IBM Tivoli Directory Server

Administration Guide

Version 6.3

SC27-2749-00

IBM Tivoli Directory Server

Administration Guide

Version 6.3

SC27-2749-00

Note

Before using this information and the product it supports, read the general information under Appendix R, "Notices," on page 629.

This edition applies to version 6, release 3, of the IBM Tivoli Directory Server and to all subsequent releases and modifications until otherwise indicated in new editions.

© Copyright IBM Corporation 2002, 2010.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

About this book ix
Intended audience for this book. ix
Publications ix
IBM Tivoli Directory Server version 6.3 library . . ix
Related publications x
Accessing terminology online. x
Accessing publications online. x
Ordering publications xi
Accessibility xi
Tivoli technical training xi
Tivoli user groups xi
Support information xii
Conventions used in this book xii
Typeface conventions xii
Operating system-dependent variables and paths xiii

Part 1. Directory overview 1

Chapter 1. Defining a directory 3
Directory clients and servers 3
Directory security. 4

Chapter 2. IBM Tivoli Directory Server . . 5

Chapter 3. Distinguished names (DNs) 13
Distinguished name syntax 13
DN escaping rules 14
Enhanced DN processing 15

Part 2. Server Administration 17

Chapter 4. Directory administration server 19
Starting an instance of the directory administration server 20
Stopping an instance of the directory administration server 20
Starting the directory server instance at operating system startup 20
Autostart on Windows systems. 20
Autostart on AIX, Linux, and Solaris systems . . 21

Chapter 5. Configuration only mode . . 23
Minimum requirements for configuration only mode 23
How to start in configuration only mode 23
Using Web Administration 23
Using the command line 24
How to verify that the server is running in configuration only mode 24
Using Web Administration 24
Using the command line 24

Chapter 6. Web Administration Tool graphical user interface (GUI) 25
Starting the Web application server to use the Web administration tool 25
Starting the Web Administration Tool. 26
Logging on to the console as the server administrator, a member of an administrative group or an LDAP user 27
Console layout 27
Logging off the console 28
Using tables in the Web Administration Tool . . . 28
Table icons 28
Select Action drop-down menu 29
Paging 30
Restore Defaults 30
Sorting 30
Finding. 30
Filtering 31
Reordering 31

Chapter 7. Setting up the Web Administration Tool 33
Managing the console 33
Changing the console administrator login . . . 33
Changing the console administration password 33
Adding, modifying, and removing servers in the console 33
Managing console properties 34
Manage properties for webadmin searches . . . 35
Viewing scenario-based help files in the Web Administration Tool 35

Chapter 8. Managing the IBM Directory schema 37
Common schema support 39
Object identifier (OID). 39
Working with object classes 39
Defining object classes. 40
Viewing object classes 41
Adding an object class. 42
Editing an object class 43
Copying an object class 44
Deleting an object class 46
Working with attributes 46
The IBMAttributeTypes attribute type 47
Equality matching rules 48
Indexing rules 50
Viewing attributes 51
Adding an attribute 52
Editing an attribute. 53
Copying an attribute 55
Deleting an attribute 57
Encrypted Attributes 57
Using command line 59

© Copyright IBM Corp. 2002, 2010

Attribute syntax 60
Managing unique attributes 61
The subschema entries. 64
The IBMsubschema object class 64
Schema queries 64
Dynamic schema 64
Access controls 65
Replication 65
Disallowed schema changes 65
Object classes. 66
Attributes 66
Syntaxes 76
Matching rules 76
Schema checking 76
Checking an entry against the schema 77
iPlanet compatibility 78
Generalized and UTC time 79

Chapter 9. Basic server administration tasks 81
Changing the primary administrator distinguished name and password 81
Using Web Administration 81
Using the command line 81
Starting and stopping the server 82
Using Web Administration 82
Using the command line or Windows Services icon:. 83
Checking server status. 83
Using Web Administration 84
Using the command line 89
View cache status 98
Entry cache 98
Filter cache 98
ACL cache. 99
Group members' cache 99
Directory cached attributes 100
Directory cache candidates 100
Viewing server capabilities (Root DSE) information 101
Using Web Administration 101
Using command line 104
Managing server connections 104
Using Web Administration 104
Using the command line 105
Managing connection properties 106
Using Web Administration 106
Using the command line 107

Chapter 10. Setting server properties 109
Changing server ports and enabling language tags 110
Using Web Administration 110
Using the command line. 111
Setting Performance 111
Using Web Administration 111
Using the command line. 112
Enforcing minimum ulimits 112
Using Web Administration 114
Using the command line. 114
Search Settings 115
Using Web Administration 115
Using the command line. 117
Searching the directory with paging and sorting 118
Sorted search control 118
Simple paged results 119
Virtual list view 121
Persistent search 122
Enabling and disabling event notification 123
Enabling event notification 124
Disabling event notification. 125
Enabling and disabling transaction support . . . 125
Enabling transaction support 125
Disabling transaction support 127
Adding and removing suffixes 127
Creating or adding suffixes. 127
Removing a suffix 128
Tombstone - Record deleted entries 129
Using Web Administration 130
Using the command line 130
Managing cache properties 131
Entry cache 131
Filter cache 131
ACL cache 132
Group members' cache 132
Adding attributes to and removing attributes from the attribute cache 133
DB2 password monitoring 136
Using Web Administration 136
Using command line 137

Chapter 11. Securing directory communications 139
Configuring security settings 139
Using Web Administration 139
Using the command line 141
Transaction Layer Security 141
Secure Sockets Layer 141
Using gskcapicmd 148
Using ikeyman 152
Setting the key database. 161
Using Web Administration 161
Using the command line 161
PKCS#11 161
How to configure the server to use PKCS#11 interface 162
Setting the level of encryption for SSL and TLS communications 163
Using Web Administration 163
Using the command line 164
Certificate revocation verification. 165
Using Web Administration 165
Using the command line 166

Chapter 12. Securing directory access 167
Password encryption 167
Using Web Administration 170
Using the command line 170
Setting password policy 170
Global Password Policy 171
Group Password Policy 171
Individual Password Policy. 171

Password Policy Evaluation 172
Password policy attributes 178
Summary of default settings 178
Password Guidelines 179
Setting the administration password and lockout policy 181
Unlocking administrative accounts 182
Setting the global password policy 182
Changing password when pwdsafemodify is set 186
Setting Kerberos 187
Using Web Administration 188
Using the command line 189
Using Kerberos 189
Identity mapping for Kerberos 189
Configuring the DIGEST-MD5 mechanism. . . . 191
Using Web Administration 191
Using the command line 192
Pass-through authentication 193
Scenarios pertaining to pass-through authentication 195
Using Web Administration Tool 198
Using command line 200
Creating the administrative group 201
Administrative Roles 201

Chapter 13. Referrals 211

Setting up referrals to other LDAP directories . . 211
Using the referral object class and the ref attribute 212
Binding with a distributed namespace 213
An example of distributing the namespace through referrals 213
Creating default referrals 215
Using Web Administration 215
Using the command line 216
Modifying referrals 217
Using Web Administration 217
Using the command line 217
Removing referrals 218
Using Web Administration 218
Using the command line 218

Chapter 14. Replication 219

Replication terminology 219
Replication topology 221
Overview of replication 222
Simple replication 222
Cascading replication. 223
Peer-to-peer replication 223
Gateway replication 224
Partial replication 225
Replication conflict resolution 225
Replication error handling 228
Replication agreements 229
Things to consider before configuring replication 233
Replicating schema and password policy updates 234
Creating a master-replica topology 234
Using Web Administration 236
Using the command line 241
Setting up a simple topology with peer replication 244
Using Web Administration 244
Using the command line 247
Creating a master-forwarder-replica topology. . . 250
Changing the replica to a forwarding server . . 251
Setting up a complex topology with peer replication 256
Using the command line 257
Unconfiguring a master/replica configuration . . 263
Setting up a gateway topology 265
Using Web Administration 267
Using the command line 269
Partial Replication. 275
Using Web Administration Tool 276
Using command line 278
Examples of replication filter 279
Excluding replication topology information . . . 280
Recovery procedures 281
Required recovery information 281
Recovering from a single-server failure 283
Recovering from a catastrophic failure 284
Multi-threaded replication 284
Replication error table 285
Web Administration tasks for managing replication 286
Replicating subtrees 286
Working with credentials 288
Managing topologies 291
Modifying replication properties 301
Creating replication schedules 303
Managing queues 305
Command line tasks for managing replication . . 307
Specifying a supplier DN and password for a subtree 307
Viewing replication configuration information 307
Monitoring replication status 308
Creating gateway servers 310

Chapter 15. Distributed directories 313

The Proxy server 313
Splitting data within a subtree based on a hash of the RDN using a proxy server. 317
DN Partition plug-in 318
Using the command line 319
The distributed directory setup tool 319
Synchronizing information 320
Partition entries 321
Setting up a distributed directory with proxy server 322
Setting up the back-end servers 322
Setting up the proxy server. 324
Schema updates in a distributed directory. . . . 332
Password policy in a distributed directory. . . . 334
Failover and load balancing 334
Auto failback 335
Health Check Feature 336
Health Check Status Interval Configuration . . 337
High consistency and failover when high consistency is configured 337
Weighted Prioritization of backend servers . . . 337
Failover between proxy servers 338
Setting up backup replication for a distributed directory with proxy servers 338

Server groups 339
Creating an LDIF file for your data entries . . 340
Setting up the replication topology 341
Setting up a topology for global policies . . . 342
Setting up proxy servers. 343
Partitioning the data 343
Loading the partitioned data 343
Monitor Search 344
Transactions in proxy server 347
Starting replication 347

Chapter 16. Directory Server backup and restore 349
Methods that back up complete directory server instance information 349
Methods that back up database information only 350
Enhanced backup 351
Using Web Administration 351
Configure directory server backup 353
Perform directory server backup 354
Schedule directory server backup. 355
Perform directory server restore 356
Using the command line 356

Chapter 17. Logging Utilities 359
Default log paths 359
Log management tool 360
Specifying custom location for an instance's idslogmgmt.log file 360
Default log management 361
Modifying global log settings 361
Using the Web Administration Tool 362
Using the command line 362
Modifying administration server log settings . . . 362
Using Web Administration Tool 363
Using the command line 364
Enabling the administration server audit log and modifying administration audit log settings . . . 364
Using Web Administration Tool 365
Using the command line 366
Disabling the administration server audit log. . . . 367
Using Web Administration 367
Using the command line 367
Configuring preaudit records 368
Enabling the server audit log and modifying server audit log settings 368
Using Web Administration 372
Using the command line 374
Disabling the audit log 375
Using Web Administration 375
Using the command line 375
Performance profiling 376
Performance profiling through the independent trace facility 376
Auditing for performance profiling 377
Modifying bulkload log settings 378
Using Web Administration 378
Using the command line 379
Modifying tools log settings 380
Using Web Administration 380
Using the command line 381
Modifying DB2 log settings. 381
Using Web Administration 381
Using the command line 382
Modifying lost and found log settings 382
Using Web Administration 382
Using the command line 384
Modifying the server log 384
Using Web Administration 384
Using the command line 385
Start/stop server tracing. 386
Using Web Administration 386
Viewing logs 386
View logs using Web Administration 386
View logs using the command line 387
Log integration into CBE and CARS format . . . 391
Log management tool for CBE, CEI, and CARS features 392
Entries for log management 393
CARS Reports 393
Configuring log management attributes for CBE and CARS 394

Part 3. Directory Management . . . 399

Chapter 18. Working with directory entries 401
Browsing the tree 401
Adding an entry 402
Using Web Administration 402
Using the command line 403
Multiple values for attributes 403
Binary data for attributes 403
Using Web Administration 403
Using the command line 404
Language tags 405
Attributes that cannot have associated language tags 406
Language tag values for attributes 407
Searching for entries containing attributes with language tags 407
Removing a language tag descriptor from an entry 408
Deleting an entry 408
Using Web Administration 408
Using the command line 409
Modifying an entry 409
Using Web Administration 409
Using the command line 410
Copying an entry 410
Using Web Administration 411
Using the command line. 411
Editing access control lists for an entry 412
Adding an auxiliary object class 412
Using Web Administration 412
Using the command line 413
Deleting an auxiliary object class 413
Using Web Administration 413
Using the command line 413
Searching the directory entries. 414

Search filters 414
Options 416

Chapter 19. Access control lists . . . 419
Overview. 419
EntryOwner information 419
Access control information 419
ACL type usage scenarios 421
The access control attribute syntax 422
Subject 423
Pseudo DNs. 424
Object filter 425
Rights 425
Propagation 427
Access evaluation 428
Working with ACLs 430
Using the Web Administration Tool utility to manage ACLs 430
Using the command line utilities to manage ACLs 436
Subtree replication considerations 440

Chapter 20. Groups and roles 441
Groups 441
Static groups 441
Dynamic groups 442
Nested groups 443
Hybrid groups 443
Determining group membership 444
Group object classes 448
Group attribute types. 448
Creating a static group entry 449
Creating a dynamic group entry 450
Creating a nested group entry 451
Verifying the group task. 452
Managing members of group entries 453
Adding a member to a group entry 453
Editing a member entry in a group 454
Removing a member from a group entry . . . 454
Managing memberships for an entry 454
Adding a group membership 455
Removing a group membership from an entry 455
Editing a memberURL in a dynamic group . . . 456
Roles 456

Chapter 21. Managing search limit groups 459
Creating a search limit group 459
Using Web Administration 459
Using the command line 461
Modifying a search limit group 461
Using Web Administration 461
Using the command line 461
Copying a search limit group 462
Using Server Administration 462
Using the command line 462
Removing a search limit group 462
Using Web Administration 462
Using the command line 462

Chapter 22. Managing a proxy authorization group. 463
Creating a proxy authorization group 463
Using Web Administration 464
Using the command line 464
Modifying a proxy authorization group 465
Using Server Administration 465
Using the command line 465
Copying a proxy authorization group 466
Using Server Administration 466
Using the command line 466
Removing the proxy authorization group 466
Using Web Administration 466
Using the command line 466

Part 4. User-related tasks 469

Chapter 23. Realms, templates, users, and groups 471
Creating a realm 471
Creating a realm administrator 471
Creating the realm administration group . . . 471
Creating the administrator entry 472
Adding the administrator to the administration group.. 472
Creating a template 473
Adding the template to a realm 475
Creating groups 475
Adding a user to the realm. 475
Managing realms 476
Adding a realm 476
Editing a realm. 476
Removing a realm. 477
Editing ACLs on the realm 477
Managing templates 477
Adding a user template 477
Editing a template. 479
Removing a template. 479
Editing ACLs on the template 479
Managing users 480
Adding users 480
Finding users within the realm 480
Editing a user's information 480
Copying a user. 481
Removing a user 481
Managing groups 481
Adding groups. 481
Finding groups within the realm 481
Editing a group's information 482
Copying a group 482
Removing a group. 482

Part 5. Appendixes 483

Appendix A. Error codes 485

Appendix B. Object Identifiers (OIDs) and attributes in the root DSE 491

Attributes in the root DSE 491
OIDs for supported and enabled capabilities . . . 493
OIDs for ACI mechanisms 501
OIDs for extended operations 501
OIDs for controls 505

Appendix C. LDAP data interchange format (LDIF) 509
LDIF example 509
Version 1 LDIF support 510
Version 1 LDIF examples 510
IANA character sets supported by platform . . . 511

Appendix D. ASCII characters from 33 to 126. 515

Appendix E. IPv6 support 517

Appendix F. Simple Network Management Protocol agent 519
SNMP Logging 523
Using the command line – idssnmp 524

Appendix G. Active Directory synchronization 525
Steps for using Active Directory synchronization 526
Files used by Active Directory synchronization . . 527
Running Active Directory synchronization. . . . 530
Configuring Active Directory synchronization to use an SSL connection to Active Directory. . . . 531

Appendix H. Additional information on password policy 533
Password policy operational attributes 533
Interoperability support for password policy response control 534
Password policy queries. 534
Overriding password policy and unlocking accounts 534
Replicating multiple password policy attributes 536
Replicating password policy operational attributes 536
Forcing an add or update for an entry 537

Appendix I. Required attribute definitions for IBM Tivoli Directory Server 539

Appendix J. Synchronizing two-way cryptography between server instances 577

Appendix K. Filtered ACLs and non-filtered ACLs – sample LDIF file . 579

Appendix L. Dynamically-changed attributes 587

Appendix M. IBM Tivoli Directory Server backup and restore 591
Introduction 591
Tivoli Directory Server directory schema and database definitions 591
Tivoli Directory Server directory schema . . . 591
Tivoli Directory Server directory database and tablespaces 592
Tivoli Directory Server change log database and tablespaces 595
Overview of backup and restore procedures for LDAP 595
Examples of offline backup and restore procedure for a directory database 596
Replication considerations 597
Overview of online backup and restore procedures for Tivoli Directory Server 597
Log management 597
Using DB2 backup and restore 599

Appendix N. Setting up SSL security – SSL scenarios 609
Using HTTPS with Embedded WebSphere Application Server Version 7.x. 609
Creating secure connections between IBM Tivoli Directory Server and the IBM Tivoli Directory Server Web Administration Tool 610
Setting up an SSL connection between a IBM Tivoli Directory Srever C-based client and IBM Tivoli Directory Server 616

Appendix O. High Availability Scenarios 621

Appendix P. Referential integrity plug-in 623

Appendix Q. Guidelines for interoperability between IBM Tivoli Directory Server and z/OS IBM Tivoli Directory Server 625
Schema considerations 625
Import or export of directory entries 627
Functional considerations 628

Appendix R. Notices 629
Trademarks 631

Glossary 633
Glossary 633

Index 639

About this book

IBM® Tivoli® Directory Server is the IBM implementation of Lightweight Directory Access Protocol for supported Windows®, AIX®, Linux® (System x®, System z®, System p®, and System i®), Solaris, and Hewlett-Packard UNIX® (HP-UX) (Itanium®) operating systems.

IBM Tivoli Directory Server Version 6.3 Administration Guide describes how to perform administrator tasks through the Web Administration Tool and the command line.

Intended audience for this book

This book is for administrators of IBM Tivoli Directory Server Version 6.3.

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

Publications

This section lists publications in the IBM Tivoli Directory Server version 6.3 library and related documents. The section also describes how to access Tivoli publications online and how to order Tivoli publications.

