1.1466.115.121.1.5
SINGLE-VALUE
```

```

USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2390
DBNAME('SslKeyRingFilePW' 'SslKeyRingFilePW')
ACCESS-CLASS critical)
```

```

attributetypes=(1.3.18.0.2.4.3058
```

```

NAME 'ibm-slapdStartupTraceEnabled'
DESC 'Must be one of [TRUE|FALSE]. Specifies whether trace information is to
be collected at server startup.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.3058
DBNAME('slapdStartupTrace' 'slapdStartupTrace')
ACCESS-CLASS normal
LENGTH 5)

attributetypes=(1.3.18.0.2.4.2388
NAME 'ibm-slapdSuffix'
DESC 'Specifies a naming context to be stored in this backend.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2388
DBNAME('slapdSuffix' 'slapdSuffix')
ACCESS-CLASS critical
LENGTH 1000)

attributetypes=(1.3.18.0.2.4.2480
NAME 'ibm-slapdSupportedWebAdmVersion'
DESC 'This attribute defines the earliest version of the web admin that
supports this servers of cn=configuration.'
EQUALITY 2.5.13.2
ORDERING 2.5.13.3
SUBSTR 2.5.13.4
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2480
DBNAME('slapdSupWebAdmVer' 'slapdSupWebAdmVer')
ACCESS-CLASS normal
LENGTH 256)

attributetypes=(1.3.18.0.2.4.2393
NAME 'ibm-slapdSysLogLevel'
DESC 'Must be one of { l | m | h }. Level at which debugging and operation
statistics are logged in ibmslapd.log file. h - high (verbose),
m - medium, l - low (terse).'
EQUALITY 2.5.13.2
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2393
DBNAME('SysLogLevel' 'SysLogLevel')
ACCESS-CLASS critical
LENGTH 1)

attributetypes=(1.3.18.0.2.4.2391
NAME 'ibm-slapdTimeLimit'
DESC 'Maximum number of number of seconds to spend on search request, regardless
of any timelimit that may have been specified on the client request.
Range = 0.... If a client has passed a limit, then the smaller value of the
client value and the value read from ibmslapd.conf will be used. If a client
has not passed a limit and has bound as admin DN, then the limit will be
considered unlimited. If the client has not passed a limit and has not bound as
admin DN, then the limit will be that which was read from ibmslapd.conf file.
0 = unlimited.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2391
DBNAME('TimeLimit' 'TimeLimit')
ACCESS-CLASS critical
LENGTH 11)

```

```

attributetypes=(1.3.18.0.2.4.3060
NAME 'ibm-slapdTraceMessageLevel'
DESC 'Any value that would be acceptable after the ibmslapd -h command line
option, sets the Debug message level'
EQUALITY 2.5.13.2
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.3060
DBNAME('slapdTraceLevel' 'slapdTraceLevel')
ACCESS-CLASS normal
LENGTH 6)

attributetypes=(1.3.18.0.2.4.3059
NAME 'ibm-slapdTraceMessageLog'
DESC 'File path or device on server host machine to which LDAP CAPI
and Debug macro messages will be written. On Windows forward
slashes are allowed and a leading slash not preceded by a drive letter
is assumed to be rooted at the install directory
(i.e., /tmp/tracemsg.log = C:\Program Files\IBM\LDAP\tmp\tracemsg.log).'
EQUALITY 2.5.13.2
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.3059
DBNAME('slapdTraceMessage' 'slapdTraceMessage')
ACCESS-CLASS normal
LENGTH 1024)

attributetypes=(1.3.18.0.2.4.2384
NAME 'ibm-slapdTransactionEnable'
DESC 'If FALSE, globally disables transaction support; the server will reject
all StartTransaction requests with the response LDAP_UNWILLING_TO_PERFORM.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2384
DBNAME('TransactionEnable' 'TransactionEnable')
ACCESS-CLASS critical
LENGTH 5)

attributetypes=(1.3.18.0.2.4.2499
NAME 'ibm-slapdUseProcessIdPW'
DESC 'If set to true the server will use user login ID
associated with the ibmslapd process to connect to the
database. If set to false the server will use ibm-slapddbUserID
and ibm-slapddbUserPW to connect to the database.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2499
DBNAME('useprocidpw' 'useprocidpw')
ACCESS-CLASS normal
LENGTH 5)

attributetypes=(1.3.18.0.2.4.2436
NAME 'ibm-slapdVersion' DESC 'IBM Slapd version Number'
EQUALITY 2.5.13.5
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.2436
DBNAME('slapdVersion' 'slapdVersion')
ACCESS-CLASS normal
LENGTH 1024)

attributetypes=(1.3.18.0.2.4.3026

```

```

NAME 'ibm-slapdWriteTimeout'
DESC 'Specifies a time-out value for blocked writes. When the time limit is
reached the connection will be dropped.'
EQUALITY 2.5.13.14
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.3026
DBNAME('slapdWriteTimeout' 'slapdWriteTimeout')
ACCESS-CLASS normal
LENGTH 11)

attributetypes=(1.3.18.0.2.4.3110
NAME 'ids-instanceDesc'
DESC 'A description of what this particular directory server is to be used for.'
EQUALITY 2.5.13.5
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.3110
DBNAME('idsInstDesc' 'idsInstDesc')
ACCESS-CLASS critical
LENGTH 256)

attributetypes=(1.3.18.0.2.4.3132
NAME 'ids-instanceLocation'
DESC 'File path or device on server host machine to which the
directory server instance\27s idsslapd-<instance name>directory
is located. On Windows forward slashes are allowed and a leading
slash not preceded by a drive letter is assumed to be rooted
at the install directory
(i.e., /tmp/idsslapd-server1 = C:\Program Files\IBM\LDAP\tmp\idsslapd-server1).'
EQUALITY 2.5.13.5
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.3132
DBNAME('idsinstLoc' 'idsinstLoc')
ACCESS-CLASS critical
LENGTH 1024)

attributetypes=(1.3.18.0.2.4.3133
NAME 'ids-instanceVersion'
DESC 'IBM Slapd version Number for the directory server instance.'
EQUALITY 2.5.13.5
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.3133
DBNAME('idsinstVeri' 'idsinstVeri')
ACCESS-CLASS critical
LENGTH 1024)