IBM Tivoli Directory Server version 6.3 library

The following documents are available in the IBM Tivoli Directory Server version 6.3 library:

- *IBM Tivoli Directory Server Version 6.3 What is New for This Release*, GC27-2746-00

  Provides information about the new features in the IBM Tivoli Directory Server Version 6.3 release.
- *IBM Tivoli Directory Server Version 6.3 Quick Start Guide*, GI11-9351-00

  Provides help for getting started with IBM Tivoli Directory Server 6.3. Includes a short product description and architecture diagram, as well as a pointer to the product Information Center and installation instructions.
- *IBM Tivoli Directory Server Version 6.3 System Requirements*, SC27-2755-00

  Contains the minimum hardware and software requirements for installing and using IBM Tivoli Directory Server 6.3 and its related software. Also lists the supported versions of corequisite products such as DB2® and GSKit.
- *IBM Tivoli Directory Server Version 6.3 Installation and Configuration Guide*, SC27-2747-00

  Contains complete information for installing, configuring, and uninstalling IBM Tivoli Directory Server. Includes information about upgrading from a previous version of IBM Tivoli Directory Server.
- *IBM Tivoli Directory Server Version 6.3 Administration Guide*, SC27-2749-00

  Contains instructions for performing administrator tasks through the Web Administration Tool and the command line.
- *IBM Tivoli Directory Server Version 6.3 Command Reference*, SC27-2753-00

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

© Copyright IBM Corp. 2002, 2010

- *IBM Tivoli Directory Server Version 6.3 Server Plug-ins Reference*, SC27-2750-00

  Contains information about writing server plug-ins.
- *IBM Tivoli Directory Server Version 6.3 Programming Reference*, SC27-2754-00

  Contains information about writing Lightweight Directory Access Protocol (LDAP) client applications in C and Java™.
- *IBM Tivoli Directory Server Version 6.3 Performance Tuning and Capacity Planning Guide*, SC27-2748-00

  Contains information about tuning the directory server for better performance. Describes disk requirements and other hardware needs for directories of different sizes and with various read and write rates. Describes known working scenarios for each of these levels of directory and the disk and memory used; also suggests rough rules of thumb.
- *IBM Tivoli Directory Server Version 6.3 Problem Determination Guide*, GC27-2752-00

  Contains information about possible problems and corrective actions that can be taken before contacting IBM Software Support.
- *IBM Tivoli Directory Server Version 6.3 Messages Guide*, GC27-2751-00

  Contains a list of all informational, warning, and error messages associated with IBM Tivoli Directory Server 6.3.
- *IBM Tivoli Directory Server Version 6.3 White Pages*, SC27-2756-00

  Describes the Directory White Pages application, which is provided with IBM Tivoli Directory Server 6.3. Contains information about installing, configuring, and using the application for both administrators and users.

Related publications

The following documents also provide useful information:

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

  IBM Tivoli Directory Server Version 6.3 uses the Java Naming and Directory Interface (JNDI) client from Sun Microsystems. See this document for information about the JNDI client.

Accessing terminology online

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

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

Accessing publications online

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

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

IBM posts publications for this and all other Tivoli products, as they become available and whenever they are updated, to the Tivoli Documentation Central Web site at http://www.ibm.com/tivoli/documentation.

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

Ordering publications

You can order many Tivoli publications online at http://www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss.

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

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

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

1. Go to http://www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss.
2. Select your country from the list and click **Go**.
3. Click **About this site** in the main panel to see an information page that includes the telephone number of your local representative.

Accessibility

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

Visit the IBM Accessibility Center at http://www.ibm.com/alphaworks/topics/accessibility/ for more information about the IBM commitment to accessibility.

For additional information, see the Accessibility Appendix in *IBM Tivoli Directory Server Version 6.3 Installation and Configuration Guide*.

Tivoli technical training

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

Tivoli user groups

Tivoli user groups are independent, user-run membership organizations that provide Tivoli users with information to assist them in the implementation of Tivoli Software solutions. Through these groups, members can share information and learn from the knowledge and experience of other Tivoli users. Tivoli user groups include the following members and groups:

- 23,000+ members
- 144+ groups

Access the link for the Tivoli Users Group at www.tivoli-ug.org.

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:

Online

Access the Tivoli Software Support site at http://www.ibm.com/software/sysmgmt/products/support/index.html?ibmprd=tivman. Access the IBM Software Support site at http://www.ibm.com/software/support/probsub.html .

IBM Support Assistant

The IBM Support Assistant is a free local software serviceability workbench that helps you resolve questions and problems with IBM software products. The Support Assistant provides quick access to support-related information and serviceability tools for problem determination. To install the Support Assistant software, go to http://www.ibm.com/software/support/isa.

Troubleshooting Guide

For more information about resolving problems, see the *IBM Tivoli Directory Server Version 6.3 Problem Determination Guide*

Conventions used in this book

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

Typeface conventions

This book uses the following typeface conventions:

Bold

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

Italic

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

`Monospace`

- Examples and code examples
- File names, programming keywords, and other elements that are difficult to distinguish from surrounding text
- Message text and prompts addressed to the user

- Text that the user must type
- Values for arguments or command options

Operating system-dependent variables and paths

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

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

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

Part 1. Directory overview

© Copyright IBM Corp. 2002, 2010

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 a specific criteria and 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 an 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

© Copyright IBM Corp. 2002, 2010

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 19, "Access control lists," on page 419 for more information.

Chapter 2. IBM Tivoli Directory Server

IBM Tivoli Directory Server 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. 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:
  - Searchable 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. 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.
- 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 139 and “Configuring the DIGEST-MD5 mechanism” on page 191.

  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 141 and “Transaction Layer Security” on page 141.

  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

© Copyright IBM Corp. 2002, 2010

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 19, "Access control lists," on page 419.

Auditing

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 server audit log and modifying server audit log settings" on page 368.

Security roles

IBM Tivoli Directory Server supports five different security roles.

Primary directory administrator

The Primary directory administrator is associated with a specific user account. There is only one Primary directory administrator account for the LDAP server. The Primary directory administrator has full rights to manage the LDAP server. The Primary directory administrator is created during product installation and configuration. The Primary directory administrator consists of a user ID and a password and predefined authorization to manipulate the entire directory. The Primary 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 Primary 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. 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. Server administrative group members are explicitly assigned various roles that define the tasks that a group member is authorized to perform. These administrative roles include such specialized roles as Password Administrator and Server Start/Stop Administrator. For more information, see "Creating the administrative group" on page 201.

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.

Note: Global administrative group members have the authority to send the administrative control.

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.
- 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 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 primary administrator. The user password policy configuration is stored within the LDAP tree and can be modified by the primary administrator or a member of the administrative group. The attribute values can be changed only when binding as administrator to IBM Tivoli Directory Server. Tivoli Directory Server provides three types of password policies: individual, group, and global password policies. For more information, see "Setting password policy" on page 170.

Password encryption

IBM Tivoli Directory Server 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:

- crypt
- MD5
- SHA-1
- Salted SHA-1
- SHA-2 (SHA 224, SHA 256, SHA 384, and SHA 512)
- Salted SHA-2 (SSHA 224, SSHA 256, SSHA 384, and SSHA 512)

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.

MD5 Passwords are encrypted by the MD5 Message Digest 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.

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

SHA-2
Passwords are encrypted by the SHA-2 family of encrypting algorithm before they are stored in the directory. The supported encryption schemes under the SHA-2 family of encryption algorithm are:

- SHA-224
- SHA-256
- SHA-384
- SHA-512

Salted SHA-2
Passwords are encrypted by the Salted SHA-2 family of encrypting algorithm before they are stored in the directory. The supported encryption schemes under the Salted SHA-2 family of encryption algorithm are:

- SSHA-224
- SSHA-256
- SSHA-384
- SSHA-512

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 IBM Tivoli Directory Server. 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 – A 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.
- administration server (idsdiradm) – Enables remote management of an instance of IBM Tivoli Directory Server. It must be installed on the machine where 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.
- Multiple instances – Enables a user to have more than one directory instance on a server.

- 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.
- Attribute encryption - Enables local administrative group members who are assigned DirDataAdmin and SchemaAdmin roles to specify attributes that are to be encrypted in the directory database using a subset of the encryption schemes supported for password information. For more information, see "Encrypted Attributes" on page 57
- Pass-through authentication- A mechanism using which if a client attempts to bind to a directory server and if the user credential is not available locally, then the server attempts to verify the credential from another external directory server or a pass-through server on behalf of the client. For more information, see "Pass-through authentication" on page 193.
- SNMP for server management- The SNMP agent can be used with the IBM Tivoli Directory Integrator assembly line to monitor and report the performance and wellness information of the directory server.
- Active directory synchronization- A tool for synchronizing users and groups between an existing Microsoft® Active Directory and an IBM Tivoli Directory Server instance. This feature supported from IBM Tivoli Directory Server 6.1 release onwards.

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> | "+" | "<" |  ">"
```

© Copyright IBM Corp. 2002, 2010

```
          | "#" | ";"

<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=sample
```

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 6.3 Programming Reference* for more information.

Part 2. Server Administration

© Copyright IBM Corp. 2002, 2010

Chapter 4. Directory administration server

The directory administration server (idsdiradm) enables remote management of an instance of IBM Tivoli Directory Server. It must be installed on the machine where IBM Tivoli Directory Server is installed and must be running continuously. The directory administration server accepts requests by way of LDAP extended operations and supports starting, stopping, restarting, and status monitoring of IBM Tivoli Directory Server.

The directory administration server does not support any access to the configuration file or the configuration backend. However, it supports dynamic update requests. By supporting dynamic update requests, the server ensures that its in memory configuration remains in sync with the server's configuration. For instance, if an update is made to the configuration file that impacts both the admin server and the directory server, the dynamic update request is sent to both the admin server and the directory server.

The admin server will not check the bind DN and password against the configuration file every time there is a bind request. Instead, it will issue a config update request for any changes to admin DN and password to take effect.

Note: All Admin Group members can bind to the admin server.

By default, the first instance of the IBM Directory administration server listens on two ports, port 3538 for non-SSL connections and port 3539 for SSL connections, if SSL communication is enabled.

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

To start the directory administration server, run the program idsdiradm from any command prompt. See "Starting an instance of the directory administration server" on page 20.

Notes:

1. The administration server supports auditing version 3 only.
2. The administration server auditing is enabled for all operations by default.
3. If you enable SSL communication, the directory administration server must be stopped and restarted for SSL to take effect. See "Using Web Administration" on page 139.
4. If you change the time zone on your Windows machine, you need to restart the server and the administration server in order for the server and administration server to recognize the time change. This ensures that the time stamps in the administration server's logs match the time stamps in the server's logs.
5. The administration server supports all read log access extended operations. This means that log files can be read remotely even when the directory server is not running.

© Copyright IBM Corp. 2002, 2010

Starting an instance of the directory administration server

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

To start an instance of the administration server 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 V6.3 - <***instancename***> Admin Server**, click **Start**.

Note: On Linux SLES systems, the admin server must not be started from inittab. Instead, start the admin server manually from the command line. See the **idsdiradm** command information in the *IBM Tivoli Directory Server Version 6.3 Command Reference* for more information.

Stopping an instance of the directory administration server

To stop an instance of the administration server 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 server. This command is not platform specific. See the **ibmdirctl** command information in the *IBM Tivoli Directory Server Version 6.3 Command Reference* for more 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, in the services panel, you can select **IBM Tivoli Directory Server Instance V6.3 - <***instancename***> Admin Server**, and click **Stop**.

Starting the directory server instance at operating system startup

If you want the directory server instance to start automatically at operating system startup, use one of the following sections:

- "Autostart on Windows systems"
- "Autostart on AIX, Linux, and Solaris systems" on page 21

Autostart on Windows systems

On Windows systems, the server (the **idsslapd** process) is started manually through the Services window or by using the **idsslapd** command. If you try to start the server automatically by updating the **Startup Type** in the Services window to **Automatic**, errors occur when you restart the computer. This is because DB2 must be running before the **idsslapd** process can start.

If you want the server to start automatically, use the following procedure:

1. Open the Services window in one of the following ways:

- Click **Start -> Run**, and then type `services.msc` in the **Open** field. Click **OK**.
- Click **Start -> Settings -> Control Panel**. In the Control Panel, double-click **Administrative Tools**, and then double-click **Services**.

2. Find the DB2 Service name for the directory server instance you want to autostart. The service starts with `DB2 - TDSV63DB2 -` (for example, `DB2 - TDSV63DB2 - DSRDBM01`). Double-click the service and look in the **Display name** field. Make a note of the name that comes after `DB2 - TDSV63DB2 -` . (In this example, make a note of `DSRDBM01`.) You will use this DB2 name later.
3. Find the service for the directory server instance you want to autostart. The service starts with `IBM Tivoli Directory Server Instance 6.3 -` (for example, `IBM Tivoli Directory Server Instance 6.3 - dsrdbm01`). Double-click the service and look in the **Display name** field. Make a note of the name that comes after `IBM Tivoli Directory Server Instance 6.3 -` , with `idsslapd-` prepended. (In this example, make a note of `idsslapd-dsrdbm01`.) You will use this instance name later.
4. While still in the IBM Tivoli Directory Server Instance V6.3 - *instance_name* window, set the **Startup type** field to **Automatic**, and then click **OK**. The service will be set to start automatically at system startup.
5. Close the Services window.
6. To open the Windows Registry, click **Start -> Run**. Type `regedit` in the **Open** field. Click **OK**. The Registry Editor window is displayed.
7. On the left side of the window, go to the following: **My Computer -> HKEY_LOCAL_MACHINE -> SYSTEM -> CurrentControlSet -> Services**.
8. Find the service corresponding to directory server instance. (In the example, this is **idsslapd-dsrdbm01**.) Click the service.
9. On the right side of the window, double-click the **DependOnService** attribute.
10. In the Edit Multi-String window, add the DB2 service name (**DSRDBM01** in our example) under **LanmanServer**, and then click **OK**. (This adds a dependency on the DB2 service.)
11. Exit the Registry Editor.

After you restart the system, the directory server instance will start automatically.

Note: The proxy server does not require DB2. Therefore, to make the proxy server start automatically at system startup, use only steps 1, 3, and 4.

Autostart on AIX, Linux, and Solaris systems

To start the directory server instance at operating system startup on AIX, Linux, and Solaris systems, you must edit the /etc/inittab file and add a line. The inittab file specifies which processes are started at system startup and during normal operation. An entry in the inittab file has the following format:

```
id:runlevels:action:process
```

where:

- *id* is a 1-4 digit unique ID in the file.
- *runlevels* indicates the runlevel mode of the operating system in which the process should start automatically. Runlevel refers to the mode of operation of an AIX, Linux, or Solaris operating system. Runlevel configuration differs among operating systems. Refer to the operating system manual for your operating system for specific runlevel configuration details.

- *action* describes the action to be taken. In this case it will have the value `boot` to indicate that the process should be started at system startup.
- *process* is the process to be started.

Add one of the following lines to your inittab file:

For AIX systems

```
srv1:2:once:/opt/IBM/ldap/V6.3/sbin/idsslapd -I server_name > /dev/null 2>&1
#Autostart IBM LDAP Directory Server Instance
```

For Linux systems

```
srv1:2345:once:/opt/ibm/ldap/V6.3/sbin/ibmslapd -I server_name > /dev/null 2>&1
#Autostart IBM LDAP Directory Server Instance
```

For Solaris systems

```
srv1:234:once:/opt/IBM/ldap/V6.3/sbin/ibmslapd -I server_name > /dev/null 2>&1
#Autostart IBM LDAP Directory Server Instance
```

where *server_name* is the name of the directory server instance you want to start automatically.

After the entry is added to the /etc/inittab file, the directory server instance (full or proxy) is ready for autostart after system restart.

Be sure that all of the full directory servers are up and running before the proxy server starts. If you have full servers and a proxy server on the same computer, add a delay between the full server and the proxy server startup. In the following example, this done in the lines starting with `srv3:2345:wait:` and `srv4:2345:wait:`.

```
#Autostart IBM LDAP Directory Server Instance
srv1:2345:wait:/opt/IBM/ldap/V6.3/sbin/idsslapd -I server1 > /dev/null 2>&1
#Autostart IBM LDAP Directory Server Instance
srv2:2345:wait:/opt/IBM/ldap/V6.3/sbin/idsslapd -I server2 > /dev/null 2>&1
#Autostart IBM LDAP Directory Server proxy instance
srv3:2345:wait:/opt/IBM/ldap/V6.3/sbin/idsslapd -I proxy -k > /dev/null 2>&1
#Autostart IBM LDAP Directory Server proxy instance
srv4:2345:wait:/opt/IBM/ldap/V6.3/sbin/idsslapd -I proxy > /dev/null 2>&1
```

where `server1` and `server2` are full directory server instances, and `proxy` is a proxy server instance.

Chapter 5. Configuration only mode

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

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. To start the server in configuration only mode, select the **Start/Restart in configuration only mode** check box.

© Copyright IBM Corp. 2002, 2010

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 the **ibmdirctl** command information in the *IBM Tivoli Directory Server Version 6.3 Command Reference* for more information.

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)

IBM Tivoli Directory Server Web Administration Tool is installed on an application server, such as the embedded version of IBM WebSphere® Application Server - Express® (WAS) included with 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.

Note: If you use the latest version of Web Administration Tool to administer an older version of the directory server instance, some of the panels may not be visible.

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

- Set the administration DN and password to be able to start a given server.
- If the server is not configured as a proxy server, configure a database to be able to start a given server in a state other than the configuration only mode.
- Ensure that either the server or the administration server is running.

See the *IBM Tivoli Directory Server Version 6.3 Installation and Configuration Guide* and Chapter 4, "Directory administration server," on page 19 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 application server to use the Web administration tool

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

To start the Web application server if you are using Embedded WebSphere Application Server as your Web application server, use one of the following files, which contain the command to start the Web application server:

- On Windows systems, *installpath*\idstools\bin\startWebadminApp.bat
- On AIX, Linux, and Solaris systems, *installpath*/idstools/bin/startWebadminApp

where *installpath* is the path where you installed Tivoli Directory Server. This path is:

- By default on Windows: c:\Program Files\IBM\LDAP\V6.3
- On AIX and Solaris: /opt/IBM/ldap/V6.3
- On Linux: /opt/ibm/ldap/V6.3

© Copyright IBM Corp. 2002, 2010

Note: Files are also provided with commands for starting the Web application server for Directory White Pages. The files are:

For Directory White Pages

For Windows systems, *installpath*\idsapps\bin\startWpgsApp

For AIX, Linux, and Solaris systems, *installpath*/idsapps/bin/startWpgsApp

Starting the Web Administration Tool

To start the Web administration tool:

1. After the Web application server is started, you can start the Web administration tool in several ways:
   - From a Web browser, type the following address:

     http://localhost:12100/IDSWebApp/
   - On Windows systems, click one of the following:
     - For non-SSL: **Start -> All Programs -> IBM Tivoli Directory Server 6.3-> Web Administration Tool**
     - For SSL: **Start -> All Programs -> IBM Tivoli Directory Server 6.3 -> Web Administration Tool (secure)**
   - Load the following file into a Web browser:
     - On Windows systems:
       - *installpath*\idstools\bin\idswebadmin.html for non-SSL
       - *installpath*\idstools\bin\idswebadminssl.html for SSL

       installpath is the path where you installed IBM Tivoli Directory Server. By default this is C:\Program Files\IBM\LDAP\V6.3
     - On AIX, Linux, or Solaris systems:
       - *installpath*/idstools/bin/idswebadmin.html for non-SSL
       - *installpath*/idstools/bin/idswebadminssl.html for SSL

       installpath is /opt/IBM/ldap/V6.3 for AIX, and Solaris systems, or /opt/ibm/ldap/V6.3 for Linux systems.

     This file points to localhost. You can modify it as needed.

   The 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. You can use the **ipconfig** command to find the IP address of the computer.
2. Log in to the console as the console administrator, using the following instructions:
   a. In the **User ID** field, type `superadmin`.
   b. In the **Password** field, type `secret`.
   c. Click **Login**.

   The IBM Tivoli Directory Server Web Administration Tool console is displayed.
3. Add a server to the console, using the following instructions:
   a. Do one of the following:
      - Click **Manage console servers** in the right side of the window.

- Expand **Console administration** in the navigation area, and then click **Manage console servers**.

A table of server host names and port numbers is displayed.

b. Click **Add**.

c. Enter a unique name in the **Server name** field to identify a registered IBM Tivoli Directory Server instance running on a specific host name or IP address and server port.

d. Type the hostname or the IP address of the server in the **Hostname** field.

e. Specify the server port number in the **Port** field.

f. Specify whether the console will be communicating with the server using Secure Sockets Layer (SSL). The **Enable SSL encryption** check box will display only if you installed an SSL-enabled version of IBM(R) Tivoli Directory Server Web Administration Tool.

g. Select the **Administration server supported** check box to enable the Administration port control.

h. Specify the Administration server port number in the **Administration port** field.

i. Click **OK** to apply the changes or click **Cancel** to exit the panel without making any changes.

4. Click **Logout** in the navigation area.

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 201) or an LDAP user:

- At the IBM Tivoli Directory Server login page select the LDAP Server Name or IP address and port for your machine from the drop-down menu.
- Enter the administrator DN and the corresponding password for that administrator DN (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. Similarly, to login as an admin group member or as a normal DIT user, enter the DN of the user and the corresponding password. For example, if the DIT user is cn=Tom Brown, o=sample then you need to enter the login name as cn=Tom Brown, o=sample.
- 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.
- To log in to the console admin, click the Log in to Console Admin link.

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 authority of the user logging in or the capabilities of the server the user is 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. The server status area also displays the user DN.

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 from 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.

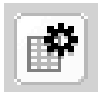
Using tables in the Web Administration Tool

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 **Select All** icon to select all items in the table.
- Click the **Deselect All** icon to deselect all selected items in the table.
- Click the **Show Filter Row** icon to display filter rows for every column in the table. See "Filtering" on page 31 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 31 for more information about filtering.
- Click the **Clear All Filters** icon clear all filters set for the table. See "Filtering" on page 31 for more information.
- Click the **Edit Sort** icon to sort the information in the table. See "Sorting" on page 30 for more information.
- Click the **Clear All Sorts** icon clear all sorts set for the table. See "Sorting" on page 30 for more information.
- Click the **Collapse Table** icon to hide the table data.
- Click the **Expand Table** icon to display the table data.
-  Click the Configure Columns icon to rearrange the columns in the table. See "Reordering" on page 31 for more information.

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 30 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.

Restore Defaults

Default settings of table filter and sorting features will be restored when this feature is invoked. The default behavior is that the filter row will be hidden and any currently set sort or filter criteria will be reset. No icon is provided for this feature in the table toolbar, it can be accessed using the table user action list. To restore defaults, do the following on the table:

- Click the **Select Action** drop-down menu, then select **Restore Defaults** and click **Go**.

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 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.

Reordering

To change the order in which the columns appear in the table or to remove any columns from the table use the Configure Columns option. To reorder columns in the table, do the following:

1. Do one of the following:
   - Click the **Configure Columns** icon on the table.
   - Click the **Select Action** drop-down menu, then select **Configure Columns** and click **Go**.
2. A section with a list containing all the column names in the table along with the check boxes are displayed. In this section, do the following:
   - To display or remove the columns getting displayed, select or clear the check boxes adjacent to column names.
   - To change the order in which a particular column appear in the table, select the column name and click the up or down arrow button as required.
3. After you have finished, do one of the following:
   - Click **OK** to save the changes made.

- Click **Cancel** to return to the panel without making any changes.

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.

© Copyright IBM Corp. 2002, 2010

2. Click **Manage console servers**. A table for listing of server host names and port numbers is displayed.
3. Click **Add**.
4. Specify a unique name that identifies a registered IBM Tivoli Directory Server instance running on a specified host name or IP address and server port. The server name is displayed in the LDAP Server Name list on the Directory server login panel. If a name is not provided in the Server name field, the hostname:port combination would be displayed for the server instance in the LDAP Server Name list on the Directory server login panel.
5. Enter the host name address or the IP address of the server. For example *servername.austin.ibm.com*
6. Select the **Administration server supported** check box to enable the Administration port control.
7. 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.
8. Specify if the server is SSL enabled. Ensure that you complete step 5 on page 35 under **Managing console properties**.
9. 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 35 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 ikeyman" on page 152 and "Secure Sockets Layer" on page 141 for information about key databases and SSL.

Manage properties for webadmin searches

Users can use the Manage properties for webadmin searches panel to configure the search settings for web admin searches. However, if the limit number of attribute values control is not supported then the Manage properties for webadmin searches panel will not be displayed.

To configure the search settings for web admin searches:

1. Expand **Console administration** in the navigation area.
2. Click **Manage properties for webadmin searches**.
3. Specify the maximum number of attributes to return for each entry. If you click **Number of attributes**, you must enter a number. Otherwise, click **Unlimited**.
4. Specify the maximum number of values to return for each attribute. If you click **Number of values**, you must enter a number. Otherwise, click **Unlimited**.
5. Click **OK** to save the changes and to return to the Introduction panel.

When you have finished setting up the console, click **Logout** to exit. See "Logging off the console" on page 28 for more information.

Viewing scenario-based help files in the Web Administration Tool

To view the scenario-based help files in the Web Administration Tool, do the following:

1. Install IBM Tivoli Directory Server version 6.3 and create a directory server instance. See the *IBM Tivoli Directory Server version 6.3 Installation and Configuration Guide* for more information.
2. Deploy the Web Administration Tool into Embedded WebSphere Application Server.
3. Log on to the Console administration login panel of the Web Administration Tool and add your directory server instance.
4. Log on to your directory server using the Directory server login panel of the Web Administration Tool.

5. Click the **?** icon at the top right corner of the Work area of the Web Administration Tool. This will launch the Table of Contents help file.
6. The example scenarios are listed at the end of the Table of Contents help file.

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. An 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 18, “Working with directory entries,” on page 401 for additional information about entries.

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

IBM Tivoli Directory Server 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 Tivoli Directory Server Version 6.3 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 39.)
- attributeTypes (See “Working with attributes” on page 46.)
- IBMAttributeTypes (See “The IBMAttributeTypes attribute type” on page 47.)
- matching rules (See “Equality matching rules” on page 48).
- ldap syntaxes (See “Attribute syntax” on page 60).

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

A sample schema entry might contain:

© Copyright IBM Corp. 2002, 2010

```
 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. Refer the *IBM Tivoli Directory Server Version 6.3 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

IBM Tivoli Directory Server 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 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 at least 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 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.

If the attribute ibm-slapdSchemaCheck is set to V3 in the configuration file, then every entry can have only one structural object class, and if multiple structural object classes are added they must have parent-child relationship. For example, an entry type person, X, can be defined using the following structural object classes:

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

In this case, for the entry, X, the MUST attributes of the child structural object class, organizationalPerson, must be defined.

If the attribute ibm-slapdSchemaCheck is set to V3_lenient in the configuration file, then an entry can have one or more structural object classes, and if multiple structural object classes are added it is not a must that they should have parent-child relationship. For example, an entry, Y, can also be defined using the following structural object classes:

```
objectClass: person
objectClass: account
```

In this case, for the entry, Y, the MUST attributes of the structural object classes, person and account, must be defined.

Note: In Tivoli Directory Server, by default, the attribute ibm-slapdSchemaCheck is set to V3_lenient.

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 28 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.

Note: When the Web admin tool is used to access the admin server:

- The status bar on the Manage object classes panel displays a message indicating that the tool is connected to the admin server. If you access panels that are not supported by admin server, a message is displayed indicating that the functions on the panels are not supported.
- The Manage object classes panel is enabled based on the capabilities present in rootDSE for ibm-supportedcapabilities attribute.

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 39. 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**.
   - 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 40 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 65 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** .

   Note: You can also open the Edit object class panel to edit attributes of an object class by clicking on the object class name in the Objectclass column.

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 40 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 39. 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 40 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 65 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 it's 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. 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, o=sample
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
      [ "ENCRYPT"  whsp scheme whsp ]   ; encryption scheme
      [ "SECURE-CONNECTION-ONLY"  whsp ] ; secure connection required
      [ "RETURN-VALUE whsp returnValue whsp ]; value to be returned
      [ "NONMATCHABLE whsp ] ;  ; attribute can only be used in existence filters
     whsp ")"

     scheme =
      "SSHA" /
      "AES-128" /
      "AES-192" /
      "AES-256" /
        "SHA-224" /
      "SHA-256" /
      "SHA-384" /
      "SHA-512" /
      "SSHA-224" /
      "SSHA-256" /
      "SSHA-384" /
      "SSHA-512"

    returnValue =
      "encrypted" /
      "type-only"

   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 425 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 Tivoli Directory Server 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 55.

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.

Equality 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 with their respective OIDs and syntaxes

Equality matching rules		
Matching Rule	**OID**	**Syntax**
bitStringMatch	2.5.13.16	Bit String
booleanMatch	2.5.13.13	Boolean
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
caseIgnoreIA5SubstringsMatch	1.3.6.1.4.1.1466.109.114.3	IA5 String syntax
caseIgnoreListMatch	2.5.13.11	Directory String
caseIgnoreMatch	2.5.13.2	Directory String syntax
distinguishedNameMatch	2.5.13.1	DN - distinguished name
generalizedTimeMatch	2.5.13.27	Generalized Time syntax

Table 1. Equality matching rules with their respective OIDs and syntaxes (continued)

Equality matching rules		
Matching Rule	**OID**	**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
numericStringMatch	2.5.13.8	Numeric String
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 (.)
octetStringMatch	2.5.13.17	Directory String syntax
presentationAddressMatch	2.5.13.22	Presentation Address
protocolInformationMatch	2.5.13.24	Protocol Information
telephoneNumberMatch	2.5.13.20	Telephone Number syntax
uniqueMemberMatch	2.5.13.23	Name And Optional UID
uTCTimeMatch	2.5.13.25	UTC Time syntax

Table 2. Ordering matching rules with their respective OIDs and syntaxes

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
integerOrderingMatch	2.5.13.15	Integer
numericStringOrderingMatch	2.5.13.9	Numeric String
octetStringOrderingMatch	2.5.13.18	Octet String

Table 3. Substring matching rules with their respective OIDs and syntaxes

Substring matching rules		
Matching rule	**OID**	**Syntax**
caseExactSubstringsMatch	2.5.13.7	Directory String syntax
caseIgnoreListSubstringsMatch	2.5.13.12	Substring Assertion

Table 3. Substring matching rules with their respective OIDs and syntaxes (continued)

Substring matching rules		
Matching rule	**OID**	**Syntax**
caseIgnoreSubstringsMatch	2.5.13.4	Directory String syntax
numericStringSubstringsMatch	2.5.13.10	Substring Assertion
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 79.

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 28 for information on how to use these options.

Note: When the Web admin tool is used to access the admin server:

- The status bar on the Manage attributes panel displays a message indicating that the tool is connected to the admin server. If you access panels that are not supported by admin server, a message is displayed indicating that the functions on the panels are not supported.
- The Manage attributes panel is enabled based on the capabilities present in rootDSE for ibm-supportedcapabilities attribute.

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 39. 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 60 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 635 for additional information about multiple values.
9. Select a matching rule from each of the drop-down menus for **equality**, **ordering**, and **substring** matching rules. See the "Equality matching rules" on page 48 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 431 for information about security classes.

- Set the **Indexing rules** by selecting one or more indexing rules. See "Indexing rules" on page 50 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.
- Select an encryption scheme from the **Select encryption scheme** box.
- Select a search return type for the attribute value from the **Value to return on search** box.
- Select the **Require secure connection to view or change values** check box to specify secure connection when accessing encrypted attributes.
- Select the **Allow attribute in search filters** check box to specify whether the attributes are allowed in search filter.

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 60) and Case Ignore Equality matching (see "Equality matching rules" on page 48). 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 <admindn> -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 the **idsldapmodify** and **idsldapadd** command information in the *IBM Tivoli Directory Server Version 6.3 Command Reference* for more information.

Editing an attribute

Not all schema changes are allowed. See "Disallowed schema changes" on page 65 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** .

   Note: You can also open the Edit attribute panel to edit an attribute by clicking on the attribute name in the Name column.
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 55.
         - 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 anIBM 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 <admindn> -w <adminpw> -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 the **idsldapadd** command information in the *IBM Tivoli Directory Server Version 6.3 Command Reference* for more information about this utility.

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 53.
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 39. 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 60 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 635 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 "Equality matching rules" on page 48 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 50 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 65 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 the **idsldapadd** command information in the *IBM Tivoli Directory Server Version 6.3 Command Reference* for more information.

Encrypted Attributes

Local Administrative group members who are assigned DirDataAdmin and SchemaAdmin roles can specify attributes that are to be encrypted in the directory database using a subset of the encryption schemes supported for password information. The attributes can be encrypted using either 2-way or 1-way encryption schemes. The supported encryption schemes include AES-256, AES-192, AES-128, SSHA, SHA-224, SHA-256, SHA-384, SHA-512, SSHA-224, SSHA-256,

SSHA-384 and SSHA-512 and the supported attribute syntaxes include directory string, IA5 string, distinguished name, and telephone number.

The encrypted attribute policy will allow local admin group members who are assigned DirDataAdmin and SchemaAdmin roles to specify access to encrypted attributes that will be limited to clients that use secure connections. Furthermore, the policy will allow group members to define specific attributes as being non-matchable. This means that such attributes can only be used in presence filters. Additionally, the policy also allows group members to specify if values to be returned on a search should be encrypted or if only attribute names should be returned.

Note: Search filter assertions for encrypted attributes can be exact match or presence. Substring matches, ordering, and approximate matching cannot be used.

After specifying the attributes that are to be encrypted, the existing server data will be encrypted only after the next server startup. The time taken for this operation will depend on the number of entries that are to be encrypted. The encrypted attribute policy can be managed using the web administration tool.

Using Web Administration

If you have not done so already, expand **Schema management** in the navigation area and click **Manage encrypted attributes**.

The Manage encrypted attributes tab provides a way to manage encrypted attributes. Users can use this tab to manage and add existing encryptable attributes to encrypted attributes.

The Manage encrypted attributes tab will be available only if the server supports ibm-supportedcapability OID for encrypted attribute and returns the OID on rootDSE search.

To manage encryptable attributes:

1. To encrypt attributes, select the required encryptable attributes from the **Select attribute** list in the Attributes available for encryption section.
2. Select an encryption scheme from the **Select encryption scheme** box.
3. Select a search return type for the attribute value from the **Value to return on search** box.
4. Select the **Require secure connection to view or change values** check box to enable secure connection when accessing encrypted attributes.
5. Select the **Allow attributes in search filters** check box to specify whether the selected encryptable attributes are allowed in search filter.
6. Click the **Add to encrypted** button to populate the Encrypted attributes table with the selected encryptable attributes from the Select attribute box.
7. When you are finished, do one of the following:
   - Click **OK** to apply your changes and exit this panel.
   - Click **Cancel** to exit this panel without making any changes.

To manage encrypted attributes:

1. To remove an attribute from the Encrypted attributes table, click the **Select** column of the required encrypted attribute, and then click the **Remove** button or select **Remove** from the Select Action box and click **Go**.

2. To edit the encryption settings for an attribute, click the **Select** column of the required encrypted attribute, and then click the **Edit encryption settings** button or select **Edit encryption settings** from the Select Action box and click **Go**.
3. To remove all the attributes from the Encrypted attributes table, click the **Remove all** button or select **Remove all** from the Select Action box and click **Go**.
4. When you are finished, do one of the following:
   - Click **OK** to apply your changes and exit this panel.
   - Click **Cancel** to exit this panel without making any changes.

Edit encryption settings

This Edit encryption settings panel contains settings that are used for specifying and modifying the existing values of the encrypted attributes such as encryption type, search return type, type of connection for accessing attributes, and search filter.

To edit encrypted attributes:

1. Select an encryption scheme from the **Select encryption scheme** box.
2. Select a search return type for the attribute value from the **Value to return on search** box.
3. Select the **Required secure connection to view or change values** check box to enable secure connection when accessing the encrypted attribute.
4. Select the **Allow attributes in search filters** check box to specify whether the selected encrypted attribute is allowed in search filter.
5. When you are finished, do one of the following:
   - Click **OK** to save the changes made to the encrypted attribute values in the directory schema.
   - Click **Cancel** to exit this panel without making any changes.

Encrypted attributes in a replication environment

During replication it is ensured that attributes are replicated over secure connections. The replication process also determines if any incompatible features are used between the supplier and the consumer. For instance, if the supplier has encrypted attributes while the consumer does not support encryption, then the replication process will not start. Also, if the network includes servers running with earlier releases, such as Tivoli Directory Server version 6.0, replicated schema changes will fail.

It is recommended that servers share a crypto key, and that the administrator ensure that attributes are encrypted on all servers. If the crypto keys differ between supplier and consumer, changes will be decoded and replicated as clear text.

Using command line

To encrypt an attribute, say for instance the uid attribute using the AES encryption scheme, issue the following command:

```
ldapmodify –D <adminDN> –w <adminPW>
dn: cn=schema
changetype: modify
replace: attributetypes
attributetypes:( 0.9.2342.19200300.100.1.1 NAME 'uid' DESC 'Typically a user
 shortname or userid.'
EQUALITY 1.3.6.1.4.1.1466.109.114.2 ORDERING 2.5.13.3 SUBSTR 2.5.13.4
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 USAGE userApplications )
-
```

```
replace: IBMAttributetypes
IBMAttributetypes:( 0.9.2342.19200300.100.1.1 DBNAME( 'uid'  'uid' )
ACCESS-CLASS normal LENGTH 256 EQUALITY ORDERING SUBSTR APPROX
ENCRYPT AES256 SECURE-CONNECTION-REQUIRED  RETURN-VALUE  encrypted))
```

Attribute syntax

Attribute syntax identifies the required format of the data.