attributetypes=(1.3.18.0.2.4.3154
NAME 'ibm-slapdSs1FIPsProcessingMode'
DESC 'Specifies server will operate in FIPS mode.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=(1.3.18.0.2.4.3154
DBNAME('slapdSs1FIPsPMode' 'slapdSs1FIPsPMode')
ACCESS-CLASS critical
LENGTH 5)

```

## Dynamically-changed attributes

The following is a list of attributes that can be changed dynamically. You do not have to restart the server for these changes to take effect.



### **Cn=Configuration**

- ibm-slapdadmindn
- ibm-slapdAdminGroupEnabled
- ibm-slapdadminpw
- ibm-slapdDerefAliases
- ibm-slapdpwencryption
- ibm-slapdsizelimit
- ibm-slapdtimelimit

### **cn=Log Management, cn=Configuration**

The dynamically-changed attributes apply to the following subentries:

- cn=Default, cn=Log Management, cn=Configuration
- cn=ibmslapd, cn=Log Management, cn=Configuration
- cn=Audit, cn=Log Management, cn=Configuration
- cn=Bulkload, cn=Log Management, cn=Configuration
- cn=DB2CLI, cn=Log Management, cn=Configuration
- cn=Tools, cn=Log Management, cn=Configuration
- cn=Replication, cn=Log Management, cn=Configuration
- cn=Admin, cn=Log Management, cn=Configuration
- cn=Admin Audit, cn=Log Management, cn=Configuration

The following are the dynamically-changed attributes for these subentries:

- ibm-slapdLog (Does not apply to cn=Default)
- ibm-slapdLogArchivePath
- ibm-slapdLogMaxArchives
- ibm-slapdLogOptions (Does not apply to cn=Default)
- ibm-slapdLogSizeThreshold

### **cn=AdminGroup, cn=Configuration**

These attributes are dynamically-changed for the subtrees under this entry.

- ibm-slapdAdminDN
- ibm-slapdAdminPW
- ibm-slapdDigestAdminUser
- ibm-slapdKrbAdminDN

### **cn=Front End, cn=Configuration**

- ibm-slapdaclcache
- ibm-slapdaclcachesize
- ibm-slapdentrycachesize
- ibm-slapdfiltercachebypasslimit
- ibm-slapdfiltercachesize
- ibm-slapdidletimeout

### **cn=Connection Management, cn=Front End, cn=Configuration**

- ibm-slapdAllowAnon
- ibm-slapdAllReapingThreshold
- ibm-slapdAnonReapingThreshold
- ibm-slapdBoundReapingThreshold
- ibm-slapdESizeThreshold

- ibm-slapdEThreadActivate
- ibm-slapdEThreadEnable
- ibm-slapdETimeThreshold
- ibm-slapdIdleTimeOut
- ibm-slapdWriteTimeout

**cn=Event Notification, cn=Configuration**

- ibm-slapdmaxeventsperconnection
- ibm-slapdmaxeventstotal

**cn=Transaction, cn=Configuration**

- ibm-slapdmaxnumoftransactions
- ibm-slapdmaxoppertransaction
- ibm-slapdmaxtimelimitoftransactions

**cn=ConfigDB, cn=Config Backends, cn=IBM Directory, cn=Schemas, cn=Configuration**

- ibm-slapdreadonly

**cn=Directory, cn=RDBM Backends, cn=IBM Directory, cn=Schemas, cn=Configuration**

- ibm-slapdCachedAttribute
- ibm-slapdCachedAttributeSize
- ibm-slapdLanguageTagsEnabled
- ibm-slapdpagedresallownonadmin
- ibm-slapdpagedreslmt
- ibm-slapdreadonly
- ibm-slapdsortkeylimit
- ibm-slapdsortsrchallownonadmin
- ibm-slapdsuffix

**cn=change log, cn=RDBM Backends, cn=IBM Directory, cn=Schemas, cn=Configuration**

- ibm-slapdCachedAttribute
- ibm-slapdCachedAttributeSize

**cn=Digest, cn=configuration**

- ibm-slapdDigestAdminUser
- ibm-slapdDigestRealm
- ibm-slapdDigestAttr

**cn=pwdPolicy Admin, cn=Configuration**

- ibm-slapdConfigPwdPolicyOn
- pwdMinLength
- pwdLockout
- pwdLockoutDuration
- pwdMaxFailure
- pwdFailureCountInterval
- passwordMinAlphaChars
- passwordMinOtherChars
- passwordMaxRepeatedChars

- passwordMinDiffChars
- cn=Replication, cn=configuration**
- ibm-slapdReplContextCacheSize



---

## Appendix L. Audit format

The following are format descriptions for server audits and admin daemon audits.

---

### Audit format for a server audit

Following the header, any control information is audited.

#### Format of control information

- controlType: OID
- criticality: TRUE or FALSE

If the audit version is set to 1, no additional information is audited.

If the audit version is set to 2 or greater, then the following is TRUE:

- If the control is a Proxy authorization control, then the following additional information is audited:
  - ProxyDN: Proxy Auth DN
- If the control is a Group authorization control, and audit is configured to audit the groups sent on a Group authorization control, then the following additional information is audited:
  - Group: Group Name
  - Group: Group Name 2 (repeat for each group)
  - Normalized: TRUE or FALSE
- If the control is an Audit control, and audit is configured to audit the additional information in the Audit control, then the following additional information is audited:
  - RequestID: request ID 1
  - RequestID: request ID 2 (repeat for each additional request ID)
  - ClientIP: client IP sent in the audit control
- If the control is a Replication update ID control, and audit is configured to audit the Replication update ID control, then the following additional information is audited:
  - value: value sent in the control

As well as the previous control information, the following operation-specific data is audited:

- Bind:
  - name: *<bindDN string>*
  - authenticationChoice: unknown, simple, krbv42LDAP, krbv42DSA, sasl
  - authenticationMechanism: CRAM-MD5
  - Admin Acct Status: Not Locked, Locked, or Lock Cleared
  - username: adminusername (for DIGEST-MD5 only)
  - mappedname: cn=root (for DIGEST-MD5 w/ authzid only)
  - authzId: u: username (for DIGEST-MD5 with authzid only)
- Search:
  - base: o=ibm\_us, c=us

- scope: unknown, baseObject, singleLevel, or wholeSubtree
- derefAliases: unknown, neverDerefAliases, derefInSearching, derefFindingBaseObj, or derefAlways
- typesOnly: FALSE
- filter: (&(cn=c\*)(sn=a\*))
- attributes: cn, sn, title (this item is not present if there are no attributes)
- Compare:
  - entry: cn=Joe Smith, o=ibm\_us, c=us
  - attribute: cn

**Note:** The attribute value is not written.

- Add:
  - entry: cn=Joe Smith, o=ibm\_us, c=us
  - attributes: cn, sn

**Note:** The attribute value is not written.

- Modify:
  - object: cn=Joe Smith, o=ibm\_us, c=us
  - add: mail
  - delete: title
  - replace: telphonenumber (repeat for each operation/attribute pair)

Modify types can be one of the following:

- unknown
- add
- delete
- replace
- Delete:
  - entry: cn=Joe Smith, o=ibm\_us, c=us
- ModifyDN:
  - entry: cn=Joe Smith, ou=Austin, o=ibm\_us, c=us
  - newrdn: Joe S. Smith
  - deleteoldrdn: true
  - newSuperior: ou=rochester (this item is not present if there is no newSuperior value)
- Event Notification: Event Registration:
  - eventID: LDAP\_change
  - base: o=ibm\_us, c=us
  - scope: wholeSubtree
  - type: unknown, changeAdd, changeDelete, changeModify, or changeModDN
- Event Notification: Unregistered Event:
  - ID: hostname.uuid

For all extended operations other than event notification, the OID is audited. Some extended operations also audit additional information.

#### **Format of the OID**

OID: OID

For more information about the auditing features for a specific extended operation, see "Appendix F. Object Identifiers (OIDs) for extended operations and controls" in *IBM Tivoli Directory Server C-Client SDK Programming Reference Version 6.0*.

## Auditing server events

The following server events are audited if auditing is enabled:

- Auditing started
- Audited stopped
- Audit configuration changed
- Server started
- Server stopped

Server events are audited in the following format:

*<Time>--<Message Text in local code page>*

For example:

```
2005-01-05-14:06:20.957-06:00--GLPSRV009I IBM Tivoli Directory (SSL),
Version 6.0 Server started.
```

## Notes

- All DNs are audited in local code page.
- If the `SLAPD_AUDIT_ENCODE_DN` is set to any value, the BindDN and Digest Bind DNs are encoded when written to the audit log. To decode the DNs, an administrator can perform a search against the server with the scope base, base NULL and a filter of `ibm-auditdecodeddn= <value to decode>`. For example:  

```
idsldapsearch -D adminDN -w password -s base -b " " ibm-auditdecodeddn=encoded DN
```

---

## Audit format for an Admin Daemon audit

Header format is the same except for on an admin bind where one of the following strings appear at the end of the header:

- Admin Acct Status: Not Locked
- Admin Acct Status: Locked
- Admin Acct Status: Lock Cleared

The Operation Types audited are one of the following:

- Bind
- Unbind
- Search
- Extended operation

**Note:** This is the format of the string that appears in the header for the operation.

The Audit control is audited in the Admin Daemon in the same format as it is audited in the server. No additional information is audited on binds, unbinds and searches. Additional information is audited in extended operations:

- Start, stop server extended operations:
  - OID: OID
  - Operation: Start, Restart, Status, Admin Stop, Stop, Unknown
  - Options: options passed in (only on start and restart)
- All other extended operations:

- OID: OID



---

## Appendix M. Distributed directory setup tool options

The following are detailed descriptions of the options to specify in the header section of the configuration file:

| Name                           | Required | Multi-valued | Description                                                                                                                                                              |
|--------------------------------|----------|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Charset                        | N        | N            | Character set used in the file. If not specified, the local codepage is used.                                                                                            |
| LogFile                        | N        | N            | Specifies a file to log messages to.                                                                                                                                     |
| ActionType                     | Y        | N            | Specifies which action the tool should take. The only possible value is SplitOnly.                                                                                       |
| SplitBy                        | Y        | N            | Specifies the criteria on which to split the data. The only possible values in this release are NumServers or RDNHash, which are equivalent.                             |
| InputFile                      | N        | N            | Specifies an input file for the tool to use.                                                                                                                             |
| DefaultOutputFile              | N        | N            | Specifies a default output file. Entries go to this file if they are not children of a BaseDN.                                                                           |
| FileDirectory or BaseDirectory | N        | N            | Specifies a base directory for the tool to use for accessing the existing files and the generated files. If a directory is not specified, the current directory is used. |

The following are detailed descriptions of the options to specify in the BaseDN section of the configuration file:

| Name   | Required | Multi-valued | Description                                                                                                     |
|--------|----------|--------------|-----------------------------------------------------------------------------------------------------------------|
| BaseDN | Y        | N            | Specifies the BaseDN to split on. Children of this DN are moved into different files based on their hash value. |

|      |   |   |                                                                                                                                         |
|------|---|---|-----------------------------------------------------------------------------------------------------------------------------------------|
| URL  | Y | Y | Specifies a URL for an LDAP server. The value is used to generate filters and should contain the hostname and port for the LDAP server. |
| File | Y | Y | Specifies a File for an LDAP server. All of the server's entries are put into this file.                                                |

---

## Appendix N. Setting up SSL security – SSL scenarios

The scenarios presented in this appendix are designed to create secure connections between the different components of your IBM Tivoli Directory Server system.

The following conditions are assumed:

- IBM Tivoli Directory Server 6.0 is installed on a machine.
- An IBM Tivoli Directory Server instance is created.
- An IBM Tivoli Directory Server database is created.
- There are no key database (.kdb) or key store (.jks) files created.

---

### Using HTTPS for the embedded version of WebSphere Application Server Version 5.1.1

The embedded version of WebSphere Application Server, Version 5.1.1, by default, has HTTPS set up on port 12101. To use HTTPS, you must change your login Web address to the following:

```
https://<hostname>:12101/IDSWebApp/IDSjsp/Login.jsp
```

For non-HTTPS connections, continue to use the following Web address:

```
http://<hostname>:12100/IDSWebApp/IDSjsp/Login.jsp
```

Additionally, if you want to change the application server's SSL certificate, you can create new key and trust store database files for the WebSphere Application Server to use. By default, the key and trust store database files are separate and are located in the <WASHOME>/etc directory. These files are named **DummyServerKeyFile.jks** and **DummyServerTrustFile.jks** respectively.

After you have created your new jks files, you can change the key and trust store database files that IBM WebSphere Application Server uses by modifying the following items (highlighted in **bold**) in the <WASHOME>/config/cells/DefaultNode/security.xml file to use your new file names, passwords, and file formats:

```
<repertoire xmi:id="SSLConfig_1" alias="DefaultNode/DefaultSSLSettings"
 <setting xmi:id="SecureSocketLayer_1"
 keyFileName="{USER_INSTALL_ROOT}/etc/DummyServerKeyFile.jks"
 keyFilePassword="{xor}CDo9Hgw="
 keyFileFormat="JKS"
 trustFileName="{USER_INSTALL_ROOT}/etc/DummyServerTrustFile.jks"
 trustFilePassword="{xor}CDo9Hgw="
 trustFileFormat="JKS"
 clientAuthentication="false" securityLevel="HIGH"
 enableCryptoHardwareSupport="false">
 <cryptoHardware xmi:id="CryptoHardwareToken_1" tokenType=""
 libraryFile="" password=""/>
 <properties xmi:id="Property_4" name="com.ibm.ssl.protocol" value="SSLv3"/>
 <properties xmi:id="Property_5" name="com.ibm.ssl.contextProvider"
 value="IBMJSSE"/>
</setting>
</repertoire>
```

---

## Creating secure connections between IBM WebSphere Application Server, and the IBM Tivoli Directory server and the administration daemon

Create a key pair and certificate request for self-signing key store file (.jks) and a key database file (.kdb).

### Notes:

1. See "Appendix O. Setting up GSKit to support CMS key databases" in *IBM Tivoli Directory Server Version 6.0 Installation and Configuration Guide*.
2. The instructions for creating a key pair and certificate request for self-signing key store file (.jks) and a key database file (.kdb) are given based on the assumption that no key database or key store files have been created. If you already have key database or key store files created that you prefer to use, you can skip to step 5 on page 592.

The only requirements are that you create the keystore file and the key database file on a machine that has GSKit and Java installed:

**Note:** There can be only one key store file (.jks) per Web Application Server. You can request one of the following certificates:

- A low assurance certificate from VeriSign, best for non-commercial purposes, such as a beta test of your secure environment
- A server certificate to do commercial business on the Internet from VeriSign or some other CA
- A self-signed server certificate if you plan to act as your own CA for a private Web network

For information about using a CA such as VeriSign to sign the server certificate, see "Creating a key pair and requesting a certificate from a Certificate Authority" on page 122.

1. Do the following to create a key database (.kdb) file:
  - a. Type `gsk7ikm` to start the Java utility.
  - b. Select **Key Database File**.
  - c. Select **New**, or **Open** if the key database already exists.
  - d. Specify a key database file name and location. Click **OK**.

**Note:** A key database is a file that the client or server uses to store one or more key pairs and certificates.

- e. When prompted, supply the password for the key database file. Click **OK**.
- f. Go to **Create->New Self-Signed Certificate**.
- g. Supply the following:
  - User-assigned label for the key pair. The label identifies the key pair and certificate in the key database file.

**Note:** Remember this label.

- The desired certificate Version.
- The desired Key Size.
- The X.500 common name of the server. Usually this is the TCP/IP fully qualified host name, for example, `www.ibm.com`.
- The organization name. This is the name of your organization.
- The organizational unit name. This is an optional field.

- The locality/city where the server is located. This is an optional field.
  - A three-character abbreviation of the state/province where the server is located. This is an optional field.
  - The zip code appropriate for the server's location. This is an optional field.
  - The two-character country code where the server is located.
  - The Validity Period for the certificate.
- h. Click **OK**.
2. Do the following to create a self-signing key store file (.jks):

**Note:** The .jks file should not have the same name as the .kdb file if they are stored in the same directory.

- a. Type `gsk7ikm` to start the Java utility.
- b. Select **Key Database File**.
- c. Select **New**, or **Open** if the key database already exists.

**Note:** Specify **JKS** in the **Key database type** list.

- d. Specify a key store file name and location. Click **OK**.
  - e. When prompted, supply the password for the key store file. Click **OK**.
  - f. Go to **Create->New Self-Signed Certificate**.
  - g. Supply the following:
    - User-assigned label for the key pair. The label identifies the key pair and certificate in the key database file.
    - The desired certificate Version.
    - The desired Key Size.
    - The X.500 common name of the server. Usually this is the TCP/IP fully qualified host name, for example, `www.ibm.com`.
    - The organization name. This is the name of your organization.
    - The organizational unit name. This is an optional field.
    - The locality/city where the server is located. This is an optional field.
    - A three-character abbreviation of the state/province where the server is located. This is an optional field.
    - The zip code appropriate for the server's location. This is an optional field.
    - The two-character country code where the server is located.
    - The Validity Period for the certificate.
  - h. Click **OK**.
3. Extract the certificate from the .kdb file to the .jks file:
- a. Select **Key Database File**.
  - b. Select **Open**.
  - c. Select the key database (.kdb) file name and location.

**Note:** This is the key database file you created previously.

- d. If asked, provide password.
- e. Click **OK**.
- f. Go to **Personal certificates**.
- g. Click **Extract Certificate**.

- h. Select **Data type**. For this scenario, select **Binary DER data**.
- i. Provide a filename and location.

**Note:** Remember this filename and location.

- j. Select **Key Database File**.
- k. Select **Open**.
- l. Select the key store (.jks) file name and location.

**Note:** This is the key store file you created previously.

- m. If asked, provide password.
  - n. Click **OK**.
  - o. Go to **Signer certificates**.
  - p. Click **Add**.
  - q. Select the **Binary DER data** (.der) file created previously.
  - r. Click **OK**.
  - s. Enter a label for this certificate.
  - t. Click **OK**.
4. Extract the certificate from the .jks file to the .kdb file:
    - a. Select **Key Database File**.
    - b. Select **Open**.
    - c. Select the key store (.jks) file name and location.

**Note:** This is the key store file you created previously.

- d. If asked, provide password.
- e. Click **OK**.
- f. Go to **Personal certificates**.
- g. Click **Extract Certificate**.
- h. Select **Data type**. For this scenario, select **Binary DER data**.
- i. Provide a filename and location.

**Note:** Remember this filename and location.

- j. Select **Key Database File**.
- k. Select **Open**.
- l. Select the key database (.kdb) file name and location.

**Note:** This is the key database file you created previously.

- m. If asked, provide password.
  - n. Click **OK**.
  - o. Go to **Signer certificates**.
  - p. Click **Add**.
  - q. Select the **Binary DER data** (.der) file created previously.
  - r. Click **OK**.
  - s. Enter a label for this certificate.
  - t. Click **OK**.
5. Start the directory server instance, if not started already. See "Starting the directory server instance" in *IBM Tivoli Directory Server Version 6.0 Installation and Configuration Guide* .

6. Start the application server. See "Starting the application server to use the Web Administration Tool" in *IBM Tivoli Directory Server Version 6.0 Installation and Configuration Guide*.
7. Log on to the Web Administration Tool to add a non-SSL-enabled server. Launch the Web Administration Tool:
  - a. After you have started the application server, from a Web browser, type the following address:  
 http://localhost:12100/IDSWebApp/IDSjsp/Login.jsp  
 The IBM Tivoli Directory Server Web Administration Tool Login page is displayed.
 

**Note:** This address works only if you are running the browser on the computer on which the Web Administration Tool is installed. If the Web Administration Tool is installed on a different computer, replace **localhost** with the hostname or IP address of the computer where the Web Administration Tool is installed.
  - b. Log in to the console as the console administrator:
    - 1) Be sure that **Console Admin** is displayed in the **LDAP Hostname** field.
    - 2) In the **Username** field, type superadmin.
    - 3) In the **Password** field, type secret.
 The IBM Tivoli Directory Server Web Administration Tool console is displayed.
  - c. Add a non-SSL-enabled server to the console, using the following instructions:
    - 1) Expand **Console administration** in the navigation area.
    - 2) Click **Manage console servers**. A table of server host names and port numbers is displayed.
    - 3) Click **Add**.
    - 4) Type the hostname or the IP address of the server in the **Hostname** field; for example, myserver.mycity.mycompany.com
    - 5) Specify the server port number in the **Port** field and the Admin daemon port number in the **Administration port** field. You can accept the defaults.
    - 6) Ensure the **Enable SSL encryption** check box is not checked.
    - 7) Click **OK**, and then click **OK** again on the confirmation panel.
  - d. Click **Logout** in the navigation area.
8. Log in as the directory server instance administrator:
  - a. On the IBM Tivoli Directory Server Web Administration Login Tool page, select the LDAP host name or IP address for your computer from the drop-down menu for the **LDAP Hostname** field.
  - b. Type the administrator DN and the password for the directory server instance. You specified these fields during instance creation.
  - c. Click **Login**.
9. Configure the security settings for the Web Administration console:
  - a. Go to the Web Administration console.
  - b. Click **Server administration**.
  - c. Click **Manage security properties**.
  - d. Click **Settings**.
  - e. To enable an SSL connection, select the **SSL** radio button.