Table 4. Attribute syntax

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
Bit String	1.3.6.1.4.1.1466.115.121.1.6
Boolean - TRUE/FALSE	1.3.6.1.4.1.1466.115.121.1.7
Certificate	1.3.6.1.4.1.1466.115.121.1.8
Certificate List	1.3.6.1.4.1.1466.115.121.1.9
Certificate Pair	1.3.6.1.4.1.1466.115.121.1.10
Country String	1.3.6.1.4.1.1466.115.121.1.11
Delivery Method	1.3.6.1.4.1.1466.115.121.1.14
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
Enhanced Guide	1.3.6.1.4.1.1466.115.121.1.21
Facsimile Telephone Number	1.3.6.1.4.1.1466.115.121.1.22
Fax	1.3.6.1.4.1.1466.115.121.1.23
Generalized Time syntax	1.3.6.1.4.1.1466.115.121.1.24
Guide	1.3.6.1.4.1.1466.115.121.1.25
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
JPEG	1.3.6.1.4.1.1466.115.121.1.28
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
MHS OR Address	1.3.6.1.4.1.1466.115.121.1.33
Name And Optional UID	1.3.6.1.4.1.1466.115.121.1.34
Name Form Description	1.3.6.1.4.1.1466.115.121.1.35
Numeric String	1.3.6.1.4.1.1466.115.121.1.36
Object Class Description syntax	1.3.6.1.4.1.1466.115.121.1.37
Octet String	1.3.6.1.4.1.1466.115.121.1.40
Other Mailbox	1.3.6.1.4.1.1466.115.121.1.39
Postal Address	1.3.6.1.4.1.1466.115.121.1.41
Presentation Address	1.3.6.1.4.1.1466.115.121.1.43

Table 4. Attribute syntax (continued)

Syntax	OID
Protocol Information	1.3.6.1.4.1.1466.115.121.1.42
Printable String	1.3.6.1.4.1.1466.115.121.1.44
String for containing OIDs. The OID is a string containing digits (0-9) and decimal points (.). See "Object identifier (OID)" on page 39.	1.3.6.1.4.1.1466.115.121.1.38
Substring Assertion	1.3.6.1.4.1.1466.115.121.1.58
Supported Algorithm	1.3.6.1.4.1.1466.115.121.1.49
Telephone Number syntax	1.3.6.1.4.1.1466.115.121.1.50
Telex Number	1.3.6.1.4.1.1466.115.121.1.52
Teletex Terminal Identifier	1.3.6.1.4.1.1466.115.121.1.51
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 79.	1.3.6.1.4.1.1466.115.121.1.53

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=uniqueattributes,cn=localhost and cn=uniqueattributes,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=IBMploicies, 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.

   Note: Starting with the IBM Tivoli Directory Server 6.3 release, attribute cache is deprecated. Henceforth, users should avoid using 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-UniqueAttributes
```

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=IBMploicies, 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=uniqueattributes,cn=localhost and cn=uniqueattributes,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=uniqueattributes,cn=localhost or cn=uniqueattributes,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 this unique attributes entry from the directory, the unique constraints enforced on the unique attributes are dropped to allow nonunique values for the attributes again.

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 Tivoli Directory Server Version 6.3 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 46 for examples using the Web Administration Tool and the **idsldapmodify** command.

See the *IBM Tivoli Directory Server Version 6.3 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, for IBM Tivoli Directory Server 6.0 and above versions 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 234.

See Chapter 14, “Replication,” on page 219 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 IBM Tivoli Directory Server:

- aclEntry
- aclPropagate
- aclSource
- aliasedObjectName, aliasedentryName
- createTimestamp
- creatorsName
- entryOwner
- hasSubordinates
- ibm-allGroups
- ibm-allMembers
- ibm-capabilitiessubentry
- 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 I, “Required attribute definitions for IBM Tivoli Directory Server,” on page 539 for more information about these attributes.

A special attribute description, "+", can be used in the attribute list of a search request to return all operational attributes. If a "+" is present in the search request, the server returns all operational attributes to which the client is authorized. For further information, see the idsldapsearch command information in the *IBM Tivoli Directory Server Version 6.3 Command Reference.*

Given below is a table that lists the supported special attributes, and the associated list of operational attributes:

Table 5. Supported special attributes and associated list of operational attributes

Attribute	Attributes returned by "+" attribute	Attributes added by ++
+	Returns all attributes listed in this column.	++ returns all attributes listed in this column
+ibmaci	aclentry aclsource aclpropagate entryowner ownersource ownerpropagate ibm-filterAclEntry ibm-filterAclInherit ibm-effectiveAcl	
+ibmentry	creatorsname createtimestamp modifiersname modifytimestamp subschemasubentry ibm-entryuuid ibm-capabilitiessubentry ibm-enabledcapabilities (1) ibm-supportedcapabilities (1) ibm-replicationThisServerIsMaster ibm-replicationIsQuiesced	++ibmentry includes the attributes from +ibmentry and adds: ibm-allgroups ibm-allmembers ibm-entryChecksum ibm-entryChecksumOp numsubordinates hassubordinates

Table 5. Supported special attributes and associated list of operational attributes (continued)

Attribute	Attributes returned by "+" attribute	Attributes added by ++
+ibmpwdpolicy	pwdAccountLockedTime pwdChangedTime pwdExpirationWarned pwdFailureTime pwdGraceUseTime pwdHistory pwdReset ibm-pwdAccountLocked ibm-pwdGroupPolicyDN ibm-pwdIndividualPolicyDN	
+ibmrepl	ibm-replicationChangeLDIF ibm-replicationLastActivationTime ibm-replicationLastChangeId ibm-replicationLastFinishTime ibm-replicationLastResult ibm-replicationLastResultAdditional ibm-replicationNextTime ibm-replicationPendingChangeCount ibm-replicationState ibm-replicationFailedChangeCount ibm-replicationperformance	++ibmrepl includes the attributes from +ibmrepl and adds: ibm-replicationPendingChanges ibm-replicationFailedChanges

Restricted attributes

The following lists of restricted attributes are supported by 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 I, "Required attribute definitions for IBM Tivoli Directory Server," on page 539 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 I, "Required attribute definitions for IBM Tivoli Directory Server," on page 539 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-slapdAclCache
- ibm-slapdAclCacheSize
- ibm-slapdAdminDN
- ibm-slapdAdminGroupEnabled
- ibm-slapdAdminPW
- ibm-slapdAllowAnon
- ibm-slapdAllReapingThreshold
- ibm-slapdAnonReapingThreshold
- ibm-slapdAuthIntegration
- ibm-slapdBoundReapingThreshold
- ibm-slapdBulkloadErrors

- ibm-slapdCachedAttribute
- ibm-slapdCachedAttributeSize
- ibm-slapdChangeLogMaxAge
- ibm-slapdChangeLogMaxEntries
- ibm-slapdCLIErrors
- ibm-slapdConfigPwdPolicyOn
- 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-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-slapdSslPKCS11Lib
- ibm-slapdSslPKCS11Keystorage
- ibm-slapdSslPKCS11Enabled
- ibm-slapdSslPKCS11AcceleratorMode
- ibm-slapdSslPKCS11TokenLabel
- ibm-replicaPKCS11Enabled
- 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
- paswordMaxRepeatedChars
- 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 I, "Required attribute definitions for IBM Tivoli Directory Server," on page 539 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 I, “Required attribute definitions for IBM Tivoli Directory Server,” on page 539 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 6.3 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.

iPlanet compatibility

The parser used by 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 specifications of IBM Tivoli Directory Server Version 6.0 and above. 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. 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 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 IBM Tivoli Directory Server describes it:

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

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 a global administrative group, or a member of the local administrative group based on their roles.

- "Changing the primary administrator distinguished name and password"
- "Starting and stopping the server" on page 82
- "Checking server status" on page 83
- "Managing server connections" on page 104
- "Managing connection properties" on page 106
- "Managing unique attributes" on page 61

Changing the primary 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 181 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**

© Copyright IBM Corp. 2002, 2010

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 6.3 Installation and Configuration Guide* for additional information on using the idsxcfg utility.

See Chapter 3, "Distinguished names (DNs)," on page 13 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 server (**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.

   Note: When the Web admin tool is used to access the admin server:
   - The status bar on the Start/Stop/Restart Server panel displays a message indicating that the tool is connected to the admin server. If you access panels that are not supported by admin server, a message is displayed indicating that the functions on the panels are not supported.
   - The Start/Stop/Restart Server panel is enabled based on the capabilities present in rootDSE for ibm-supportedcapabilities attribute.
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 6. 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 23 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 server (**idsdiradm**) must be running for the ibmdirctl

```
ibmdirctl -h mymachine -D myDN -w mypassword -p <adminportnumber> start
```

or

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

Use the following commands to stop the server:

```
ibmdirctl -h mymachine -D myDN -w mypassword -p <adminportnumber> stop
```

or

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

to start and stop the server respectively. See the **ibmdirctl** and **idsdiradm** command information in the *IBM Tivoli Directory Server Version 6.3 Command Reference* for more 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 V6.3 - <***instancename***>** and click **Start**.
6. To stop the server select **Control Panel ->Administrative Tools->Services**, select **IBM Tivoli Directory Server Instance V6.3 - <***instancename***>** and click **Stop**.

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

To start or stop a directory server instance using the Instance Administration Tool:

- In the Instance Administration Tool, select the directory server instance you want to start or stop, and then click **Start/Stop**.

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.

Note: When the Web admin tool is used to access the admin server:

- The title of the View server status panel will change to View Admin Server status.
- The status bar on the View Admin Server Status panel displays a message indicating that the tool is connected to the admin server. If you access panels that are not supported by admin server, a message is displayed indicating that the functions on the panels are not supported.
- The View Admin Server Status panel is enabled based on the capabilities present in rootDSE for ibm-supportedcapabilities attribute.

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** , **Running configuration only mode**, or **Stopped**. 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.

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.

System Information

Click **System information** to display the following information:

Operating System name
Specifies the name of the operating system running on the LDAP server.

Disk space used by directory where the DB2 database is stored (Kbytes)
Specifies the amount of disk space in kilobytes used by the directory that contains DB2 database.

Disk space available to DB2 database (Kbytes)
Specifies the amount of disk space in kilobytes available for DB2 database.

Operation counts

Click **Operation counts 1** 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 delete requests since the server was started.

Number of delete operations completed
The number of completed delete 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.

Note: When accessing admin server using the Web admin tool, some fields will not be displayed.

Click **Operation counts 2** to display the following information:

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.

Number of operations not in a transaction failed due to a deadlock condition
The number of operations that are not in a transaction failed because of a deadlock condition.

Number of operations waiting in the deadlock detector
The number of operations waiting in the deadlock detector.

Maximum number of operations waiting in the deadlock detector
The maximum number of operations waiting in the deadlock detector at any point of time.

Number of operations not in a transaction that have been retried
The number of operations that are not in a transaction and have been retried to avoid deadlocks.

Note: When accessing admin server using the Web admin tool, some fields will not be displayed.

Transaction counts

Click **Transaction counts** to display the following:

Number of transactions requested
The number of transaction requests initiated since the server was started.

Number of transactions completed
The number of transactions completed that can be either commit or rollback requests.

Number of transaction commits requested
The number of transaction commits requested since the server was started.

Number of transactions committed
The number of transactions committed successfully since the server was started.

Number of end transaction rollbacks requested
The number of end transaction rollback requests received since the server was started.

Number of transactions rolled back
The number of transactions rolled back either by requests or because of operation failure.

Number of transaction prepare operations requested
The number of transaction prepare operations requested since the server was started.

Number of transaction prepare operations completed
The number of transaction prepare operations completed since the server was started.

Number of transactions that have requested a prepare, but have not yet been committed or rolledback
The number of transactions that have requested a prepare but have not yet been committed or rolled back.

Note: You can click **Refresh** to refresh the information on this panel. Click **Close** to return to the "Introduction" panel.

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.

Note: When accessing admin server using the Web admin tool, some fields will not be displayed.

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 receive, 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
derefaliases=neverDerefAliases
typesonly=false
filter=(objectclass=*)
attributes=all
```

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

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 the **ldaptrace** command information

in the *IBM Tivoli Directory Server Version 6.3 Command Reference* 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 the section on **Debugging levels** in the *IBM Tivoli Directory Server Version 6.3 Command Reference.*

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

Number of messages added to DB2 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.

Persistent search

Click **Persistent search** to display the following information:

Number of changes sent
Indicates the number of changes sent after the server startup.

Number of active connections
Indicates the number of active persistent search connections.

Number of dropped connections
Indicates the number of connections that have been dropped as a result of network or client failure.

Number of pending changes
Indicates the number of new updates in the queue that are yet to be processed by the persistent search thread.

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=system,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 6.3

directoryversion
The specific version number indicating fixpack level.

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.

addsfromsuppliers
The number of update operations received from replication supplier.

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

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

deletesfromsuppliers
The number of delete operations received from replication supplier.

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.

modrdnsfromsuppliers
The number of modify RDN operations received from replication supplier.

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

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

modifiesfromsuppliers
The number of modify operations received from replication supplier.

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.

transactionsrequested
The number of transaction requests initiated.

transactionscompleted
The number of transaction operations completed.

transactionpreparesrequested
The number of prepare transaction operations requested.

transactionpreparescompleted
The number of prepare transaction operations completed.

transactioncommitsrequested
The number of commit transaction operations requested.

transactionscommitted
The number of transaction operations committed.

transactionrollbacksrequested
The number of transaction operations requested for rollback.

transactionsrolledback
The number of transaction operations rolled back.

transactionspreparedwaitingoncommit
The number of transaction operations which are prepared and waiting for commit/rollback.

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.

group_members_cache_size
The maximum number of groups whose members needs to be cached.

group_members_cache_current
The number of groups whose members are currently cached.

group_members_cache_hit
The number of groups whose members were requested and retrieved from the group members' cache.

group_members_cache_miss
The number of groups whose members were requested and found in the group members' cache that needed to have the members retrieved from DB2.

group_members_cache_bypass
The maximum number of members allowed in a group that will be cached in the group members' 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.

operations_waiting
The number of operations waiting in the deadlock detector.

maximum_operations_waiting
The maximum number of operations that have waited at one time in the deadlock detector.

operations_retried
The number of operations retired due to deadlocks.

operations_deadlocked
The number of operations in deadlock.

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_auto_adjust
Indicates if attribute cache auto adjusting is configured to be on or off.

cached_attribute_auto_adjust_time
Indicates the configured time on which to start attribute cache auto adjusting.

cached_attribute_auto_adjust_time_interval
Indicates the time interval after which to repeat attribute cache auto adjusting for the day.

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 the **ldaptrace** command information in the *IBM Tivoli Directory Server Version 6.3 Command Reference* 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 the section on **Debugging levels** in the *IBM Tivoli Directory Server Version 6.3 Command Reference.*

trace_message_log
The current LDAP_DEBUG_FILE environment variable setting for the server.

auditinfo
Contains the current audit configuration. This attribute is displayed only if the monitor search is initiated by an administrator.

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.

currentpersistentsearches
Indicates number of active persistent search connections.

persistentsearchpendingchanges
Indicates the number of new updates in the queue that are yet to be processed by the persistent search thread.

persistentsearchprocessedchanges
Indicates number of changes processed by persistent search process.

lostpersistentsearchconns
Indicates the number of lost persistent search connections.

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.

Note: Starting with the IBM Tivoli Directory Server 6.3 release, attribute cache is deprecated. Henceforth, users should avoid using attribute cache.

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 V3 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
derefaliases=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 DN that the connection is bound as.

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.

Note: Starting with the IBM Tivoli Directory Server 6.3 release, attribute cache is deprecated. Henceforth, users should avoid using attribute cache.

cn=system,cn=monitor

To collect system information from machines on which the directory server is running, issue the following command:

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

The information that is returned will depend on the operating system on which directory server is running. Following information is returned for machines running on windows operating system:

memoryUsed
The amount of virtual memory used (KB)

memoryFree
The amount of idle memory (KB).

operatingSystem
Operating system name. For instance, Windows or Windows-X640.

diskSpaceUsedByDB
Disk space used by the directory where DB2 database is stored (KB).

diskSpaceAvailableToDB
Disk space available to DB2 database (KB).

The following information is returned for machines running on non-windows operating systems:

operatingSystem
Operating system name. For instance, Linux-x32, Linux-x64, Linux-PPC, Linux-Z, Solaris, Solaris-x86, or AIX.

diskSpaceUsedByDB
Disk space used by the directory where DB2 database is stored (KB).

diskSpaceAvailableToDB
Disk space available to DB2 database (KB).

View cache status

Expand the **Server administration** category in the navigation area. Click **View cache status**. This panel has six 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.

Entry cache

Click **Entry cache** to display the following information:

Number of elements in entry cache
The value in the **Number of elements in entry cache** field indicates the number of elements present in the entry cache currently. The attribute "entry_cache_current" of the "cn=monitor" entry is associated with this field.

Maximum number of elements in entry cache
The value in the **Maximum number of elements in entry cache** field indicates the maximum number of elements specified for entry cache. The attribute "entry_cache_size" of the "cn=monitor" entry is associated with this field.

Entry cache hits
The value in the **Entry cache hits** field indicates the number of times elements were found in the entry cache during search or other LDAP operations. The attribute "entry_cache_hit" of the "cn=monitor" entry is associated with this field.

Entry cache misses
The value in the **Entry cache misses** field indicates the number of times elements were unavailable in the entry cache during search or other LDAP operations. The attribute "entry_cache_miss" of the "cn=monitor" entry is associated with this field.

Filter cache

Click **Filter cache** to display the following information:

Number of elements in filter cache
The value in the **Number of elements in filter cache** field indicates the number of elements present in the filter cache currently. The attribute "filter_cache_current" of the "cn=monitor" entry is associated with this field.

Maximum number of elements in filter cache
The value in the **Maximum number of elements in filter cache** field indicates the maximum number of elements specified for filter cache. The attribute "filter_cache_size" of the "cn=monitor" entry is associated with this field.

Filter cache hits
The value in the **Filter cache hits** field indicates the number of times

elements were found in the filter cache during search or other LDAP operations. The attribute "filter_cache_hit" of the "cn=monitor" entry is associated with this field.

Filter cache misses
The value in the **Filter cache misses** field indicates the number of times elements were unavailable in the filter cache during search or other LDAP operations. The attribute "filter_cache_miss" of the "cn=monitor" entry is associated with this field.

Maximum number of elements from a single search added to filter cache
The value in the **Maximum number of elements from a single search added to filter cache** field indicates the maximum number of elements from a search operation added to the filter cache. The attribute "filter_cache_bypass_limit" of the "cn=monitor" entry is associated with this field.

ACL cache

Click **ACL cache** to display the following information:

Cache ACL information
The value in the **Cache ACL information** field indicates whether the ACL caching is enabled or not. The attribute "acl_cache" of the "cn=monitor" entry is associated with this field.

Maximum number of elements in ACL cache
The value in the **Maximum number of elements in ACL cache** field indicates the maximum number of elements specified for ACL cache. The attribute "acl_cache_size" of the "cn=monitor" entry is associated with this field.

Group members' cache

Click **Group members' cache** to display the following information:

Maximum number of groups allowed in cache
The value in the **Maximum number of groups allowed in cache** field indicates the maximum number of groups that is to be cached. The attribute "group_members_cache_size" is associated with this field.

Maximum number of members in a group that can be cached
The value in the **Maximum number of members in a group that can be cached** field indicates the maximum number of members that can be cached for a group in the group members' cache. The attribute "group_members_cache_bypass_limit" is associated with this field.

Number of groups in cache
The value in the **Number of groups in cache** field indicates the number of groups whose members are currently cached in the group members' cache. The attribute "group_members_cache_current" is associated with this field.

Group cache hits
The value in the **Group cache hits** field indicates the number of requests for the members of groups that were successfully retrieved from the group members' cache. The attribute "group_members_cache_hit" is associated with this field.

Group cache misses
The value in the **Group cache misses** field indicates the number of requests for the members of groups that were unavailable in the group

members' cache and were successfully retrieved from DB2. The attribute "group_members_cache_miss" is associated with this field.

Directory cached attributes

Click **Directory cached attributes** to display the following information. The status items are displayed in a table format.

Note: The Directory cached attributes table will not be displayed if directory cached attributes do not exist. Instead a message is displayed indicating that there are no directory cached attributes.

Table 7. Directory cached attributes table

Attribute ∧	Number of cache hits ∧	Cache size ∧

Attribute
Indicates the name of the attribute.

Number of cache hits
Indicates the number of times the attribute filter has been used after it was cached.

Cache size
Indicates the amount of memory used by this attribute cache.

This tab also contains two non-editable fields:

Cached attribute total size (in kilobytes)
Indicates 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 (in kilobytes)
Indicates the maximum amount of memory that can be used by attribute caching.

Note: Starting with the IBM Tivoli Directory Server 6.3 release, attribute cache is deprecated. Henceforth, users should avoid using attribute cache.

Directory cache candidates

Click **Directory cache candidates** to display the following information. The information about directory cached candidates are displayed in a table

Note: The Directory cached candidates table will not be displayed if directory cached candidates do not exist. Instead a message is displayed indicating that there are no directory cached candidates.

Table 8. Directory cache candidates table

Attribute ∧	Number of hits ∧

Attribute
Indicates the name of the attribute.

Number of hits
Indicates the number of times the attribute filter has been used.

Viewing server capabilities (Root DSE) information

A root DSE entry contains information about an LDAP server instance, which can be queried by a root DSE search. On performing a root DSE search on a server instance, root DSE attributes and their values, OIDs of supported and enabled capabilities, OIDs of supported extensions and controls are displayed. To view root DSE information, use any one of the following methods.

Using Web Administration

If you have not done so already, click **Server administration** in the Web Administration navigation area and then click **View server capabilities (Root DSE)** in the expanded list. Next, click **General**.

The **General** tab displays the following information.

Server instance name
This field displays the name of the directory server instance running on the server. This field is populated with the value of the ibm-slapdServerInstanceName attribute in the root DSE entry.

Server Id
This field displays the unique ID assigned to the server at the first startup of the server. This ID is used in replication topology to determine a server's role. This field is populated with the value of the ibm-serverId attribute in the root DSE entry.

Port number
This field displays the non secure port on which the server is listening. This is present only if the server does not have a secure port enabled. This field is populated with the value of the port attribute in the root DSE entry.

Directory version
This field displays the version of IBM Tivoli Directory Server installed on the server. This field is populated with the value of the ibmdirectoryversion attribute in the root DSE entry.

Server backend
This field specifies whether this server loads a database or proxy backend. This field is populated with the value of the ibm-slapdServerBackend attribute in the root DSE entry.

Supported audit version
This field displays the supported version of auditing. This field is populated with the value of the ibm-supportedAuditVersion attribute in the root DSE entry.

LDAP service name
This field displays the host name of the server. If a Kerberos realm is defined, the value is displayed in the form hostname@realmname. This field is populated with the value of the ibm-ldapservicename attribute in the root DSE entry.

Security
This field displays the secure SSL port the server is listening on. This field is populated with the value of the security attribute in the root DSE entry.

Size limit
This field displays the limit on the number of entries returned by a search initiated by non administrative users. This field is populated with the value of the ibm-slapdSizeLimit attribute in the root DSE entry.

Time limit (seconds)
This field displays the maximum amount of time in seconds the server spends processing a search request initiated by non administrative users. This field is populated with the value of the ibm-slapdTimeLimit attribute in the root DSE entry.

Dereferences alias
This field displays how the server is configured to handle dereferencing. This field is populated with the value of the ibm-slapdDerefAliases attribute in the root DSE entry.

Vendor name
This field displays the supplier of this version of LDAP running on the server. This field is populated with the value of the vendorname attribute in the root DSE entry. For example, for IBM Tivoli Directory Server, this is set to International Business Machines (IBM).

Vendor version
This field displays the version of the directory server. This field is populated with the value of the vendorversion attribute in the root DSE entry. For example, for IBM Tivoli Directory Server (TDS) 6.3, the vendor version is set to 6.3.

Sub schema sub entry
This field displays the name of a subschema entry in which the server makes available attributes specifying the schema. This field is populated with the value of the subschemasubentry attribute in the root DSE entry. It value is set to cn=schema.

SASL digest realm name
This field displays the SASL digest realm name associated with the server. This field is populated with the value of the ibm-sasldigestrealmname attribute in the root DSE entry.

Supported LDAP version
This list displays the LDAP versions implemented by the current server. This list is populated with the values of the supportedldapversion attribute in the root DSE entry. The values of this attribute are the versions of the LDAP protocol that the server implements.

Naming context
This list displays the naming contexts available in the server. This list is populated with the values of the namingcontexts attribute in the root DSE entry. 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 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.

Configuration naming context
This field displays the suffix where the server's configuration entries are stored. This field is populated with the value of the ibm-configurationnamingcontext attribute in the root DSE entry.

You can click **Refresh** to refresh the information on this panel. Click **Close** to return to the "Introduction" panel. To view information about the supported capabilities, click **Supported Capabilities**. The **Supported Capabilities** tab displays the following information:

Supported Capabilities
This list displays the server capabilities currently supported by the server. This list is populated with the values of the ibm-supportedcapabilities attribute in the root DSE entry.

You can click **Refresh** to refresh the information on this panel. Click **Close** to return to the "Introduction" panel. To view information about the enabled capabilities, click **Enabled Capabilities**. The **Enabled Capabilities** tab displays the following information:

Enabled Capabilities
This list displays the server capabilities currently enabled for use on the server. This list is populated with the values of the ibm-enabledcapabilities attribute in the root DSE entry.

You can click **Refresh** to refresh the information on this panel. Click **Close** to return to the "Introduction" panel. To view information about the supported extensions, click **Supported Extensions**. The **Supported Extensions** tab displays the following information:

Supported Extensions
This list displays the OBJECT IDENTIFIERS (OIDs) of the supported extended operations which the server supports. This list is populated with the values of the supportedExtension attribute in the root DSE entry.

You can click **Refresh** to refresh the information on this panel. Click **Close** to return to the "Introduction" panel. To view information about the supported controls, click **Supported Controls**. The **Supported Controls** tab displays the following information:

Supported Controls
This list displays the OBJECT IDENTIFIERS (OIDs) of the supported controls which the server supports. This list is populated with the values of the supportedControl attribute in the root DSE entry.

You can click Refresh to refresh the information on this panel. Click Close to return to the "Introduction" panel. To view information about the supported SASL mechanism, click Supported SASL Mechanism. The Supported SASL Mechanism tab displays the following information:

Supported SASL Mechanism
This list displays all the names of the supported SASL mechanisms supported by the server. This list is populated with the values of the supportedsaslmechanisms attribute in root DSE entry. This attribute contains any SASL mechanism that is registered to the server.

You can click **Refresh** to refresh the information on this panel. Click **Close** to return to the "Introduction" panel.

Using command line

On performing a root DSE search on a server instance, root DSE attributes and their values, OIDs of supported and enabled capabilities, OIDs of supported extensions and controls are displayed. To initiate a root DSE search issue the following command:

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

For more information about root DSE attributes, see "Attributes in the root DSE" on page 491

To list the server capabilities currently enabled for use on the server, issue the following command:

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

To list the server capabilities currently enabled for use on the server, issue the following command:

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

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 connection 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.

Note:

- 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 Local administration group having DirDataAdmin or ServerConfigGroupMember role, 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 9. 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 the **ldapexop** command information in the *IBM Tivoli Directory Server Version 6.3 Command Reference* 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 10. Connections that exceed this limit are closed when the cleanup process is initiated.

   Note: For a Windows system, a connection that exceeds 30 seconds is automatically dropped. Therefore the **Result timeout limit** setting is overridden by the operating system after 30 seconds.
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.

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: 10
```

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 Appendix L, "Dynamically-changed attributes," on page 587 for a list of the attributes that can be updated dynamically.

Chapter 10. Setting server properties

You can set the following properties for your server:

- "Changing server ports and enabling language tags" on page 110
- "Search Settings" on page 115
- "Enabling and disabling transaction support" on page 125
- "Enabling and disabling event notification" on page 123
- "Adding and removing suffixes" on page 127
- Chapter 13, "Referrals," on page 211
- "Adding attributes to and removing attributes from the attribute cache" on page 133
- "Enforcing minimum ulimits" on page 112

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 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 Appendix L, "Dynamically-changed attributes," on page 587 for a list of the attributes that can be updated dynamically. See the **idsldapmodify** and **idsldapexop** command information in the *IBM Tivoli Directory Server Version 6.3 Command Reference* for more information.

Note: Only the administrator and members of the administrative group are allowed to update the server configuration settings.

© Copyright IBM Corp. 2002, 2010

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 34.

Using Web Administration

Click the **Server administration** category in the Web administration navigation area and then click **Manage server properties** tab 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 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 34.
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 34.
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 405 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 82.

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 server locally to resynchronize the ports. See Chapter 4, "Directory administration server," on page 19. 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 82.

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 server locally to resynchronize the ports. See Chapter 4, “Directory administration server,” on page 19.

Setting Performance

Note: For the latest tuning information, see the *IBM Tivoli Directory Server Version 6.3 Performance Tuning and Capacity Planning Guide* located on the Tivoli Software Library Web site.

You can change the search limits and connections settings to enhance performance.

Using Web Administration

If you have not done so already, click **Server administration** in the Web Administration navigation area and then click **Manage server properties** in the expanded list. Next, click the **Performance** tab.

The Performance tab allows you to enhance directory performance when you configure database connection settings. The LDAP server maintains a certain number of connections to the DB2(R) servers. This number can be set in the Maximum number of database connections field. By increasing the number of DB2

connections, LDAP can increase its level of concurrency and can improve throughput performance. To change the database connections settings to enhance performance do the following:

1. Specify the maximum number of database connections in the **Maximum number of database connections** field. This sets the number of DB2 connections used by the server. This field is not available if the server you are connected to is configured as a proxy server.
2. Specify the maximum number of database connections for replication in the **Maximum number of database connections for replication** field. This sets the number of DB2 connections used by the server for replication. This field is not available if the server you are connected to is configured as a proxy server.
3. Specify the maximum number of retries the back-end should attempt for operations not in a transaction to avoid deadlocks. If you select **Retries**, you must enter the number of retries allowed for operations not in a transaction. Otherwise, select **Unlimited**. You must specify numeric values only.
4. When you are finished, do one of the following:
   - Click **OK** to apply your changes and exit this panel.
   - Click **Apply** to apply your changes and stay on this panel.
   - 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 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
```

Enforcing minimum ulimits

The directory server tries to enforce minimum ulimit option values that are considered important for the smooth running of the server. During startup, the directory server verifies whether the ulimit option values for the current process are greater than or equal to the prescribed ulimit option values specified in the configuration file. If the verification fails, then the server attempts to set the ulimit option values of the current process to the prescribed values. If the server fails to do so, it will start in configuration only mode.

Given below is a list of all the typical ulimit options whose values are critical for the smooth running of the directory server.

Note: Ulimit options are applicable only to the proxy and back-end servers. No minimum ulimit options values are prescribed for the admin server process.

Critical memory parameters

Virtual memory size
This option includes all types of memory including stack, heap, and

memory-mapped files. Attempts to allocate memory in excess of this limit will fail with an out-of-memory error. The value for this option is specified in kilobytes.

Maximum resident set size (RSS)
This option limits the amount of memory that can be swapped in to physical memory on behalf of any one process. The value for this option is specified in kilobytes.

Note: AIX defines this ulimit option, while Solaris does not specify this option.

Data segment
This option limits the amount of memory that a process can allocate to a heap. The value for this option is specified in kilobytes.

Stack size
This option limits the amount of memory a process can allocate to a stack. The value for this option is specified in kilobytes.

Critical File parameters

File size
This option limits the maximum size of a file that a process can create. This is specified in 512-byte blocks.

Nofile This option limits the number of file descriptors belonging to a single process. File descriptors includes not only files but also sockets for Internet communication.

Note: On Solaris, the number of open files limit is set to the hard limit of the number of open files while starting the sever. The number of open files limit cannot be changed using the ulimit feature.

The table below lists the operating system default values and the prescribed minimum ulimit values of the critical options.

Table 10. System specific ulimit values

Ulimit Option	AIX		Solaris	
	Operating system default	**Prescribed minimum**	**Operating system default**	**Prescribed minimum**
Data segment size	256 MB	256 MB	Unlimited	256 MB
Virtual memory	Unlimited	1 GB	Unlimited	1 GB
Nofile	2000	500	256	256
Maximum resident set size (rss)	64 MB	256 MB	N/A	N/A
File size	1024 MB	1024 MB	Unlimited	1024 MB
Stack size	64 MB	64 MB	8 MB	8 MB

Table 11. System specific ulimit values

Ulimit Option	Linux	
	Operating system default	**Prescribed minimum**
Data segment size	Unlimited	256 MB

Table 11. System specific ulimit values (continued)

Ulimit Option	Linux	
	Operating system default	**Prescribed minimum**
Virtual memory	Unlimited	1 GB
Nofile	1024	500
Maximum resident set size (rss)	N/A	N/A
File size	Unlimited	1024 MB
Stack size	10 MB	10 MB

Note: Operating system default ulimit option values may vary for different kernel versions and for different shells in the same kernel version.

An administrator can modify the minimum ulimit option values using the web administration tool or through command line.

Using Web Administration

If you have not done so already, click **Server administration** in the Web Administration navigation area and then click **Manage server properties** in the expanded list. Next, click the **Ulimit settings** tab.

1. To specify the virtual memory size, select **Size** and specify a value in kilobytes in the text box. Alternatively, to specify the virtual memory size as unlimited, select **Unlimited**.
2. To specify the resident set size, select **Size** and specify a value in kilobytes in the text box. Alternatively, to specify the resident set size as unlimited, select **Unlimited**.
3. To specify the data segment size, select **Size** and specify a value in kilobytes in the text box. Alternatively, to specify the data segment size as unlimited, select **Unlimited**.
4. To specify the stack size, select **Size** and specify a value in kilobytes in the text box. Alternatively, to specify the stack size as unlimited, select **Unlimited**.
5. To specify the file size in blocks of 512 bytes, select **File size** and specify a value in the text box. Alternatively, to specify the file size as unlimited, select **Unlimited**.
6. Enter the number of file descriptors belonging to a single process in the **Number of open file descriptors** text box.
7. Click **OK** or **Apply** for the settings to take effect.

Using the command line

Ulimit option values can be modified using the ldapmodify command line utility. For example, to modify the ulimit value for virtual memory, issue the following command:

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

where <filename> contains:

dn: cn=ulimits, cn= configuration
changetype: modify
replace: ibm-slapdUlimitVirtualMemory
ibm-slapdUlimitVirtualMemory: <New prescribed ulimit for virtual memory>
```

Similarly, the ldapmodify command can be used to modify other ulimit option values such as data segment size, nofile, maximum resident set size (rss), file size, and stack size.

Search Settings

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. 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 deference 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.

To configure search settings, do the following:

Using Web Administration

If you have not done so already, click **Server administration** in the Web Administration navigation area and then click **Manage server properties** in the expanded list. Next, click the **Search settings** tab.

1. Under **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. Under **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. Under **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.

4. Under **Page search settings**, do the following:
   a. To restrict search paging capabilities to administrators, select the **Allow only administrators to perform page searches** check box.
   b. In the **Idle time out for paged searches (seconds)** field, specify the idle time out for paged searches in seconds.
   c. In the **Maximum number of concurrent paged searches** field, specify the maximum number of outstanding paged results operations allowed by the server at any given time. The default setting is **3**.

   Note: Setting the value to **0** disables paged searches.
5. Under **Sorted search settings**, do the following:
   a. To restrict search sorting capabilities to administrators, select the **Allow only administrators to perform sort searches** check box.
   b. In the **Maximum number of attributes allowed in sorted searches** field, specify the maximum number of attributes that is allowed in sorted searches. The default setting is **3**.

   Note: Setting the value to **0** disables sorted searches.
6. Under **Virtual list view search**, do the following:
   a. To enable or disable virtual list view search, select or clear the **Enable virtual list view search** check box. This control is associated with the ibm-slapdVLVEnabled attribute of the cn=VirtualListView, cn=Configuration entry.

   Note: Virtual list view support can be enabled or disabled dynamically.
   b. In the **Maximum number of entries before offset in a virtual list view search** field, specify the maximum number of entries before offset that each virtual list view search can send. This field is associated with the ibm-slapdMaxVLVBeforeCount attribute of the cn=VirtualListView, cn=Configuration entry.

   Note: For additional information about virtual list view, see "Virtual list view" on page 121
7. Under **Persistent search**, do the following:
   a. Select the **Enable persistent search** check box to enable persistent search.
   b. Enter a numeric value in the **Maximum number of concurrent persistent searches (Max 2000)** field to specify the maximum number of concurrent persistent searches to be allowed.

   Note: For additional information about persistent search, see "Persistent search" on page 122
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.

See “Searching the directory with paging and sorting” on page 118 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

dn: cn=VirtualListView, cn=Configuration
changetype: modify
replace: ibm-slapdVLVEnabled
ibm-slapdVLVEnabled: <value to be set as either true or false>
-
replace ibm-slapdMaxVLVBeforeCount
ibm-slapdMaxVLVBeforeCount: <value to be set in numerals>

dn: cn=Persistent Search, cn=Configuration
changetype: modify
replace: ibm-slapdEnablePersistentSearch
ibm-slapdEnablePersistentSearch: TRUE
-
replace: ibm-slapdMaxPersistentSearches
ibm-slapdMaxPersistentSearches: <value to be set in numerals>
```

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

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

See the **ldapsearch** command information in the *IBM Tivoli Directory Server Version 6.3 Command Reference* for more 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 table.