**Note:** The security settings you set for the IBM Tivoli Directory Server here apply to the directory administration daemon as well.

- f. Select the **Server and client authentication** radio button.

**Note:** You must distribute the server certificate to each client. For server and client authentication you also must add the certificate for each client to the server's key database.

- g. Select the **Key database** tab:

- 1) Specify the **Key database path and file name**. This is the fully qualified file specification of the key database file. If a password stash file is defined, it is assumed to have the same file specification, with an extension of **.sth**.
- 2) Specify the **Key password**. If a password stash file is not being used, the password for the key database file must be specified here. Then specify the password again in the **Confirm password** field.
- 3) Specify the **Key label**. This administrator-defined key label indicates what part of the key database to use.

**Note:** In order for the server to use this file, it must be readable by the user ID **idsldap**. See the *IBM Tivoli Directory Server version 6.0 Problem Determination Guide* for information about file permissions.

- h. When you are finished, do one of the following:

- Click **Apply** to save your changes without exiting.
- Click **OK** to apply your changes and exit.
- Click **Cancel** to exit this panel without making any changes.

- i. You must stop and restart both the IBM Tivoli Directory Server and the administration daemon for the changes to take effect.

10. Configure the console properties settings for the Web Administration console:

- a. After you have restarted the application server, log in to the console as the console administrator:

- 1) Be sure that **Console Admin** is displayed in the **LDAP Hostname** field.
- 2) In the **Username** field, type superadmin.
- 3) In the **Password** field, type secret.

- b. Expand **Console administration** in the navigation area.

- c. Click **Manage console properties**.

- d. Click **Component management** to specify the components that are enabled for all servers in the console. By default all the components are enabled.

**Note:** You might not see a management component or some of its tasks, even if it is enabled, if you do not have the correct authority on the server or the server does not have the needed capabilities, or both.

- e. Click **Session properties** to set the time out limit for the console session. The default setting is 60 minutes.

**Note:** A session might be valid for three to five minutes more than what you have set. This is because the invalidations are performed by a background thread in the application server that acts on a timer interval. This timer interval extends the session time out duration.

- f. Click **SSL key database** to set up the console so that it can communicate with other LDAP servers using the Secure Sockets Layer (SSL), if necessary.



Set the key database path and file name, the key password, the trusted database path and file name, the trusted password in the appropriate fields.

**Note:** The supported file type is jks. Use the .jks file you created previously.

See “Using gsk7ikm” on page 121 and “Secure Sockets Layer” on page 115 for information about key databases and SSL.

**Note:** The LDAP server and the administration daemon can have separate credentials (key database files).

g. Click **OK**.

11. Add an SSL-enabled server to the console:

a. Expand **Console administration** in the navigation area.

b. Click **Manage console servers**.

c. Click **Add**.

d. Type the hostname or the IP address of the server in the **Hostname** field; for example, myserver.mycity.mycompany.com

e. Specify the server secure port number in the **Port** field and the Administration daemon secure port number in the **Administration port** field.

**Note:** The Port number and Administration port numbers are different for an SSL-enabled server. Click **Help** for more information.

f. Select the **Enable SSL encryption** check box.

g. Click **OK**, and then click **OK** again on the confirmation panel.

h. Click **Logout** in the navigation area.

**Note:** You might need to restart the IBM WebSphere Application Server.

12. Log in as the directory server instance administrator to verify that the SSL-enabled server was added correctly:

a. On the IBM Tivoli Directory Server Web Administration Login Tool page, select the LDAP host name or IP address for your computer from the drop-down menu for the **LDAP Hostname** field.

b. Type the administrator DN and the password for the directory server instance. You specified these fields during instance creation.

c. Click **Login**.

13. Configure the SSL-enabled localhost as SSL only-enabled:

a. Go to the Web Administration console.

b. Click **Server administration**.

c. Click **Manage security properties**.

d. Click **Settings**.

e. To enable an SSL connection, select the **SSL only** radio button.

f. Select the **Server and client authentication** radio button.

**Note:** You must distribute the server certificate to each client. For server and client authentication you also must add the certificate for each client to the server’s key database.

g. When you are finished, click **Apply** to save your changes without exiting. Click **OK** to apply your changes and exit. Click **Cancel** to exit this panel without making any changes.

- h. You must stop and restart both the IBM Tivoli Directory Server and the administration daemon for the changes to take effect.
14. Issue the following command to verify that the server is functioning as an SSL server:
- ```
idsldapsearch -D <admin_dn> -w <admin_pw> -Z -K <server_kdb_file>
-P <keyfile_password> -b "cn=localhost"
-p <server_secure_port> objectclass=*
```

Setting up an SSL connection between a client and server

1. Do the following to create a key database (.kdb) file and self-signed certificate on the server:
 - a. Type `gsk7ikm` to start the Java utility.
 - b. Select **Key Database File**.
 - c. Select **New**, or **Open** if the key database already exists.
 - d. Specify a key database file name and location, for example, `<server_file>.kdb`. Click **OK**.
 - e. When prompted, supply the password for the key database file.
 - f. Make sure the **Stash a password to a file?** box is checked.
 - g. Click **OK**.
 - h. Go to **Create->New Self-Signed Certificate**.
 - i. Supply the following:
 - User-assigned label for the key pair. The label identifies the key pair and certificate in the key database file.

Note: Remember this label.
 - The desired certificate Version.
 - The desired Key Size.
 - The X.500 common name of the server. Usually this is the TCP/IP fully qualified host name, for example, `www.ibm.com`.
 - The organization name. This is the name of your organization.
 - The organizational unit name. This is an optional field.
 - The locality/city where the server is located. This is an optional field.
 - A three-character abbreviation of the state/province where the server is located. This is an optional field.
 - The zip code appropriate for the server's location. This is an optional field.
 - The two-character country code where the server is located.
 - The Validity Period for the certificate.
2. Do the following to create a new .kdb file on the client machine:
 - a. Type `gsk7ikm` to start the Java utility.
 - b. Select **Key Database File**.
 - c. Select **New**, or **Open** if the key database already exists.
 - d. Specify a key database file name and location, for example, `<client_file>.kdb`. Click **OK**.
 - e. When prompted, supply the password for the key database file.
 - f. Make sure the **Stash a password to a file?** box is checked.
 - g. Click **OK**.
3. Do the following on the server machine (in the GSKit utility):

- a. Open the <server_file>.kdb file.
- b. Go to **Personal certificates**.
- c. Click **Extract Certificate**.
- d. Provide a filename and location.

Note: Remember this filename and location.

4. Go to a command prompt on the server machine:
 - a. Go to the directory to which you extracted the server self-signed certificate in the previous step.
 - b. FTP the server self-signed certificate to the client machine.
5. Do the following on the client machine (in the GSKit utility):
 - a. Open the <client_file>.kdb file.
 - b. Go to **Signer certificates**.
 - c. Click **Add**.
 - d. Click **Browse** to find the server self-signed certificate you added to the client machine in the previous step.
 - e. Open the file.
 - f. Click **OK**.
 - g. Enter the label for this certificate.

Note: This label must match the label you defined in step 596.

- h. Click **View/Edit**. Make sure the **Set the certificate as a trusted root** box is checked.
- i. Go to **Create->New Self-Signed Certificate**.
- j. Supply the following:
 - User-assigned label for the key pair. The label identifies the key pair and certificate in the key database file.
- k. Click **OK**.
- l. Click **Extract Certificate**.
- m. Provide a filename and location.

Note: Remember this filename and location.

- n. Click **OK**.

6. Go to a command prompt on the client machine:
 - a. Go to the directory to which you extracted the client self-signed certificate in the previous step.
 - b. FTP the server self-signed certificate to the server machine.
7. Do the following on the server machine (in the GSKit utility):
 - a. Open the `<server_file>.kdb` file.
 - b. Go to **Signer certificates**.
 - c. Click **Add**.
 - d. Click **Browse** to find the client self-signed certificate you added to the server machine in the previous step.
 - e. Open the file.
 - f. Click **OK**.
 - g. Enter the label for this certificate.

Note: This label must match the label you defined in step 597.

- h. Click **View/Edit**. Make sure the **Set the certificate as a trusted root** box is checked.
8. Issue the following command, from either the client or the server, to modify the `cn=SSL,cn=Configuration` entry in the `ibmslapd.conf` file:

```
idsldapmodify -D <admin_dn> -w <admin_pw> -i <filename>
```

where `<filename>` contains:

```
dn: cn=SSL,cn=Configuration
changetype: modify
replace: ibm-slapdSslAuth
ibm-slapdSslAuth: serverClientAuth
-
replace: ibm-slapdSecurity
ibm-slapdSecurity: SSLOnly
-
replace: ibm-slapdSSLKeyDatabase
ibm-slapdSSLKeyDatabase: <server_keyfile>
-
replace: ibm-slapdSSLKeyDatabasePW
ibm-slapdSSLKeyDatabasePW: <server_keyfile_password>
-
replace: ibm-slapdSslKeyRingFilePW
ibm-slapdSslKeyRingFilePW: <server_keyfile_password>
```

9. Restart the server and the administration daemon for the changes to take effect.
10. Issue the following command, from either the client or the server, to verify that the server is functioning as an SSL server:

```
idsldapsearch -D <admin_dn> -w <admin_pw> -K <keyfile>
-b "cn=localhost" -p <server_secure_port> objectclass=*
```

Notes:

- a. You do not need to specify the `-P` option here because the keyfile password was attached to a stash file.
- b. If issuing this command from a client, you must use the `-h` option, for example:

```
idsldapsearch -D <admin_dn> -w <admin_pw> -h <hostname> -K <keyfile>
-b "cn=localhost" -p <server_secure_port> objectclass=*
```

Appendix O. Support information

This section describes the following options for obtaining support for IBM products:

- “Searching knowledge bases”
- “Obtaining fixes”
- “Contacting IBM Software Support” on page 600

Searching knowledge bases

If you have a problem with your IBM software, you want it resolved quickly. Begin by searching the available knowledge bases to determine whether the resolution to your problem is already documented.

Search the information center on your local system or network

IBM provides extensive documentation that can be installed on your local computer or on an intranet server. You can use the search function of this information center to query conceptual information, instructions for completing tasks, reference information, and support documents.

Search the Internet

If you cannot find an answer to your question in the information center, search the Internet for the latest, most complete information that might help you resolve your problem. To search multiple Internet resources for your product, expand the product folder in the navigation frame to the left and select **Web search**. From this topic, you can search a variety of resources including:

- IBM technotes
- IBM downloads
- IBM Redbooks™
- IBM developerWorks®
- Forums and newsgroups
- Google

Obtaining fixes

A product fix might be available to resolve your problem. You can determine what fixes are available for your IBM software product by checking the product support Web site:

1. Go to the IBM Software Support Web site (<http://www.ibm.com/software/support>).
2. Under **Products A - Z**, select your product name. This opens a product-specific support site.
3. Under **Self help**, follow the link to **All Updates**, where you will find a list of fixes, fix packs, and other service updates for your product. For tips on refining your search, click **Search tips**.
4. Click the name of a fix to read the description and optionally download the fix.

To receive weekly e-mail notifications about fixes and other news about IBM products, follow these steps:

1. From the support page for any IBM product, click **My support** in the upper-right corner of the page.
2. If you have already registered, skip to the next step. If you have not registered, click register in the upper-right corner of the support page to establish your user ID and password.
3. Sign in to **My support**.
4. On the My support page, click **Edit profiles** in the left navigation pane, and scroll to **Select Mail Preferences**. Select a product family and check the appropriate boxes for the type of information you want.
5. Click **Submit**.
6. For e-mail notification for other products, repeat Steps 4 and 5.

For more information about types of fixes, see the *Software Support Handbook* (<http://techsupport.services.ibm.com/guides/handbook.html>).

Contacting IBM Software Support

IBM Software Support provides assistance with product defects.

Before contacting IBM Software Support, your company must have an active IBM software maintenance contract, and you must be authorized to submit problems to IBM. The type of software maintenance contract that you need depends on the type of product you have:

- For IBM distributed software products (including, but not limited to, Tivoli, Lotus, and Rational® products, as well as DB2 and WebSphere products that run on Windows or UNIX operating systems), enroll in Passport Advantage in one of the following ways:
 - **Online:** Go to the Passport Advantage Web page (http://www.lotus.com/services/passport.nsf/WebDocs/Passport_Advantage_Home) and click **How to Enroll**
 - **By phone:** For the phone number to call in your country, go to the IBM Software Support Web site (<http://techsupport.services.ibm.com/guides/contacts.html>) and click the name of your geographic region.
- For IBM eServer™ software products (including, but not limited to, DB2 and WebSphere products that run in zSeries®, pSeries™, and iSeries™ environments), you can purchase a software maintenance agreement by working directly with an IBM sales representative or an IBM Business Partner. For more information about support for eServer software products, go to the IBM Technical Support Advantage Web page (<http://www.ibm.com/servers/eserver/techsupport.html>).

If you are not sure what type of software maintenance contract you need, call 1-800-IBMSERV (1-800-426-7378) in the United States or, from other countries, go to the contacts page of the IBM Software Support Handbook on the Web (<http://techsupport.services.ibm.com/guides/contacts.html>) and click the name of your geographic region for phone numbers of people who provide support for your location.

Follow the steps in this topic to contact IBM Software Support:

1. Determine the business impact of your problem.
2. Describe your problem and gather background information.

3. Submit your problem to IBM Software Support.

Determine the business impact of your problem

When you report a problem to IBM, you are asked to supply a severity level. Therefore, you need to understand and assess the business impact of the problem you are reporting. Use the following criteria:

Severity 1	Critical business impact: You are unable to use the program, resulting in a critical impact on operations. This condition requires an immediate solution.
Severity 2	Significant business impact: The program is usable but is severely limited.
Severity 3	Some business impact: The program is usable with less significant features (not critical to operations) unavailable.
Severity 4	Minimal business impact: The problem causes little impact on operations, or a reasonable circumvention to the problem has been implemented.

Describe your problem and gather background information

When explaining a problem to IBM, be as specific as possible. Include all relevant background information so that IBM Software Support specialists can help you solve the problem efficiently. To save time, know the answers to these questions:

- What software versions were you running when the problem occurred?
- Do you have logs, traces, and messages that are related to the problem symptoms? IBM Software Support is likely to ask for this information.
- Can the problem be re-created? If so, what steps led to the failure?
- Have any changes been made to the system? (For example, hardware, operating system, networking software, and so on.)
- Are you currently using a workaround for this problem? If so, please be prepared to explain it when you report the problem.

Submit your problem to IBM Software Support

You can submit your problem in one of two ways:

- **Online:** Go to the "Submit and track problems" page on the IBM Software Support site (<http://www.ibm.com/software/support/probsub.html>). Enter your information into the appropriate problem submission tool.
- **By phone:** For the phone number to call in your country, go to the contacts page of the IBM Software Support Handbook on the Web (techsupport.services.ibm.com/guides/contacts.html) and click the name of your geographic region.

If the problem you submit is for a software defect or for missing or inaccurate documentation, IBM Software Support creates an Authorized Program Analysis Report (APAR). The APAR describes the problem in detail. Whenever possible, IBM Software Support provides a workaround for you to implement until the APAR is resolved and a fix is delivered. IBM publishes resolved APARs on the IBM product support Web pages daily, so that other users who experience the same problem can benefit from the same resolutions.

For more information about problem resolution, see Searching knowledge bases and Obtaining fixes.

Appendix P. 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 LZKS
11400 Burnet Road
Austin, TX 78758
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

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

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, or other countries, or both:

AIX	Lotus	SecureWay
DB2	Passport Advantage	Tivoli
developerWorks	pSeries	WebSphere
eServer	RACF	World Registry
IBM	Rational	z/OS
iSeries	Redbooks	zSeries

Java is a registered trademark of Sun Microsystems, Inc..

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

Windows is a registered trademark of Microsoft® Corporation

UNIX is a registered trademark of The Open Group.

Other company, product, and service names may be trademarks or service marks of others.

Glossary

Glossary

Use this section to locate definitions of some of the IBM Directory product terms

access control list (ACL)

In computer security, a list associated with an object that identifies all the subjects that can access the object and their access rights.

access control groups

Groups to be used for access control. Each group contains a multivalued attribute consisting of member DNs. Access control groups have an object class of 'AccessGroup'.

access permissions

There are two sets of access permissions:

- Permissions that apply to an entire object
- Permissions that apply to attribute access classes or individual attributes.

aclEntry

A multivalued attribute that contains information pertaining to the access allowed to the entry and its attributes. An aclEntry lists the following types of information: who has rights to the entry (scope of the protection), what attributes or classes of attributes the user has access to (attribute access classes), and what rights the user or group has (permission).

aclPropagate

The attribute that controls ACL propagation. If the value is set to true, ACLs are propagated down the hierarchy tree. If the value is set to false, the ACL becomes an override, pertaining only to this particular object.

aclSource

A read only operational attribute that is associated with each object. This attribute contains the distinguished name (DN) of the entry in which the access control list (ACL) is defined.

alias A pointer to another directory object.

Aliases can be used within LDAP to reference entries anywhere within the directory tree.

attribute access class

Class that consists of attributes that require similar permission for access. Attributes are assigned to an access class within the schema files. The user-modifiable access classes are normal, sensitive, critical, and restricted. An additional class of system is not user-modifiable.