Note: This restriction is specific to 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 50.

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 "Search Settings" on page 115 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 "Search Settings" on page 115. 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 multiserver 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-slapdPagedResAllowNonAdmin
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-slapdPagedResAllowNonAdmin: true` to `ibm-slapdPagedResAllowNonAdmin: false`, in the ibmslapd.conf file. See "Search Settings" on page 115. If the line does not exist, add it with a value of false to allow only Administrator bind.

ibm-slapdPagedResLmt
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-slapdPagedResLmt** 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-slapdPagedResLmt: 3`, in the ibmslapd.conf file. See "Search Settings" on page 115. 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-slapdIdleTimeOut
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 "Search Settings" on page 115. 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.

If paging is supported on backend servers, then the proxy server will also support page control and register the control in its root DSE. However, the proxy server will not verify the ibm-slapdPagedResAllowNonAdmin and ibm-slapdPagedResLmt values of the backend servers. It is the administrator's responsibility to keep the values in sync. Any error returned by backend server because of difference in value of these two attributes will be considered an error and will be returned to the client.

Virtual list view

Virtual list view (VLV) is a GUI technique that may be employed where ordered lists containing a large number of entries need to be displayed. VLV provides a scrollable view of large sorted data set through a window containing a small number of visible entries.

Note: Tivoli Directory Server support for VLV follows the following Internet Drafts:

- Virtual List View extension for LDAP search operations (draft-ietf-ldapext-ldapv3-vlv-09.txt).
- Virtual List View extension for LDAP C API (draft-smith-ldap-c-api-ext-vlv-00.txt)

VLV search requests include criteria for identifying a desired target entry and the number of entries before (before count) and number of entries after (after count) the target entry. The target entry is specified in the VLV request control by one of two methods-:

- Offset based- This method provides target entry in VLV request control by indicating the target entry’s offset within the list. The list here refers to ordered search result set. The server examines the content count (the client’s estimate of the list count) and offset given by the client and computes the corresponding offset within the list, based on its own idea of content count.
  - An offset with value 1 and content count with value that is not equivalent to 1 indicate that the target is the first entry in the list.
  - If the values of offset and content count are equivalent then it indicates that the target is the last entry in the list.
  - A content count with value zero indicates that the server must use its own content count estimate.

- Assertion based- In this method the client supplies an attribute assertion value. The assertion value is then compared against the values of the attribute specified as the primary sort key in the sort control attached to the search operation. The target entry is identified as the first entry in the list with value greater than or equal to the presented value.

  Note: It is possible that no entry satisfies the conditions specified in assertion based method, in which case there is no target entry.

Consider an example where you need to display names in a telephone directory. Let us consider a telephone directory with the following 10 names in alphabetical order: Ari, Bob, Chris, David, John, Mike, Nancy, Peter, Rosy, and Ted.

Now, consider an offset based vlv search request specifying a sort on attribute "cn" with the following parameters:

```
offset=4
before count=1
after count=1
content count=0 (This means that the server must use its own content count estimate)
```

The search result in this case will yield the following:

```
Chris, David, John
```

Now, consider an assertion based vlv search request with the following parameters-:

```
before count = 1
after count = 1
assertion = Jake
```

The search result in this case will yield the following:

```
David, John, Mike
```

Note: Since Jake is not present, the next entry in sorted order becomes the index entry, which in this case is John.

For information on how to enable vlv, see step 6 on page 116 under **Search settings**.

Persistent search

Persistent search enables LDAP clients to receive notification of changes that occur in an LDAP server. The persistent search mechanism is available to all users. However, ACL checks are enforced on each entry that is returned. This means that users can retrieve only those entries or parts of entries that they have access to. Updates to the directory data that are a part of a transaction are also reported by persistent search. Since the persistent search mechanism is available to all users, it is mandatory to limit the number of concurrent persistent searches that the server will handle. This is done by setting the ibm-slapdMaxPersistentSearches option in the configuration file.

Note: Persistent search is not supported for the subtree cn=Deleted Objects.

Although the persistent search mechanism can keep returning entries, the search size and time limits applicable for non-administrative users will be applicable for persistent search as well. The size and time limits will be applicable irrespective of whether the entries being returned are a part of the initial matching set or the

updated ones. For instance, if the size limit is 500 and 450 entries have been sent as a part of the initial result set, then after 50 update notifications, the persistent search will return LDAP_SIZELIMIT_EXCEEDED error. Similarly, if the time limit is 10 seconds, then, irrespective of whether entries are returned from the initial matching set or update notifications, after 10 seconds an LDAP_TIMELIMIT_EXCEEDED error is returned.

When the persistent search mechanism is used along with paging or sorting, paging or sorting will be applicable only on the initial result set. Also, the change log plug-in will need to run before the persistent search plug-in, if change-log is enabled.

Note: Tivoli Directory Server will return the OID 2.16.840.1.113730.3.4.3 for the attribute ibm-supportedcontrol in case of a root DSE search.

The following addition is made to the configuration file to support the persistent search mechanism:

```
dn: cn=Persistent Search, cn=Configuration
objectclass: top
objectclass: ibm-slapdConfigEntry
objectclass: ibm-slapdPersistentSearch
cn: Persistent Search
ibm-slapdEnablePersistentSearch:  TRUE
ibm-slapdMaxPersistentSearches:  100
```

ibm-slapdEnablePersistentSearch is a Boolean type attribute that determines if persistent search is enabled. This attribute can be assigned a value of either TRUE or FALSE. The default value of this attribute is TRUE. The ibm-slapdMaxPersistentSearches attribute determines the maximum number of concurrent persistent searches allowed. The default value of this attribute is 100 and the maximum allowed value is 2000. For information on how to enable persistent search, see step 7 on page 116 under **Search settings**.

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. See Chapter 19, “Access control lists,” on page 419 for information about ACLs.

Enabling event notification

To enable event notification, use one of the following procedures.

Using Web Administration

If you have not done so already, click **Server administration** in the Web Administration navigation area and then click **Manage server properties** in the expanded list. Next, click 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 Appendix L, "Dynamically-changed attributes," on page 587 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

If you have not done so already, click **Server administration** in the Web Administration navigation area and then click **Manage server properties** in the expanded list. Next, click 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 6.3 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

If you have not done so already, click **Server administration** in the Web Administration navigation area and then click **Manage server properties** in the expanded list. Next, click 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 per transaction is 500. The smaller the number, the better the performance. The default is 5 operations.
4. Under **Timeout between prepare and commit**, select either Seconds or Unlimited. If you select **Seconds**, you must specify in the field the maximum number of seconds allowed between a prepare and commit transaction operation.
5. 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.
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.
7. 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 Appendix L, "Dynamically-changed attributes," on page 587 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

If you have not done so already, click **Server administration** in the Web Administration navigation area and then click **Manage server properties** in the expanded list. Next, click 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 6.3 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=sample.

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=sample'. 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=Deleted Objects, cn=schema and cn=ibmpolicies cannot be added or removed. Consequently, they are not displayed in the panel.

If you have not done so already, click **Server administration** in the Web Administration navigation area and then click **Manage server properties** in the expanded list. Next, click 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>
```

Note:

- See the **idscfgsuf** command information in the *IBM Tivoli Directory Server Version 6.3 Command Reference* for more information.
- You can also use the configuration utility **idsxcfg** to add and remove suffixes. See the *IBM Tivoli Directory Server Version 6.3 Installation and Configuration Guide* for more information.
- To add suffixes using idscfgsuf or idsxcfg the server instance must be stopped.

Removing a suffix

To remove a suffix use, one of the following methods.

Using Web Administration

Note: Defined suffixes such as cn=localhost, cn=Deleted Objects, cn=schema and cn=ibmpolicies cannot be added or removed. Consequently, they are not displayed in the panel.

If you have not done so already, click **Server administration** in the Web Administration navigation area and then click **Manage server properties** in the expanded list. Next, click 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.

You can also use the **idsucfgsuf** command to delete suffixes one at a time:

```
idsucfgsuf -I <instancename> -s <suffixname>
```

Note:

- See the **idsucfgsuf** command information in the *IBM Tivoli Directory Server Version 6.3 Command Reference* for more information.
- You can also use the configuration utility **idsxcfg** to add and remove suffixes. See the *IBM Tivoli Directory Server Version 6.3 Installation and Configuration Guide* for more information.
- To delete suffixes using idsucfgsuf or idsxcfg the server instance must be stopped.

Tombstone - Record deleted entries

Tivoli Directory Server provides the tombstone feature to record information, such as all the attributes, of a to-be deleted entry into a tombstone subtree before the entry gets deleted from the backend database.

Using the tombstone feature, you can move the to-be-deleted entries to the tombstone subtree, cn=Deleted Objects. Subsequently, the attribute table is updated for the entry to mark the entry as deleted by adding an attribute such as isDeleted.

Note:

- This feature is supported only in the primary RDBM backend of the directory server.
- Tombstones are not supported in configuration, schema, or change log backend.
- Tombstone feature is disabled by default.

The tombstone feature is defined by the *ibm-slapdTombstoneEnabled* attribute in the cn=Directory, cn=RDBM Backends, cn=IBM Directory, cn=Schemas, cn=Configuration entry of the ibmslapd.conf file. Additionaly, the

ibm-slapdTombstoneLifetime attribute in the cn=Directory, cn=RDBM Backends, cn=IBM Directory, cn=Schemas, cn=Configuration entry of the configuration file defines the tombstone lifetime. The tombstone lifetime determines the time that deleted entries are retained, the default value being 7 days.

Use any of following methods to enable or disable tombstone-:

Using Web Administration

If you have not done so already, click **Server administration** in the Web Administration navigation area and then click **Manage server properties** in the expanded list. Click the **Delete settings** tab.

This panel allows you to control tombstone configuration parameters. This panel is displayed only to Primary admin or Server config group members.

1. To enable tombstones, click the **Record deleted entries** check box. This control is associated with the ibm-slapdTombstoneEnabled attribute.
2. Under the **Deleted entries lifetime** section, enter a value for tombstone lifetime. You can specify the value in either Days or Hours by selecting the desired value from the combo box. The default value is 7 days. This control is associated with the ibm-slapdTombstoneLifetime attribute.

Using the command line

To enable the tombstone feature, issue the following command:

```
idsldapmodify –D <bindDN> –w <password> –f <file>

where <file> contains:

dn: cn=Directory, cn=RDBM Backends, cn=IBM Directory, cn=Schemas, cn=Configuration
ibm-slapdTombstoneEnabled: TRUE
```

To reread the configuration file, issue the following command:

```
idsldapexop -D <bindDN> -w <password> -op readconfig -scope entire
```

To set the tombstone lifetime value, issue the following command-:

```
idsldapmodify –D <bindDN> –w <password>

dn: cn=Directory, cn=RDBM Backends, cn=IBM Directory, cn=Schemas, cn=Configuration
ibm-slapdTombstoneLifeTime: <value to be set in hours>
```

To reread the configuration file, issue the following command-:

```
idsldapexop -D <bindDN> -w <password> -op readconfig -scope entire
```

You can use the –L parameter of the ldapdelete utility to delete entries under cn=Deleted Objects. To do this, you first display all tombstones under cn=Deleted Objects by issuing the following command:

```
idsldapsearch -b "cn=Deleted Objects"  -r -D <bindDN> -w <password> objectclass=* dn
```

Next, you save the output in an ldif file and then use the ldif file as input to the ldapdelete command by issuing the following command:

```
idsldapdelete -c -L -f <file> -D <bindDN> -w <password>
```

Managing cache properties

If you have not done so already, click **Server administration** in the Web Administration navigation area and then click **Manage cache properties** in the expanded list. This panel has five tabs. You can use these tabs to configure entry cache, filter cache, ACL cache, group members' cache, and attribute cache.

Note: Starting with the IBM Tivoli Directory Server 6.3 release, attribute cache is deprecated. Henceforth, users should avoid using attribute cache.

Entry cache

To configure entry cache, click the **Entry cache** tab and follow the steps given below:

Using Web administration

1. In the **Maximum number of elements in entry cache** field, enter a value for the maximum number of elements to be stored in entry cache.
2. When you are finished, do one of the following:
   - Click **OK** to save your changes and exit this panel.
   - Click **Apply** to apply your changes and stay on this panel.
   - Click **Cancel** to exit this panel without making any changes.

Using command line

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

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

where <filename> contains:

dn: cn=Front End, cn=Configuration
changetype: modify
replace: ibm-slapdEntryCacheSize
ibm-slapdEntryCacheSize: <value to be set in numerals>
```

Filter cache

To configure filter cache, click the **Filter cache** tab and follow the steps given below:

Using Web administration

1. In the **Maximum number of elements in search filters cache** field, enter a value for the maximum number of elements to be stored in search filter cache.
2. Specify the maximum number of elements from a single search operation to be added to the search filter cache. If you select **Elements**, you must enter a numeric value in the field. Otherwise, select **Unlimited**.
3. When you are finished, do one of the following:
   - Click **OK** to save your changes and exit this panel.
   - Click **Apply** to apply your changes and stay on this panel.
   - Click **Cancel** to exit this panel without making any changes.

Using command line

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

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

where <filename> contains:

dn: cn=Front End, cn=Configuration
changetype: modify
replace: ibm-slapdFilterCacheSize
ibm-slapdFilterCacheSize: <value to be set in numerals>

-
replace: ibm-slapdFilterCacheBypassLimit
ibm-slapdFilterCacheBypassLimit: <value to be set in numerals>
```

ACL cache

To configure ACL cache, click the **ACL cache** tab and follow the steps given below:

Using Web administration

1. Select the **Cache ACL information** check box to enable caching of ACL information.
2. In the **Maximum number of elements in ACL cache** field, enter a value for the maximum number of elements to be cached in ACL cache.
3. When you are finished, do one of the following:
   - Click **OK** to save your changes and exit this panel.
   - Click **Cancel** to exit this panel without making any changes.

Using command line

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

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

where <filename> contains:

dn: cn=Front End, cn=Configuration
changetype: modify
replace: ibm-slapdACLCache
ibm-slapdACLCache: TRUE
-
replace: ibm-slapdACLCacheSize
ibm-slapdACLCacheSize: <value to be set in numerals>
```

Group members' cache

The group members' cache is an extension of the Entry cache. This cache stores member and uniquemember attribute values with their entries. To configure the group members' cache, perform either of the following tasks.

Using Web Administration

To configure group members' cache, click the **Group members' cache** tab and follow the steps given below:

1. In the **Maximum number of groups in cache** field, enter a value for the maximum number of groups with members to be cached in the group members' cache.
2. In the **Maximum number of members in a group that can be cached** field, enter a value for the maximum number of members in a group to be cached in the group members' cache.
3. When you are finished, do one of the following:
   - Click **OK** to save your changes and exit this panel.

- Click **Apply** to apply your changes and stay on this panel.
- Click **Cancel** to exit this panel without making any changes.

Using command line

To configure group members' cache 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-slapdGroupMembersCacheSize
ibm-slapdGroupMembersCacheSize:25
-
replace: ibm-slapdGroupMembersCacheBypassLimit
ibm-slapdGroupMembersCacheBypassLimit: 50
```

Adding attributes to and removing attributes from the attribute cache

Note: Starting with the IBM Tivoli Directory Server 6.3 release, attribute cache is deprecated. Henceforth, users should avoid using 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

Note: The attribute cache manager can resolve both exact match filters and presence filters. It can also resolve complex filters that are conjunctive or disjunctive. Additionally, the subfilters within complex filters must be exact match, presence, conjunctive, or disjunctive.

Not all attributes can be added to the attribute cache. To determine if an attribute can be added to the cache, use the ldapexop command:

- For attributes that can be added:

  ```
  ldapexop -D <adminDN> -w <adminPW> -op getattributes -attrType attribute_cache
  -matches true
  ```

- For attributes that cannot be added:

  ```
  ldapexop -D <adminDN> -w <adminPW> -op getattributes -attrType attribute_cache
  -matches false
  ```

Attribute caching can be configured either manually or automatically. To manually configure attribute caching, an administrator first needs to know which attributes to cache. To make attribute caching most effective, the administrator should perform cn=monitor searches. These searches provide information about the attributes that are cached, the amount of memory used by each attribute in the cache, the total amount of memory used by attribute caching, the amount of memory configured for attribute caching, and a list of the attributes most often used in search filters. Using this information, an administrator can determine the

amount of memory that is allowed to be used for attribute caching, as well as which attributes to cache whenever necessary.

Alternatively, when an administrator enables automatic attribute caching, the directory server tracks the combination of attributes that would be most useful to cache within the memory limits defined by the administrator. The server then updates the attribute caching at a particular time and according to a time interval configured by the administrator.

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 83. Also, see "Determining which attributes to cache" in the *IBM Tivoli Directory Server Version 6.3 Performance Tuning and Capacity Planning 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: Click the **Attribute cache** tab and follow the steps given below:

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.

To enable directory automatic attribute caching, perform the following steps:

1. Select the **Enable directory automatic attribute cache** check box. This enables other elements within this group.
2. Enter the start time for directory automatic attribute caching in the **Start Time** text box.
3. From the **Interval** combo box, select the interval at which the directory automatic attribute caching is to be performed again.

To enable change log automatic attribute caching, perform the following steps:

1. Select the **Enable change log automatic attribute cache** check box. This enables other elements within this group.
2. Enter the start time for change log automatic attribute caching in the **Start Time** text box.
3. From the **Interval** combo box, select the interval at which the change log automatic attribute caching is to be performed again.

Note: Automatic attribute caching for change log should not be enabled unless frequent searches within the change log are required and the performance of these searches are critical.

To add an attribute:

1. Select the attribute that you want to cache from the **Available attributes** drop-down 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 Directory and the Changelog containers.

2. Click either **Add to Database** or **Add to Change log** button. The attribute is displayed in the appropriate list box. You can list the same attribute in both containers.
3. Repeat this process for each attribute you want to cache.

   Note: An attribute is removed from the drop-down list when it is added to both the **Cached attributes under Database** and **Cached attributes under Change log** listboxes. If changelog is not enabled, then the **Add to Change log** button is disabled and the entry cannot be added to Cached attributes under Change log list box. The attribute is removed from the available attributes list when it is added to **Cached attributes under Database** list box.
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 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: If you have not done so already, click **Server administration** in the Web Administration navigation area and then click **Manage cache properties** in the expanded list. Next, click the **Attribute cache** tab.

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
```

DB2 password monitoring

DB2 password monitoring feature enables you to configure the server to periodically monitor the DB2 password value in the server configuration file. This ensures that the password value can be used to establish a connection with the database. The DB2 password monitoring feature retrieves the password from the configuration file and uses it to attempt a connection with the database.