cascading replication

A replication topology in which there are multiple tiers of servers. A peer/master server replicates to a small set of read-only servers which in turn replicate to other servers. Such a topology off-loads replication work from the master servers.

consumer server

A server which receives changes through replication from a supplier server.

directory schema

The valid attribute types, object classes, matching rules and syntaxes that can appear in a directory. The attribute types and object classes define the syntax of the attribute values, which attributes must be present, and which attributes may be present for specific object classes.

directory server instance

A directory server instance is comprised of all of the nonexecutable files that are required for a directory server and its corresponding administration daemon to run on a machine. These files include the ibmslapd.conf file, the schema files, the stash files, and the log files of the directory server instance. Each server instance and its corresponding administration daemon listens on a unique port with the same IP address.

distinguished name (DN)

The name that uniquely identifies an entry in a directory. A distinguished name is made up of attribute=value pairs, separated by commas.

dynamic group

A group that is defined using a search expression. A directory entry that matches the search expression is automatically a member of the group.

entryOwner

An attribute whose value can refer to a user or a group. Each entry has an associated entryOwner attribute. However, the entryOwner subject has all authority to the entry.

Forwarding server

A read-only server that replicates all changes sent to it. This contrasts to a peer/master server in that it is read only and it can have no peers.

Gateway server

A server that forwards all replication traffic from the local replication site where it resides to other Gateway servers in the replicating network. Also receives replication traffic from other Gateway servers within the replication network, which it forwards to all servers on its local replication site.

Gateway servers must be masters (writable).

group A logical organization of users based on some common criteria. Groups can be used in specifying a common set of directory access permissions.

gsk7ikm

A utility that creates public-private key pairs and certificate requests, receives certificate requests into a key database, and manages keys in a key database.

idsbulkload

A command line utility that is used for bulk-loading large amounts of data in LDIF format.

idsldapadd

The LDAP modify-entry and LDAP add-entry tool idsldapmodify is a shell-accessible 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.

idsldapdelete

The LDAP delete-entry tool ldapdelete is

a shell-accessible interface to the ldap_delete library call. ldapdelete opens a connection to an LDAP server and binds and deletes one or more entries. If one or more dn arguments are provided, entries with those Distinguished Names (DN) are deleted. Each DN should be a string-represented DN.

idsldapmodify

The LDAP modify-entry and LDAP add-entry tools idsldapmodify is a shell-accessible 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.

idsldapmodrdn

LDAP modify-entry RDN tool idsldapmodrdn is a shell-accessible interface to the ldap_modrdn library call. **idsldapmodrdn** opens a connection to an LDAP server and binds and modifies the RDN of entries. The entry information is read from standard input, from a file, through the use of the **-f** option, or from the command-line pair DN and RDN.

idsldapsearch

The LDAP search tool idsldapsearch is a shell-accessible interface to the ldap_search library call. **idsldapsearch** opens a connection to an LDAP server and binds and performs a search using the filter. The filter should conform to the string representation for LDAP filters.

idsldif2db

This program is used to load entries specified in text LDAP Directory Interchange Format (LDIF) into a directory stored in a relational database. 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.

indexing rules

Index rules attached to attributes make it possible to retrieve information faster. The IBM Tivoli Directory Server provides the following indexing rules:

- Equality
- Approximate
- Substring

- Reverse

See “Indexing rules” on page 42.

LDAP Data Interchange Format (LDIF)

A format used by the LDAP import-export tools as well as `idsldapmodify`, `idsldapadd`, and `idsldapsearch` command-line utilities to represent LDAP entries or changes to entries in a standard portable text form. See RFC 2849.

matching rule

A rule that describes how to perform a comparison.

multiple values

Multiple values are used to assign more than one value to an attribute. The attribute can have multiple values, for example, to accommodate a maiden and married last name. To add multiple values to an attribute, click **Multiple values**, then add one value per line. If an attribute contains multiple values, the field displays as a drop-down list.

nested group

A child group entry whose distinguished name (DN) is referenced by an attribute contained within a parent group entry. The `ibm-membergroup` attribute has been defined to explicitly distinguish nested groups from ordinary members.

nested subtree

A subtree within another subtree of the directory.

object class definition

Statement that specifies which attributes must be present in an object of that class, as well as attributes that might be present. Every entry contains an `objectClass` attribute that identifies what type of information the entry contains.

object class types

Object classes can be structural, for example, **person**; abstract, for example **top**; or auxiliary, for example **ePerson**.

ownerPropagate

The attribute that controls directory object ownership propagation. If the value is set to true, directory object ownership is propagated down the hierarchy tree. If that attribute is set to false, the entry

owner specified is an override, pertaining only to this particular entry.

ownerSource

A read only operational attribute that contains the distinguished name (DN) of the entry in which the owner values are defined. Each entry has an associated `ownerSource` attribute. This attribute is maintained by the server but can be retrieved for administrative purposes.

peer server

The term used for a master server when there are multiple masters for a given subtree. A peer server does not replicate changes sent to it from another peer server; it only replicates changes that are originally made on it.

proxy server

A server that receives requests intended for another server and that acts on the client’s behalf (as the client’s proxy) to obtain the requested service. A proxy server is often used when the client and the server are incompatible for direct connection. For example, the client is unable to meet the security authentication requirements of the server but should be permitted some services.

quiesce

To put the server into a state in which it does not accept client updates, except for those done by the administrator and accompanied by replication management control.

referral

A way for servers to refer clients to additional directory servers. Referrals can distribute namespace information among multiple servers, provide knowledge of where data resides within a set of interrelated servers, and route client requests to the appropriate server. The general format for a referral is: `ldap[s]://hostname:port`. Typically the format for a referral to a nonsecure server is: `ldap://hostname:389` and to a secure SSL server is: `ldaps://hostname:636`.

relative distinguished name (RDN)

The first component of the distinguished name (DN). For example, if the entry’s DN is `cn=John Doe,ou=Test,o=IBM,c=US`, the RDN is `cn=John Doe`.

replica

A server that contains a copy of the directory or a copy of part of the directory of another server. Replicas back up servers in order to enhance performance or response times and to ensure data integrity.

replicated subtree

A portion of the directory information tree that is replicated from one server to another. Under this design, a given subtree can be replicated to some servers and not to others. Subtrees can be writable on a given server, or read-only.

Replicating network

A network that contains connected replication sites.

replication agreement

Information contained in the directory that defines the connection or replication path between two servers. One server is called the supplier (the one that sends the changes) and the other is the consumer (the one that receives the changes). The agreement contains all the information needed for making a connection from the supplier to the consumer and scheduling replication.

replication context

The replication context identifies the root of a replicated subtree. The configuration information related to replication is maintained in a set of entries created below a replication context.

replication site

A Gateway server and any master, peer or replica servers configured to replicate together.

role

A job function that identifies the tasks that a user can perform and the resources to which a user has access. A user can be assigned one or more roles.

or

Defines what access levels a given user has and the specific resources they can modify at those levels. The user may be limited in how they can access information if they do not have the proper role. Multiple roles are permissible.

Secure Sockets Layer (SSL)

A security protocol that provides communication privacy. SSL enables client and server applications to communicate in a way that is designed to prevent eavesdropping, tampering, and message forgery. SSL was developed by Netscape Communications Corp. and RSA Data Security, Inc.

sorted search

Search that allows a client to receive search results sorted based on a list of criteria, where each criteria represents a sort key. This moves the responsibility of sorting from the client application to the server, where it might be done more efficiently.

subtree

A section of a directory hierarchy, which is also called a directory tree. The subtree typically starts at a particular directory and includes all subdirectories and objects below that directory in the directory hierarchy; that is, any subdirectories or objects connected to the directory or to any lower level of its subdirectories.

suffix

A distinguished name (DN) that identifies the top entry in a locally held directory hierarchy. Because of the relative naming scheme used in Lightweight Directory Access Protocol (LDAP), this suffix applies to every other entry within that directory hierarchy. A directory server can have multiple suffixes, each identifying a locally held directory hierarchy. A suffix is also known as a naming context.

supplier server

A server that sends changes to a consumer server.

syntax Syntax refers to the required format for the values of an attribute. Supported syntaxes are:

IBM Attribute Type Description
 Matching Rule Description
 Name Form Description
 Attribute Type Description
 Object Class Description
 DIT Structure Rule Description
 DIT Content Rule Description
 LDAP Syntax Description
 OID
 Matching Rule Use Description
 Boolean - TRUE/FALSE
 Binary - octet string
 INTEGER - integral number

Generalized Time
IA5 String - case-sensitive string
Directory String - case-insensitive
string
UTC time
Telephone Number
DN - distinguished name

Index

Numerics

24/7 259

A

- access control lists 309
- access controls
 - dynamic schema 52
- access evaluation
 - combinatory rule 317
 - specificity rule 317
- access permissions
 - LDAP operations 314
- access rights 313
- ACI mechanisms
 - OIDs 477
- ACL
 - propagation of 315
- ACL cache size 97
- ACLs 309
 - filter-based 310
 - filtered 321
 - non-filtered 320
 - syntax 310
- adding an auxiliary object class 303
- adding servers 181, 228
- admin daemon audit
 - audit format 585
- administration
 - name 67
 - password 67
- administration daemon 17
 - audit logs 271
 - disabling 273
 - error logs 270
 - SSL security 590
 - Starting an instance of the directory
 - administration daemon 17
 - Stopping an instance of the directory
 - administration daemon 17
- administration daemon audit logs 271
 - disabling 273
- administration daemon error logs 270
- administration DN
 - changing 440
- administration password
 - configuring 440
- administrative group 86
 - adding members 88
 - modifying members 89
 - removing members 90
- administrator
 - administrator group 86
 - realms 353
- agreements
 - replication 169
- application servers
 - apache tomcat 21
 - embedded version of IBM WebSphere
 - Application Server - Express 21

- ASCII characters
 - 33 to 126 487
 - allowable in encryption seed string 487
- associating
 - servers with referrals 154
- attribute
 - cache 110
 - MAY 62
 - MUST 62
 - syntax 50
- attribute cache 110
 - adding attributes 110
 - removing attributes 110
- attribute types
 - group 334
 - schema file 31
- attributes 39
 - adding 44
 - binary 295
 - configuration schema 547, 551
 - copying 47
 - deleting 49
 - dynamically- changed 578
 - editing 46
 - multiple values 295
 - unique 91
 - viewing 43
- audit
 - error logs 274
 - disabling 278
- Audit error logs 274
 - disabling 278
- authentication
 - client 120
 - server 114
 - server and client 114
- auxiliary object class
 - adding 303
 - deleting 304

B

- binary attributes 295
- browsing the directory tree 293
- bulkload 423
 - error logs 279

C

- certificate authority 121
 - distinguished names 127
- certificate requests 125
- certificates 121
- change log
 - configuring 429
 - unconfiguring 455
- changing ports 95
- checking
 - entries 61

- client
 - SSL security 596
- client authentication 120
- client utilities
 - idsldapadd 368, 396
 - idsldapchangepwd 370
 - idsldapdelete 373
 - idsldapdiff 377
 - idsldapexop 386
 - idsldapmodify 368, 396
 - idsldapmodrdrn 402
 - idsldapsearch 406
 - ldapadd 368, 396
 - ldapchangepwd 370
 - ldapdelete 373
 - ldapdiff 377
 - ldapexop 386
 - ldapmodify 368, 396
 - ldapmodrdrn 402
 - ldapsearch 406
- commands 367
 - bulkload 423
 - db2ldif 436
 - dbback 434
 - dbrestore 435
 - ddsetup 420
 - ibmdiradm 438
 - ibmdirctl 368
 - ibmslapd 454
 - idsbulkload 423
 - idscfgchglg 429
 - idscfgdb 431
 - idscfgsch 432
 - idscfgsuf 433
 - idsdb2ldif 436
 - idsdbback 434
 - idsdbrestore 435
 - idsdiradm 438
 - idsdirctl, ibmdirctl 369
 - idsdnpw 440
 - idsgendirksf 441
 - idsicrt 442
 - idsidrop 445
 - idsilist 446
 - idsimigr 448
 - idsldapadd 368, 396
 - idsldapchangepwd 368
 - idsldapdelete 373
 - idsldapdiff 377
 - idsldapexop 368, 386
 - idsldapmodify 368, 396
 - idsldapmodrdrn 402
 - idsldapsearch 406
 - idsldaptrace 368, 415
 - idsldif2db 448
 - idslink 450
 - idslogmgmt 449
 - IDSProgRunner 450
 - idsrunstats 451
 - idssethost 451
 - idssetport 452

- commands (*continued*)
 - idsslapd 454
 - idssnmp 455
 - idssupport 455
 - idsucfgchlg 455
 - idsucfgdb 456
 - idsucfgsch 457
 - idsucfgsuf 459
 - ldapadd 368, 396
 - ldapchangepwd 368
 - ldapdelete 373
 - ldapdiff 377
 - ldapexop 368, 386
 - ldapmodify 368, 396
 - ldapmodrdn 402
 - ldapsearch 406
 - ldaptrace 368, 415
 - ldif2db 448
 - ldtrc 460
 - runscript 462
 - runstats 451
 - common schema 33
 - complex topology with peer-to-peer
 - creating 192
 - configuration only mode 19
 - how to start 19
 - requirements 19
 - using Web Administration to start 19
 - verifying that the server is running in configuration only mode 20
 - configuration schema
 - attributes 547
 - object classes 547
 - configuration tools
 - error logs 280
 - configuration tools log 280
 - connections 82
 - preventing denial of service 84
 - properties 84
 - console 22
 - adding servers to 27
 - changing login 27
 - changing password 27
 - changing properties 28
 - logging off of 23
 - logging on to 22
 - managing 27
 - modifying servers 28
 - removing servers from 28
 - controls
 - OIDs 479
 - copying an entry 302
 - customer support
 - see Software Support 600
- ## D
- data interchange format 481
 - database
 - backing up 434
 - configuring 431
 - partitioning 420
 - recovering 217
 - catastrophic failure 220
 - single-server failure 219
 - restoring 435
 - unconfiguring 456
 - database connections
 - number of 97
 - DB2
 - error logs 281
 - DB2 error logs 281
 - db2ldif 436
 - dbback 434
 - dbrestore 435
 - ddsetup 420
 - debug
 - tracing 460
 - debugging
 - levels of 462
 - default log paths 267
 - default log settings
 - modifying 269
 - defining a directory 3
 - Directory clients and servers 3
 - Directory security 4
 - deleting
 - keys 124
 - deleting an auxiliary object class 304
 - deleting an entry 300
 - deleting entries 373
 - DEN 62
 - DIGEST-MD5
 - configuring 150
 - directories
 - distributed 247
 - directory key stash file
 - regenerating 441
 - directory management 293
 - attributes 295
 - browsing the directory tree 293
 - copying an entry 302
 - deleting an entry 300
 - directory entries 293
 - editing entry ACLs 303
 - entries 294
 - modifying an entry 301
 - directory server
 - error logs 284
 - directory server error logs 284
 - directory server instance
 - configuring a changelog 429
 - configuring a database 431
 - configuring a schema file 432
 - creating 442
 - listing 446
 - regenerating a directory key stash file 441
 - removing 445
 - unconfiguring a changelog 455
 - unconfiguring a database 456
 - unconfiguring a schema file 457
 - directory-enabled network
 - schema support 62
 - disallowed changes
 - schema 52
 - attributes 53
 - matching rules 61
 - object classes 52
 - syntaxes 61
 - distinguished name 11
 - pseudo 312
 - distributed directories 247
 - back-end servers 252
 - distributed directories (*continued*)
 - backup replication 260
 - server groups 261
 - creating 252, 254
 - distributed directory setup tool 249
 - fail over & load balancing 259
 - global policies topology
 - creating 264
 - LDIF file
 - creating 262
 - partition entries 252
 - partitioned data
 - loading 265
 - partitioning the data 265
 - proxy 247, 252, 254, 260, 261
 - proxy servers
 - creating 264
 - RDN hash 247
 - replication topology
 - creating 263
 - splitting data 247
 - starting replication 266
 - synchronizing information 251
 - distributed directory
 - partitioning databases 420
 - distributed directory setup tool
 - options 587
 - DN 11
 - pseudo 312
 - DN escape characters 12
 - dynamic
 - changes
 - schema 51
 - dynamic group
 - editing a memberURL 341
 - dynamic group entry
 - creating 336
 - dynamic groups 329
 - dynamic schema
 - access controls 52
 - changes 51
 - matching rules 41
 - replication 52
 - dynamically-changed attributes 578
- ## E
- editing entry ACLs 303
 - encryption
 - levels of 130
 - one-way encrypting
 - crypt 132
 - SHA-1 132
 - SSL 130
 - two-way encrypting
 - AES128 132
 - AES192 132
 - AES256 132
 - Enhanced DN processing 13
 - entries 335, 336
 - adding 294
 - adding an auxiliary objectclass 303
 - deleting an auxiliary object class 304
 - entry
 - changing passwords 370
 - deleting 373
 - idsldapchangepwd 370

- entry (*continued*)
 - modifying 402
 - searching 406
- entry checking
 - against schema 61
- error
 - tracing 460
- error codes 467
- error handling
 - replication 168
- error numbers 467
- errors
 - ldap 467
- escaping rules 12
- event notification
 - disabling 104
 - enabling 104
- example
 - LDIF 481
 - Version 1 482
- exporting
 - keys 126
- extended operations 386
 - OIDs 478

F

- filtered ACLs 310, 321
 - sample LDIF file 539
- fixes, obtaining 599

G

- gateway topology
 - creating 207
- generalized time 64
- global administration group 6
- global security 121
- group
 - attribute types 334
 - membership 340
 - object classes 334
- group task
 - verifying 338
- groups 329
 - creating 357
 - dynamic 329
 - hybrid 331
 - management of 363
 - members of 339
 - membership 331
 - nested 331
 - proxy authorization 347
 - copying 349
 - creating 347
 - modifying 349
 - removing 350
 - search limit 343
 - static 329
- GSKit 121

H

- hybrid groups 331

I

- IANA character sets 483
- IBM Directory schema
 - managing 31
- IBM Tivoli Directory Integrator 491
- IBM Tivoli Directory server
 - SSL security 590
- IBMAttributeTypes 40
- ibmdiradm 438
- ibmdirctl 368
- ibmslapd 454
- ibmslapd options 19
- ibmslapd.conf 108
- IBMSubschema 50
- identity mapping
 - Kerberos 146
- idsbulkload 279, 423
 - error logs 279
- idscfgchglg 429
- idscfgdb 431
- idscfgsch 432
- idscfgsuf 433
- idsdb2ldif 436
- idsdbback 434
- idsdbrestore 435
- idsdiradm 438
- idsdirctl, ibmdirctl 369
- idsdnpw 440
- idsexop 95
- idsgendirksf 441
- idsicrt 442
- idsidrop 445
- idsilist 446
- idsimigr 448
- idsldapadd 396
- idsldapchangePwd 368, 370
- idsldapdelete 373
- idsldapdiff 377
- idsldapexop 368, 386
- idsldapmodify 95, 368, 396
- idsldapmodrtn 402
- idsldapsearch 406
- idsldaptrace 368, 415
- idsldif2db 448
- idslink 450
- idslogmgmt 449
- IDSProgRunner 450
- idsrunstats 451
- idssethost 451
- idssetport 452
- idsslapd 454
- idsslapd options 19
- idsslapd.conf 108
- idssnmp 455
- idssupport 455
- idsucfgchglg 455
- idsucfgdb 456
- idsucfgsch 457
- idsucfgsuf 459
- importing
 - keys 127
- information centers, searching to find
 - software problem resolution 599
- inheritance
 - object class 34

instance

- directory server
 - creating 442
 - listing 446
 - removing 445
- Internet, searching to find software
 - problem resolution 599
- iPlanet
 - compatibility 63
 - grammar 63
- ipv4 489
- Ipv6 489

K

- Kerberos 144
- key
 - certificate request for existing
 - key 128
 - changing the database password 124
 - defaults 125
 - deleting 124
 - exporting 126
 - importing 127
 - self-signing 125
 - showing information about 124
 - trusted root 127
 - trusted root removal 128
- key database
 - setting 129
- key pairs 121
- key stash file
 - regenerating 441
- keyring file
 - migration 129
- keys
 - private 121
 - public 121
- knowledge bases, searching to find
 - software problem resolution 599

L

- language support 483
- language tags 297
 - attributes cannot have associated
 - language tags 298
 - attributes containing language tags
 - searching 299
 - attributes language tag values 298
 - disabling 95
 - enabling 95
 - language tag descriptor
 - removing 299
- ldapadd 368, 396
- ldapchangePwd 368, 370
- ldapdelete 373
- ldapdiff 377
- ldapexop 368, 386
- ldapmodify 368, 396
- ldapmodrtn 402
- ldapsearch 406
- ldaptrace 368, 415
- LDIF 481
- ldif2db 448
- ldtrc 460

- levels of debugging 462
- load balancing 259
- log paths
 - default 267
- log settings
 - default
 - modifying 269
- Logging in to the console 22
 - console administrator 22
- logs
 - administration daemon audit 267
 - administration daemon error 267
 - audit 267
 - administration daemon 271, 273