A directory server instance relies on the database owner password information in the configuration file to establish a connection with the DB2 database. If the password for the user on the system is not in sync with the value in the configuration file then the connection to the database will fail.

The DB2 password monitoring feature enables monitoring of the DB2 user's password on the system and issues alerts when the password is found to be no longer consistent with what is being used by the directory server instance. When an inconsistency is detected, a message is written to the directory server instance's log file. If auditing is enabled, a message is written to the audit log file as well.

Tivoli Directory Server also enables you to use the Web Admin tool to update the DB2 password while the directory server instance is running.

To update the DB2 password and enable DB2 password monitoring, use one of the following methods-:

Using Web Administration

To update DB2 password:

If you have not done so already, click **Server administration** in the Web Administration navigation area and then click **DB2 Instance Owner** in the expanded list. This panel displays the DB2 instance name and the DB2 instance owner name. On this panel, do the following to change the DB2 administrator's password:

1. Type the new password in the **New password** field.
2. Retype the password in the **Confirm password** field.
3. Click **Change password** to save your changes, or click **Cancel** to return to the "Introduction" panel without making any changes.

Note: This panel is displayed only to Primary Administrator or Server configuration group members. It is better to use SSL when using the Web admin tool to update the DB2 password.

To enable password monitoring:

If you have not done so already, click **Server administration** in the Web Administration navigation area and then click **Manage server properties** in the expanded list. Click the **Database** tab.

Using this panel you can enable DB2 password monitoring and set the password monitoring level. This panel is displayed only to Primary Admin or Server configuration group members. To set the password monitoring level, do the following-:

1. To enable db2 password monitoring, click the **Enable db2 instance password monitoring** check box.
2. Specify the password monitoring interval in the **Password monitoring interval (Max 65535 minutes/45 Days):** field. The default value of this field is 1 day.
3. Click the **OK** button.

Using command line

To update the DB2 password, issue the following command:

```
ldapmodify -h <ldaphost> -p <ldap port> -D <bindDN> -w <password> -f <filename>

where filename contains:

dn:cn=Directory, cn=RDBM Backends, cn=IBM Directory, cn=Schemas, cn=Configuration
changetype: modify
replace: ibm-slapdDbUserPw
ibm-slapdDbUserPw: <password value to be set>
```

Issue the readConfig extended operation to update the password being used by the monitoring function. To do this, issue the following command:

```
ldapexop -op readconfig -scope entire
```

To enable db2 password monitoring, issue the following command-:

```
ldapmodify -h <ldaphost> -p <ldap port> -D <bindDN> -w <password> -f <filename>

where filename contains:

dn:cn=Directory, cn=RDBM Backends, cn=IBM Directory, cn=Schemas, cn=Configuration
changetype: modify
replace: ibm-slapdDbPwMonIntervalMins
ibm-slapdDbPwMonIntervalMins: <value to be set in minutes>
```

Note: The directory server instance will be modified to periodically monitor by default every 24 hours/1440 minutes, or as specified by the ibm-slapdDbPwMonIntervalMins attribute in the configuration file. If the ibm-slapdDbPwMonIntervalMins attribute is set to zero, then no monitoring will be done by the server.

Chapter 11. Securing directory communications

This section describes the steps necessary for keeping the data in your directory secure.

Configuring security settings

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 IBM Tivoli Directory Server, both server authentication and client authentication are supported. See "Secure Sockets Layer" on page 141 and "Transaction Layer Security" on page 141 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. To know about creating Certificate Management Services (CMS) key databases using the GSkit command line utility, see "Using gskcapicmd" on page 148. To manage key databases other than CMS or PKCS11, see "Using ikeyman" on page 152.

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 141 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 141 for more information.

© Copyright IBM Corp. 2002, 2010

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, 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 IBM Tivoli Directory Server and the client application.

   For server authentication to work, 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 IBM Tivoli Directory Server. See "Client authentication" on page 146.

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 IBM Tivoli Directory Server and the administration server for the changes to take effect.

   a. Stop the server. See "Starting and stopping the server" on page 82, if you need information about performing this task.

   b. Stop the administration server 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 server" on page 20, if you need information about performing this task.

   c. Start the administration server. This must be done locally.

      - Issue the command:

        ```
        idsdiradm <instancename>
        ```

      See "Starting an instance of the directory administration server" on page 20, if you need information about performing this task.

   d. Start the server. See "Starting and stopping the server" on page 82, 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 | SSlOnly | TLS | SSLTLS}
```

User must provide the required permissions on the file for the instance owner for which the key database files will be used, and also must restart the server and the administration server 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

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 IBM Tivoli Directory Server, both server authentication and client authentication are supported.

With server authentication, IBM Tivoli Directory Server must have a digital certificate (based on the X.509 standard). This digital certificate is used to authenticate 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, IBM Tivoli Directory Server supplies the client with 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 IBM Tivoli Directory Server and the client application.

For server authentication to work, 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 IBM Tivoli Directory Server. See "Client authentication" on page 146.

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 Tivoli Directory Server for server authentication. These steps assume you have already installed and configured IBM Tivoli Directory Server:

1. Install the IBM GSKit package if it is not installed. See the *IBM Tivoli Directory Server Version 6.3 Installation and Configuration Guide* for information on installing the GSKit package.

   Notes:

   a. If the GSKIT_LOCAL_INSTALL_MODE environment variable is set to true, it allows user to use the GSKit version of their choice based on the path they set in LD_LIBRARY_PATH. If the environment variable is set, then the library using the path set in LD_LIBRARY_PATH, LIB, or LIBPATH is loaded. If this environment variable is not set, then the GSKit library installed on system (for example on UNIX based system: /usr/lib or /usr/lib64, etc) is loaded. This environment variable is supported only on the client server. All server side wrapper scripts explicitly unassign this variable.

   b. The GSKIT_CLIENT_VERSION environment variable is set to the major version of GSKit library. Using this environment variable, user can set the major version number of GSKit library that to use with Tivoli Directory Server. The name of the GSKit libraries change with the change in the major version number. For example, if the name of ssl library shipped with the GSKit 7 is gsk7ssl and with GSKit 8 is gsk8ssl. This environment variable is supported only on the client side. All server side wrapper scripts explicitly unassign this variable.

2. Generate the IBM Tivoli Directory Server private key and server certificate using the **ikeyman** utility. The server's certificate can be signed by a commercial CA, such as VeriSign, or it can be self-signed with the **ikeyman** tool. The CA's public certificate (or the self-signed certificate) must also be distributed to the client application's key database file.

   Note: With Tivoli Directory Server version 6.3, GSkit version 8 is provided. The gskikm utility is not available with GSkit version 8.

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 139 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.

Notes:

1. Only one key database is permitted per ldap server.
2. User must provide the required permissions on the key database files for the instance owner for which the files will be used.
3. For SSL setup in a replication environment, you can have a separate kdb file between supplier and consumer than the one used in the front end of supplier (under cn=SSL, cn=Configuration) to communicate with LDAP client in SSL mode.
4. In case of Proxy Server, if the proxy server is configured for SSL communication with backend server, it uses the same kdb files specified in the server configuration file (under cn=SSL, cn=Configuration).

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 139.

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 server for the changes to take effect.

Server certificate from an external Certificate Authority (CA)

In order to provide a secure connection between IBM Tivoli Directory Server 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 IBM Tivoli Directory Server 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 **ikeyman** 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 **ikeyman** utility is used to generate a private-public key pair and a certificate request. See "Using ikeyman" on page 152 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 6.3 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 **ikeyman** 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 139.
6. If you have more than one certificate in the key database file, the certificate you want to use for IBM Tivoli Directory Server must be the default.
7. Start IBM Tivoli Directory Server.

Note: If you instruct **ikeyman** 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 IBM Tivoli Directory Server in an intranet environment, use **ikeyman** to create your own server certificates. You can also use **ikeyman** to test IBM Tivoli Directory Server 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 **ikeyman** 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 ikeyman" on page 152 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 **ikeyman** 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 Tivoli Directory Server

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 **ikeyman** 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 **ikeyman**.

See "Using ikeyman" on page 152 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 **ikeyman**.
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 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 139.

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 server for the changes to take effect.

Using gskcapicmd

GSKCapiCmd is a tool that can be used to manage keys, certificates, and certificate requests within a CMS key database. GSKCapiCmd supports CMS and PKCS11 key databases. If you are intending to manage key databases other than CMS or PKCS11, you will need to use the Java tool, ikeyman. GSKCapiCmd can be used to manage all aspects of a CMS key database. GSKCapiCmd does not require Java to be installed on the system. For information about the GSKit tool GSKCapiCmd, see the GSKCapiCmd User's Guide.

Using the GSKCapiCmd tool, you can create CMS key database to support server authentication or server client authentication between an LDAP server and a C-based LDAP client. In this example, server authentication and server client authentication between an LDAP server and a C-based LDAP client is performed using the self-signed certificate.

Note: On 32-bit platforms use the gsk8capicmd utility, and on 64-bit platforms use the gsk8capicmd_64 utility.

Configuring server authentication using the CMS key database

To setup server authentication between an LDAP server and C-based LDAP client, do the following:

On the LDAP server system

1. Create a directory on your Tivoli Directory Server system where you want to create and store the key database file and change to the working directory.
2. Create the CMS key database to be used by the IBM Tivoli Directory Server.

   ```
   gsk8capicmd -keydb -create -db serverkey.kdb -pw serverpwd -stash
   ```

   where, *serverkey.kdb* is the key database to be created and *serverpwd* is the password.
3. Create a default self-signed certificate and add it to the serverkey.kdb key database.

   ```
   gsk8capicmd -cert -create -db serverkey.kdb -pw serverpwd \
   -label serverlabel -dn "cn=LDAP_Server,o=sample" -default_cert yes
   ```

   where, the *-dn* value is to uniquely identify the certificate.
4. Extract the certificate from the key database to a file in the binary der format. In this example, the certificate is extracted to a file in binary der format.

   Note: You can also extract the certificate in the base64-encoded ASCII data (.arm).

   ```
   gsk8capicmd -cert -extract -db serverkey.kdb -pw serverpwd \
   -label serverlabel -target server.der -format binary
   ```

5. Configure the Tivoli Directory Server instance to use the certificate in the configuration file.

```
idsldapmodify -h server.in.ibm.com -p 389 -D cn=root -w root \
    -i /home/dsrdbm01/serverauth.ldif
```

where, the *serverauth.ldif* file contains the following:

```
dn: cn=SSL, cn=Configuration
changetype: modify
replace: ibm-slapdSslAuth
ibm-slapdSslAuth: serverAuth

dn: cn=SSL, cn=Configuration
changetype: modify
replace: ibm-slapdSecurity
ibm-slapdSecurity: SSL

dn: cn=SSL, cn=Configuration
changetype: modify
replace: ibm-slapdSslKeyDatabase
ibm-slapdSslKeyDatabase: /home/dsrdbm01/keys/serverkey.kdb

dn: cn=SSL, cn=Configuration
changetype: modify
replace: ibm-slapdSslCertificate
ibm-slapdSslCertificate: serverlabel

dn: cn=SSL, cn=Configuration
changetype: modify
replace: ibm-slapdSslKeyDatabasepw
ibm-slapdSslKeyDatabasepw: serverpwd
```

6. Stop the directory server instance and administration server.

```
ibmslapd -I dsrdbm01 -k
ibmdiradm -I dsrdbm01 -k
```

7. Start the directory server instance and administration server.

```
ibmslapd -I dsrdbm01 -n -t
ibmdiradm -I dsrdbm01 -t
```

On the C-based LDAP client system

1. On the LDAP client system, create a directory where you want to store the key database file and change the working directory.
2. Create the CMS key database file to be used by the C-based LDAP client.

```
gsk8capicmd -keydb -create -db clientkey.kdb -pw clientpwd
```

3. Import the extracted server certificate, server.der, from the server system to the client system.
4. Add the extracted server certificate to the client's key database file.

```
gsk8capicmd -cert -add -db clientkey.kdb -pw clientpwd \
 -label serverlabel -file server.der -format binary
```

5. To verify the added certificate, issue the following command.

```
gsk8capicmd -cert -list -db clientkey.kdb -pw clientpwd
```

To verify the SSL communication between the LDAP client and LDAP server, issue an idsldapsearch command of the following format:

```
idsldapsearch -Z -h server.in.ibm.com -p 636 -K /usr/client/clientkey.kdb \
    -P clientpwd -s base -b "o=sample" objectclass=*
o=sample
objectclass=top
objectclass=organization
o=sample
```

Configuring server client authentication using the CMS key database

To setup server client authentication between an LDAP server and C-based LDAP client, do the following:

On the C-based LDAP client system

1. Create a directory where you want to store the key database file and change the working directory.
2. Create the CMS key database file to be used by the C-based LDAP client.

   ```
   gsk8capicmd -keydb -create -db clientkey.kdb -pw clientpwd
   ```

   where, *clientkey.kdb* is the key database to be created and *clientpwd* is the password.
3. Create a default self-signed certificate and add it to the clientkey.kdb key database.

   ```
   gsk8capicmd -cert -create -db clientkey.kdb -pw clientpwd -label \
    clientlabel -dn "cn=LDAP_Client,o=sample" -default_cert yes
   ```

   where, the *-dn* value is used to uniquely identify the certificate.
4. Extract the certificate from the client's key database to a file in the binary der format. In this example, the certificate is extracted to a file in binary der format.

   Note: You can also extract the certificate in the base64-encoded ASCII data (.arm).

   ```
   gsk8capicmd -cert -extract -db clientkey.kdb -pw clientpwd -label \
     clientlabel -target client.der -format binary
   ```
5. Import the extracted server certificate, server.der, from the server system to the client system.
6. Add the extracted server certificate to the client's key database file.

   ```
   gsk8capicmd -cert -add -db clientkey.kdb -pw clientpwd \
    -label serverlabel -file server.der -format binary
   ```

On the LDAP server system

1. Create a directory on your Tivoli Directory Server system where you want to create and store the key database file and change the working directory.
2. Create the CMS key database to be used by the Tivoli Directory Server.

   ```
   gsk8capicmd -keydb -create -db serverkey.kdb -pw serverpwd -stash
   ```

   where, *serverkey.kdb* is the key database to be created and *serverpwd* is the password.
3. Create a default self-signed certificate and add it to the serverkey.kdb key database.

   ```
   gsk8capicmd -cert -create -db serverkey.kdb -pw serverpwd -label \
     serverlabel -dn "cn=LDAP_Server,o=sample" -default_cert yes
   ```

where, the *-dn* value is used to uniquely identify the certificate.

4. Extract the certificate from the server's key database to a file in the binary der format. In this example, the certificate is extracted to a file in binary der format.

   Note: You can also extract the certificate in the base64-encoded ASCII data (.arm).

   ```
   gsk8capicmd -cert -extract -db serverkey.kdb -pw serverpwd \
    -label serverlabel -target server.der -format binary
   ```

5. Import the extracted client certificate, client.der, from the client system to the server system.
6. Add the extracted client certificate to the server's key database file.

   ```
   gsk8capicmd -cert -add -db serverkey.kdb -pw serverpwd \
    -label clientlabel -file client.der -format binary
   ```

7. Configure the Tivoli Directory Server instance to use the certificate in the configuration file.

   ```
   idsldapmodify -h server.in.ibm.com -p 389 -D cn=root -w root \
      -i /home/dsrdbm01/clientserverauth.ldif
   ```

   where, the *clientserverauth.ldif* file contains the following:

   ```
   dn: cn=SSL, cn=Configuration
   changetype: modify
   replace: ibm-slapdSslAuth
   ibm-slapdSslAuth: serverClientAuth

   dn: cn=SSL, cn=Configuration
   changetype: modify
   replace: ibm-slapdSecurity
   ibm-slapdSecurity: SSL

   dn: cn=SSL, cn=Configuration
   changetype: modify
   replace: ibm-slapdSslKeyDatabase
   ibm-slapdSslKeyDatabase: /home/dsrdbm01/cskeys/serverkey.kdb

   dn: cn=SSL, cn=Configuration
   changetype: modify
   replace: ibm-slapdSslCertificate
   ibm-slapdSslCertificate: serverlabel

   dn: cn=SSL, cn=Configuration
   changetype: modify
   replace: ibm-slapdSslKeyDatabasepw
   ibm-slapdSslKeyDatabasepw: serverpwd
   ```

8. Stop the directory server instance and administration server.

   ```
   ibmslapd -I dsrdbm01 -k
   ibmdiradm -I dsrdbm01 -k
   ```

9. Start the directory server instance and administration server.

   ```
   ibmslapd -I dsrdbm01 -n -t
   ibmdiradm -I dsrdbm01 -t
   ```

To verify the SSL communication between the client and server, issue an idsldapsearch command of the following format on the client system.