 - bulkload 267
 - configuration tools 267
 - DB2 267
 - default settings 267
 - disabling 273, 278
 - errors
 - administration daemon 270
 - audit 274, 278
 - bulkload 279
 - configuration tools 280
 - DB2 281
 - directory server 284
 - idsbulkload 279
 - lost and found 283
 - viewing 285
 - idslogmgmt 268, 449
 - log management tool 268
 - lost and found 267
 - server error 267
- lost and found
 - error logs 283
- lost and found log 283

M

- managing
 - replication 242
- master-replica topology
 - creating 172
- master/replica
 - unconfiguring 205
- matching rules 41
- members
 - managing 339
- memberships 340
- messages
 - error 467
- migration 448
 - keyring file 129
- modifying an entry 301
- modifying entries 402
- monitor
 - service status 76
- multi-threaded
 - replication 221

N

- namespace 155
- nested group entry
 - creating 337
- nested groups 331

- non-filtered ACLs 320
 - sample LDIF file 539
- notification
 - event 104

O

- object class
 - auxiliary 303
 - IBMAttributeTypes 40
 - IBMsubschema 50
- object classes 33
 - adding 35
 - configuration schema 547
 - copying 38
 - deleting 39
 - editing 36
 - group 334
 - viewing 34
- object identifier 33
- OID 33
- OIDs
 - ACI mechanisms 477
 - controls 479
 - extended operations 478
 - root DSE 473
 - supported and enabled
 - capabilities 475
- operational attributes
 - password policy 495
- operations
 - extended 386

P

- partitioning
 - databases 420
- password
 - administrator 67
 - console administrator 27
 - security 134
 - syntax requirements 143
- password policy 137
 - add/update for an entry 498
 - operational attributes 495, 497
 - overriding 496
 - queries 495
 - replicating 497
 - unlocking accounts 496
- passwords
 - administration 67
 - changing 370
- peer-to-peer
 - replication 192
- performance 97
- problem determination
 - describing problem for IBM Software Support 601
 - determining business impact for IBM Software Support 601
 - submitting problem to IBM Software Support 601
- propagation
 - ACL 315
- proxy authorization
 - groups 347

- proxy server
 - backing up 259
 - failover 259
- pseudo DNs 312

Q

- queries
 - schema 51
- queues
 - replication 240

R

- rdn 402
- realms 353
 - adding 357
 - adding user 357
 - administrator 353
 - creating 353
 - management of 358
 - template 357
- recovery
 - database 217
- ref attribute 153
- referral
 - object class 153
 - ref attribute 153
- referrals 153
 - default
 - creating 157
 - distributing the namespace 155
 - entries 153
 - modifying 159
 - removing 160
 - server association 154
- replica
 - creating servers 161
- replicating
 - operational attributes 497
 - password policy 497
- replicating servers 181, 228
- replication
 - adding a subtree 222
 - adding credentials 224
 - command line tasks 242
 - configuration information 242
 - creating gateway servers 245
 - monitoring status 243
 - supplier DN and password for a subtree 242
 - complex topology with
 - peer-to-peer 192
 - creating a master-replica topology 172
 - credentials 224
 - demoting a master 233
 - dynamic schema 52
 - editing a subtree 223
 - editing an agreement 234
 - error handling 168
 - error table 221
 - managing credential ACLs 227
 - managing gateway servers 234
 - managing topologies 228
 - master server 173

- replication (*continued*)
 - master-forwarder-replica 186
 - modifying credentials 227
 - modifying properties 237
 - moving or promoting a server 233
 - multi-threaded 221
 - of subtrees 173
 - overview 164
 - cascading replication 165
 - gateway replication 166
 - peer-to-peer replication 165
 - replication conflict resolution 167
 - simple replication 164
 - queues 240
 - quiescing a subtree 223
 - recovery procedures 217
 - removing a server 233
 - removing a subtree 223
 - removing credentials 227
 - replicas 174, 231
 - replication schedule 235
 - schedules 238
 - schema and password policy
 - updates 171
 - server errors 235
 - server information 235
 - server roles 163
 - setting up a gateway topology 207
 - simple topology with peer
 - replication 181
 - subtrees 222
 - supplier information 177, 236
 - terminology 161
 - unconfiguring a master/replica 205
 - viewing topologies 228
- replication conflict resolution 167
- replication method
 - multi-threaded 221
- replication topology
 - example procedure 204
- required attribute definitions 501
- required permissions 314
- restoring a database 435
- roles 342
- root DSE
 - attributes 473
- rules
 - indexing 42
 - attributes 42
- runscript 462
- runstats 451

S

- sample LDIF file
 - filtered ACLs 539
 - non-filtered ACLs 539
- schedules
 - daily 239
 - weekly 238
- schema
 - attribute types 31
 - attributes 39
 - adding 44
 - copying 47
 - deleting 49
 - editing 46

- schema (*continued*)
 - attributes (*continued*)
 - viewing 43
 - changes
 - disallowed 52
 - checking 61
 - common 33
 - support 32
 - dynamic
 - changes 51
 - file
 - attribute types 31
 - IBM Tivoli Directory Server Version 6.0 545
 - object classes 33
 - adding 35
 - attributes 34
 - copying 38
 - defining 33
 - deleting 39
 - editing 36
 - viewing 34
 - queries 51
 - subschema entries 50
- schema file
 - configuring 432
 - unconfiguring 457
- search
 - paged results 103
 - paging 99
 - settings 99
 - size limits 99
 - sorted 99
 - time limits 99
- search filter elements
 - number of 97
- search limit group
 - copying 346
 - creating 343
 - modifying 345
 - removing 346
- search limits
 - groups 343
- searches
 - advanced 306
 - entries 305
 - extended controls 101
 - manual 307
 - paging and sorting 101
 - simple 305
 - size limit 343
 - time limit 343
- searching entries 406
- secure sockets layer 113
- security 121
 - Kerberos 144
 - password policy 134
 - self-signed server certificate 118
 - setting the key database 129
 - SSL 113, 115
 - TLS 115
- self-signing keys 125
- server
 - SSL security 596
 - starting 68
 - stopping 68
- server and client authentication 114

- server audit
 - audit format 583
- server authentication 114
- server certificates 117
- server performance
 - settings 97
- server properties
 - setting 95
- server replication
 - gateway 181, 228
 - masters 181, 228
 - peer 181, 228
- server startup
 - configuration only mode 19
- server status 69
 - changelog cached attributes 74
 - changelog cached candidates 75
 - directory cached attributes 73
 - directory cached candidates 74
 - general server status 69
 - operation counts 71
 - work queue 72
 - worker status 72
- server utilities
 - bulkload 423
 - db2ldif 436
 - dbback 434
 - dbrestore 435
 - ddsetup 420
 - ibmdiradm 438
 - ibmslapd 454
 - idsbulkload 423
 - idscfgchglg 429
 - idscfgdb 431
 - idscfgsch 432
 - idscfgsuf 433
 - idsdb2ldif 436
 - idsdbback 434
 - idsdbrestore 435
 - idsdiradm 438
 - idsdirctl, ibmdirctl 369
 - idsdnpw 440
 - idsgendirksfh 441
 - idsicrt 442
 - idsidrop 445
 - idsilist 446
 - idsimigr 448
 - idsldaptrace 415
 - idsldif2db 448
 - idslink 450
 - idslogmgmt 449
 - IDSProgRunner 450
 - idsrunstats 451
 - idssethost 451
 - idssetport 452
 - idsslapd 454
 - idssnmp 455
 - idsupport 455
 - idsucfgchglg 455, 457
 - idsucfgdb 456
 - idsucfgsuf 459
 - ldaptrace 415
 - ldif2db 448
 - ldtrc 460
 - runscript 462
 - runstats 451
- setting searches 99

- Simple Networking Management Protocol 491
- simple topology with peer replication
 - creating 181
- SNMP 491
- Software Support
 - contacting 600
 - describing problem for IBM Software Support 601
 - determining business impact for IBM Software Support 601
 - submitting problem to IBM Software Support 601
- sorted search 102, 409
 - idsldapsearch 409
- SSL 113, 419
- SSL scenarios
 - administration daemon 590
 - client and server 596
 - IBM Tivoli Directory server 590
 - SSL security 589, 590, 596
 - WebSphere Application Server Version 5.1.1 589, 590
- Starting an instance of the directory
 - administration daemon 17
- starting the server 68
 - configuration only mode 19
- static group entry
 - creating 335
- static groups 329
- status
 - connections 82
 - server 69
- Stopping an instance of the directory
 - administration daemon 17
- stopping the server 68
- subclassing 34
- subschema entries 50
- subtree
 - comparing 377
- subtree comparison 377
- subtree replication considerations 328
- suffix
 - removing 459
- suffixes 108
 - adding 108
 - configuring 433
 - removing 109
- supplier information 177, 236
- supported and enabled capabilities
 - OIDs 475
- synchronizing
 - instances 537
 - two-way cryptography 537
- syntax
 - ACL 310
 - attribute 50
 - Backus Naur Form 11
 - distinguished name 11
 - special characters 12

T

- Tables
 - Filtering 26
 - Finding 25
 - Paging 25
- Tables (*continued*)
 - Select Action drop-down menu 24
 - Sorting 25
 - table icons 24
 - Web Administration Tool 23
- tbindmsg 418
- template
 - adding 357
 - realms 357
- templates
 - creating 355
 - management of 359
- The IBM Tivoli Directory Server 5
- time
 - generalized 64
 - UTC 64
- TLS 113, 419
- topology
 - replication 161, 163
- tracing
 - errors 460
- transaction layer security 113
- transaction support
 - disabling 106
 - enabling 106
- transactions
 - settings 106
- trusted root 127
- two-way cryptography
 - instances 537
 - synchronizing 537

U

- unique attributes 91
 - creating 91
 - removing an attribute 92
- unlocking accounts
 - password policy 496
- URL formats
 - ipv4 489
 - ipv6 489
- users
 - management of 362
- UTC time 64
- UTF-8 483
- utilities
 - client 368
 - command line 367
 - bulkload 423
 - db2ldif 436
 - dbback 434
 - dbrestore 435
 - ddsetup 420
 - ibmdiradm 438
 - ibmslapd 454
 - idsbulkload 423
 - idscfgchglg 429
 - idscfgdb 431
 - idscfgsch 432
 - idscfgsuf 433
 - idsdb2ldif 436
 - idsdbback 434
 - idsdbrestore 435
 - idsdiradm 438
 - idsdirectl, ibmdirctl 369
 - idsdnpw 440
- utilities (*continued*)
 - command line (*continued*)
 - idsgendirksf 441
 - idsicrt 442
 - idsidrop 445
 - idsiist 446
 - idsimigr 448
 - idsldapadd 368, 396
 - idsldapchangepwd 370
 - idsldapdelete 373
 - idsldapdiff 377
 - idsldapexop 386
 - idsldapmodify 368, 396
 - idsldapmodrdrn 402
 - idsldapsearch 406
 - idsldaptrace 415
 - idsldif2db 448
 - idslink 450
 - idslogmgmt 449
 - IDSProgRunner 450
 - idsrunstats 451
 - idssethost 451
 - idssetport 452
 - idsslapd 454
 - idssnmp 455
 - idssupport 455
 - idsucfgchglg 455
 - idsucfgdb 456
 - idsucfgsch 457
 - idsucfgsuf 459
 - ldapadd 368, 396
 - ldapchangepwd 370
 - ldapdelete 373
 - ldapdiff 377
 - ldapexop 386
 - ldapmodify 368, 396
 - ldapmodrdrn 402
 - ldapsearch 406
 - ldaptrace 415
 - ldif2db 448
 - ldtrc 460
 - runscript 462
 - runstats 451
 - ibmdiradm 438
 - idsdiradm 438
 - idsimigr 448
 - idsldapadd 368, 396
 - idsldapchangepwd 370
 - idsldapdelete 373
 - idsldapdiff 377
 - idsldapexop 386
 - idsldapmodify 368, 396
 - idsldapmodrdrn 402
 - idsldapsearch 406
 - idslink 450
 - idslogmgmt 449
 - IDSProgRunner 450
 - idssnmp 455
 - idssupport 455
 - idsucfgsuf 459
 - ldapadd 368, 396
 - ldapchangepwd 370
 - ldapdelete 373
 - ldapdiff 377
 - ldapexop 386
 - ldapmodify 368, 396
 - ldapmodrdrn 402

utilities (*continued*)
 ldapsearch 406
 runscript 462
 server 420

V

viewing
 error logs 285
viewing log 285

W

Web address protocols
 ipv4 489
 ipv6 489
Web admin daemon 17
Web administration console 22
Web Administration console
 logs 267
Web Administration Tool
 console 22
 managing the console 27
 setting up 27
 starting 21
WebSphere Application Server Version
 5.1.1
 SSL security 590
worker
 server status 80



Printed in USA

SC32-1674-00