```
idsldapsearch -Z -h server.in.ibm.com -p 636 -K /usr/client/clientkey.kdb \
 -P clientpwd -s base -b "o=sample" objectclass=*
o=sample
objectclass=top
objectclass=organization
o=sample
```

Using ikeyman

The following key-management program, **ikeyman**, is provided with IBM JAVA. It is a user-friendly GUI for managing key files, which is implemented as a Java applet. IBM JAVA version 6 is available when you install IBM Tivoli Directory Server version 6.3. The ikeyman utility is available on Windows in the <*TDS_Install_Directory*>\java\jre\bin directory, on Linux in the /opt/ibm/ldap/V6.3/java/jre/bin directory, and on AIX and Solaris systems in the /opt/IBM/ldap/V6.3/java/jre/bin directory.

Note: If you are prompted to set JAVA_HOME, you can set it to the java subdirectory of the IBM Tivoli Directory Server. If you use IBM Tivoli Directory Server, you also need to set the LIBPATH environment variable as follows:

On Linux platform

```
$export LIBPATH=$JAVA_HOME/bin:$JAVA_HOME/jre/bin:$LIBPATH
```

On Windows platform

```
c:\> set LIB=%JAVA_HOME%\bin; %JAVA_HOME%\jre\bin; %LIB%
```

On AIX systems use the LIBPATH environment variable to specify the library path, and on Solaris systems use the LD_LIBRARY_PATH environment variable.

Use **ikeyman** 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 IBM Tivoli Directory Server require **CMS** key database file types.

The tasks you can perform with **ikeyman** 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 158.

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 **ikeyman**.
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 **ikeyman** Java utility by typing:
   ```
   ikeyman
   ```
2. Select **Key Database File**.
3. Select **New** (or **Open** if the key database already exists).
4. Specify a key database type, 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 **ikeyman** 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 `ikeyman` to start the Java utility.
2. Select **Key Database File**.
3. Select **Open**.
4. Specify the key database type, 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 `ikeyman` to start the Java utility.
2. Select **Key Database File**.
3. Select **Open**.
4. Specify the key database type, 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 `ikeyman` to start the Java utility.
2. Select **Key Database File**.
3. Select **Open**.
4. Specify the key database type, 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** from the list under the **Key database content** section.
   - 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** from the list under the **Key database content** section.
   - 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 `ikeyman` to start the Java utility.
2. Select **Key Database File**.
3. Select **Open**.

4. Specify the key database type, key 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 from the list under the **Key database content** section (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 `ikeyman` to start the Java utility.
2. Select **Key Database File**.
3. Select **Open**.
4. Specify the key database type, 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** from the list under the **Key database content** section.
7. Select the desired certificate.
8. Click **View/Edit**.
9. Select the **Set the certificate 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 153.

The basic steps to creating a self-signed certificate are:

1. Type `ikeyman` 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 type, 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 desired Signature Algorithm.
   - 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 `ikeyman` to start the Java utility.
2. Select **Key Database File**.
3. Select **Open**.
4. Specify the key database type, 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** from the list under the **Key database content** section.
7. Select the desired certificate.
8. Click **Export/Import**.
9. For **Action type**, select **Export Key**.
10. Select the Key file type.

    Note: IBM Tivoli Directory Server requires CMS key database file types.
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 `ikeyman` to start the Java utility.
2. Select **Key Database File**.
3. Select **Open**.
4. Specify the key database type, 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** from the list under the **Key database content** section.
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 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 defined in each new key database:

- Entrust.net Certification Authority (2048)
- Entrust.net Client Certification Authority
- Entrust.net Global Client Certification Authority
- Entrust.net Global Secure Server Certification Authority
- Entrust.net Secure Server Certification Authority
- RSA Secure Server Certification Authority
- Thawte Personal Basic CA
- Thawte Personal Freeemail CA
- Thawte Personal Premium CA
- Thawte Premium Server CA
- Thawte Server CA
- VeriSign Class 1 Public Primary Certification Authority
- VeriSign Class 1 Public Primary Certification Authority - G2
- VeriSign Class 1 Public Primary Certification Authority - G3
- VeriSign Class 2 Public Primary Certification Authority
- VeriSign Class 2 Public Primary Certification Authority - G2
- VeriSign Class 2 Public Primary Certification Authority - G3
- VeriSign Class 3 Public Primary Certification Authority
- VeriSign Class 3 Public Primary Certification Authority - G2
- VeriSign Class 3 Public Primary Certification Authority - G3

- VeriSign Class 4 Public Primary Certification Authority - G2
- VeriSign Class 4 Public Primary Certification Authority - G3

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 `ikeyman` to start the Java utility.
2. Select **Key Database File**.
3. Select **Open**.
4. Specify the key database type, 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** from the list under the **Key database content** section.
7. Click **Populate**.
8. From the Add CA Certificates dialog box, select the desired certificates.
9. Click **View/Edit**.
10. Check the **Set the certificate as a trusted root** check box, and click **OK**.
11. 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 defined in each new key database:

- Entrust.net Certification Authority (2048)
- Entrust.net Client Certification Authority
- Entrust.net Global Client Certification Authority
- Entrust.net Global Secure Server Certification Authority
- Entrust.net Secure Server Certification Authority
- RSA Secure Server Certification Authority
- Thawte Personal Basic CA
- Thawte Personal Freeemail CA
- Thawte Personal Premium CA
- Thawte Premium Server CA
- Thawte Server CA
- VeriSign Class 1 Public Primary Certification Authority
- VeriSign Class 1 Public Primary Certification Authority - G2
- VeriSign Class 1 Public Primary Certification Authority - G3
- VeriSign Class 2 Public Primary Certification Authority
- VeriSign Class 2 Public Primary Certification Authority - G2
- VeriSign Class 2 Public Primary Certification Authority - G3
- VeriSign Class 3 Public Primary Certification Authority
- VeriSign Class 3 Public Primary Certification Authority - G2
- VeriSign Class 3 Public Primary Certification Authority - G3
- VeriSign Class 4 Public Primary Certification Authority - G2
- VeriSign Class 4 Public Primary Certification Authority - G3

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 `ikeyman` to start the Java utility.
2. Select **Key Database File**.
3. Select **Open**.
4. Specify the key database type, 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** from the list under the **Key database content** section.
7. Select the desired certificate.
8. Click **View/Edit**.
9. Clear the **Set the certificate as a trusted root** check 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 `ikeyman` to start the Java utility.
2. Select **Key database file**.
3. Select **Open**.
4. Specify the key database type, 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** from the list under the **Key database content** section.
7. Select the desired certificate.
8. Click **Export/Import**.
9. For **Action type**, select **Export Key**.
10. Select the desired key file type.
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 **ikeyman** program can be used to migrate an existing key ring file, as created with **mkkf**, to the format used by **ikeyman**.

To migrate a key ring file:

1. Type `ikeyman` to start the Java utility.
2. Select **Key Database File**.
3. Select **Open**.
4. Specify the key database type, 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** 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

If you have not done so already, click **Server administration** in the Web Administration navigation area and then click **Manage security properties** in the expanded list. Next, click 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.
4. When you are finished, click **OK** to apply your 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 6.3 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 server for the changes to take effect.

PKCS#11

PKCS#11 is an interface that enables an LDAP user to use crypto hardware. Using PKCS#11, an LDAP user can use the crypto hardware to securely store the key database file and accelerate cryptographic operations as well. The PKCS#11 interface can be used to configure the following types of crypto devices:

- Accelerators: These devices are usually connected to the host by a permanent connection such as a card slot or a LAN connection. The primary purpose of an accelerator is to increase the number of cryptographic operations per second for a server. Private key storage is maintained in an SSL KDB (Key Database) file, which is loaded into the accelerator as needed. This type of device should be considered for use when the objective is to increase the number of cryptographic operations only and stronger hardware protection of the server's private key is not a concern.
- Key storage with accelerators: These devices are primarily for server applications where cryptographic performance is an issue and stringent security of the server's private key is also essential. The private key and certificate are stored on the device. If a cryptographic operation requires use of the private key, the hardware device uses the key locally on the adapter. The application can never access the key in an un-encrypted format. These devices usually employ tamper-resistant procedures to protect external access to the key.

How to configure the server to use PKCS#11 interface

The directory server can be configured to use the PKCS#11 interface under the entry “dn: cn=SSL, cn=Configuration”.

Using Web Administration

Expand the **Server administration** category in the navigation area of the Web Administration Tool and click **Manage security properties** tab. Next, click the **PKCS#11 settings** tab. The **PKCS#11 settings** panel is displayed. This panel is displayed only if the root DSE search on ibm-supportedCapabilities returns the PKCS#11 interface support OID 1.3.18.0.2.32.67.

Note: For the settings specified in this panel to take effect, you must select the **Enable PKCS#11 interface support** check box in the **Settings** panel under **Manage security properties** category.

To set PKCS#11 interface supported hardware:

1. Select the **Enable crypto hardware key storage** check box to specify the key storage location as the crypto hardware.
2. Select the required acceleration facility of the crypto hardware by selecting the **Symmetric cipher**, **Digest**, or **Random data generator** check box.

   Note: You can select one or more check boxes under the Accelerator mode options section.
3. In the **Crypto hardware library path and file name** text box, specify the library path of the crypto hardware driver to be accessed using the PKCS#11 interface.
4. In the **Token password** text box, specify the password to be used for accessing the crypto hardware slot.
5. In the **Confirm password** text box, reenter the password.
6. In the **Token label** text box, specify the token label of the crypto hardware’s slot to be accessed.
7. When you are finished, do one of the following:
   a. Click **OK** to apply your changes and exit this panel.
   b. Click **Apply** to apply your changes and stay on this panel.
   c. Click **Cancel** to exit this panel without making any changes.

Note: You must restart the server for the changes to take effect.

Using the command line

To configure the server to use PKCS#11 interface using the command line, issue the following command:

```
ldapmodify –D <adminDN> –w <adminPW> -i <filename>

where <filename> contains:

dn: cn=ssl,cn=configuration
changetype: modify
replace: ibm-slapdSecurity
ibm-slapdSecurity: SSLOnly
-
replace: ibm-slapdSslAuth
ibm-slapdSslAuth: serverauth
-
replace: ibm-slapdSslCertificate
ibm-slapdSslCertificate: tlabel1
-
replace: ibm-slapdSslPKCS11Enabled
ibm-slapdSslPKCS11Enabled: True
-
replace: ibm-slapdSslPKCS11Lib
ibm-slapdSslPKCS11Lib: /opt/nfast/toolkits/pkcs11/libcknfast.so
-
replace: ibm-slapdSslPKCS11Keystorage
ibm-slapdSslPKCS11Keystorage: true
-
replace:  ibm-slapdSslPKCS11TokenLabel
ibm-slapdSslPKCS11TokenLabel: OpCard
-
replace: ibm-slapdSslPKCS11TokenPW
ibm-slapdSslPKCS11TokenPW: PASSWORD
```

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: IBM Tivoli Directory Server 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 12. 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
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.

4. 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.

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 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 12 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 server for the changes to take effect.

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. Select an LDAP server and port that contains the certificate revocation list from the **Server hostname:port** drop-down list or enter a host name and port number of a server in the field in the hostname:port format.
2. In the **Bind DN** field, specify the bind DN used for connection to the server. If a bind DN is not specified, an anonymous bind is used.
3. In the **Bind password field**, specify the bind password. 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: 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-slapdCrlHost
ibm-slapdCrlHost: <newhostname>
-
replace: ibm-slapdCrlPassword
ibm-slapdCrlPassword: <password>
-
replace: ibm-slapdCrlPort
ibm-slapdCrlPort: <portnumber>
-
replace: ibm-slapdCrlUser
ibm-slapdCrlUser: <username>
```

You must restart the server and the administration server for the changes to take effect.

Chapter 12. Securing directory access

Access to directory data can be fully controlled by the directory administrator. LDAP directories require clients to perform a bind operation that identifies who is trying to use the directory. Tivoli Directory Server supports several bind mechanisms:

- Simple
- DIGEST-MD5
- Kerberos (also known as GSSAPI)
- EXTERNAL

The directory server supports pass-through authentication, which allows the administrator to configure the directory server to use other directory servers such as OpenLDAP or Active Directory to provide authentication for the binds. See "Pass-through authentication" on page 193.

Simple binds require a DN and a password. If no DN is supplied, the binds are said to be anonymous. The administrator can configure the directory so that anonymous binds are not allowed. (See "Managing connection properties" on page 106.) Generally, the DN corresponds to an entry in the directory. The password used for binding to the directory server is the value of the userpassword attribute associated with the entry with the given DN. The directory server can be configured to enforce password policies that determine what kinds of values passwords can have and how often they must be changed. (See "Setting password policy" on page 170.) The password data stored in the directory is encrypted. (See "Password encryption.") The directory administrator can delegate some administrative responsibilities by configuring an administrative group. The members of this group can be assigned specific authorities in the directory. The DN and passwords for these are stored as part of the server configuration. The passwords are encrypted and an administrative password policy can be configured. See "Setting the administration password and lockout policy" on page 181.

DIGEST-MD5 and Kerberos (GSSAPI) are described in this chapter. The EXTERNAL mechanism, also referred to as PKI or certificate based authentication, relies on the authentication performed by a directory server using SSL or TLS when the server is configured for server and client authentication. The client connection is established only after the client provides a certificate issued by a Certifying Authority (CA) trusted by the server. The client's certificate has a DN and it is this DN that is used to identify the user of this client connection. See "Configuring security settings" on page 139 for information about how to configure a directory server to support EXTERNAL binds.

Password encryption

IBM Tivoli Directory Server 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.

© Copyright IBM Corp. 2002, 2010

One-way encrypting formats:

- crypt
- MD5
- SHA-1
- Salted SHA-1
- SHA-2
- Salted SHA-2

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.

MD5 Passwords are encrypted by the MD5 Message Digest 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.

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

SHA-2
Passwords are encrypted by the SHA-2 family of encrypting algorithm

before they are stored in the directory. The supported encryption schemes under the SHA-2 family of encryption algorithm are:

- SHA-224
- SHA-256
- SHA-384
- SHA-512

Salted SHA-2
Passwords are encrypted by the Salted SHA-2 family of encrypting algorithm before they are stored in the directory. The supported encryption schemes under the Salted SHA-2 family of encryption algorithm are:

- SSHA-224
- SSHA-256
- SSHA-384
- SSHA-512

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 6.3 Installation and Configuration Guide* for additional information about the configuration file.

To specify the type of password encryption, use one of the following methods:

Using Web Administration

If you have not done so already, click **Server administration** in the Web Administration navigation area and then click **Manage security properties** in the expanded list. Click the **Password encryption** tab.

To set password encryption:

1. Select a password encryption type from the **Set the password encryption mechanism** combo box.
2. When you are finished, do one of the following:
   - Click **OK** to apply your changes and exit this panel.
   - Click **Apply** to apply your changes and stay on this panel.
   - Click **Cancel** to exit this panel without making any changes.

Using the command line

To change the type of encryption 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: <password encryption mechanism>

Here, the ibm-slapdPWEncryption attribute can be assigned any of the
following values: none,aes128,aes192,aes256,crypt,sha,ssha, md5
sha224, sha256, sha384, sha512, ssha224, ssha256, ssha384, or ssha512.
```

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.

Setting password policy

Password policy is a set of rules that controls how passwords are used and administered in IBM Tivoli Directory Server. 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 can also restrict the reuse of old passwords and ensure that users are locked out after a defined number of failed bind attempts.

When an administrator sends a request to turn on password policy, the ibm-pwdPolicyStartTime attribute is generated by the server. This attribute is an optional attribute which cannot be deleted or modified by a client request. Only administrators with administrative control can modify the ibm-pwdPolicyStartTime attribute. The value of this attribute is changed when the Password Policy is turned on and off by an administrator. When the ibm-pwdPolicyStartTime attribute

is turned on and off, the value of the attribute gets reset and the user entry's last changed time which is evaluated based on the entry's modifyTimestamp and the ibn-pwdPolicyStartTime may get changed. As a result, some old passwords which would have expired may not expire when the password policy is turned off and on.

Note: It is essential to note that a password policy entry has to be created before it can be associated with a user or a group entry as an individual or a group password policy. If the referenced password policy entry does not exist, a message "unwilling to perform" is returned. Once a password policy entry has been referenced by a user or group entry, it cannot be renamed or deleted unless the association between the entry and the user or group entry has been removed.

For additional information about passwords see "Password Guidelines" on page 179.

Tivoli Directory Server provides three types of password policies: individual, group, and global password policies.

Global Password Policy

When a global password policy entry (cn=pwdpolicy,cn=ibmpolicies) is created by the server, the attribute ibm-pwdPolicy is set to FALSE, which is the default value. This means that all password policy entries will be ignored by the server. Only when the ibm-pwdPolicy attribute is set to TRUE the password rules are enforced by the server. When a global password policy is enforced and the ibm-pwdGroupAndIndividualEnabled attribute in cn=pwdpolicy,cn=ibmpolicies is set to TRUE, the group and individual password policies are also considered when evaluating the password policy.

Note: A global administrative group member, primary administrator, and local admin group members with administrative control can enable/disable group and individual password policies.

Group Password Policy

The group password policy enables members of a group to be controlled by a set of special password rules. For group password policy, ibm-pwdGroupPolicyDN attribute pointing to a password policy entry can be used in any user group objects such as accessGroup, accessRole, and groupOfNames.

Since a user entry may belong to more than one group, multiple group password policy entries will be evaluated before the user's group policy can be determined. In order to evaluate a composite group policy, group password policy entries are combined to form a union of attributes with the most restrictive attribute values taking precedence.

Individual Password Policy

Individual password policy enables every user entry to have its own password policy. For individual password policy, attribute ibm-pwdIndividualPolicyDN pointing to a password policy entry can be used to extend a user to have its own password policy entry. By changing the attributes of the password policy entry, an administrator can effectively manage password policy for a set of users without modifying any of the user entries.

Note: By assigning a value of cn=noPwdPolicy to attribute ibm-pwdIndividualPolicyDN for a password policy extended user entry, an administrator may exempt a user from any password policy controls.

Password Policy Evaluation

To evaluate a user's effective password policy, all password policies associated with a user are taken into consideration starting with the individual password policy. Next, the group password policy is considered and finally the global password policy is taken into consideration. If an attribute is not defined in the individual password policy entry, it will be searched in the composite group password policy entry. If it is not found in the composite group policy entry, the attribute in the global password policy entry will be used. In case the attribute is not defined in the global password policy entry, then the default value will be assumed.

Note: The effective password policy extended operation (**effectpwdpolicy**) is used to display the effective password policy of a given user. Information about the password policy entries which are used to calculate the effective password policy is also displayed using this extended operation. For more information about this extended operation, see the *IBM Tivoli Directory Server Version 6.3 Command Reference*.

Evaluation of a user's Group Password Policy

Since a user entry may belong to more than one group, multiple group password policy entries may be evaluated to determine a user's composite group policy. Following are the rules for determining a user's composite group password policy:

1. If ibm-pwdPolicy is set to False in a Password policy entry, no attributes defined in the entry will be used to determine the composite group password policy. If the attribute is not set, then the default value of False is assumed for the attribute.
2. If ibm-pwdGroupPolicyDN has a value of cn=noPwdPolicy in all the groups that a user belongs to, no composite group password will be evaluated for the user. In this case, unless the user has an individual password policy defined, no policy (not even the global) will be applied.
3. An attribute defined with a non-default value is more restrictive than if defined with a default value which, in turn, is more restrictive than if it is not defined at all.
4. The password policy attributes passwordMinAlphaChars, pwdMinLength, and passwordMinOtherChars are interdependent. For instance, the value of passwordMinAlphaChars must be set to less than or equal to the value in pwdMinLength minus the value in passwordMinOtherChars. Due to this inter-dependency among attribute values, if one attribute is selected from a policy, then the other two attributes are also selected from the same policy.

   The order of selection will be pwdMinLength, passwordMinOtherChars, and passwordAlphaChars. This means that the selection will be based on picking the largest value for pwdMinLength. In case of a situation where two group policies have the same value for the pwdMinLength attribute, then the one with the largest value for passwordMinOtherChars will be selected. Once an attribute is selected, the other two attributes will be selected automatically.
5. The passwordMaxConsecutiveRepeatedChars attribute is used to restrict the maximum successive repetitions of a given character in the password. Both passwordMaxRepeatedChars and passwordMaxConsecutiveRepeatedChars can be enabled or disabled independent of each other. However, if both these attributes are enabled, then the following rules are applicable-:

- The value of passwordMaxRepeatedChars attribute must be greater than or equal to the value of passwordMaxConsecutiveRepeatedChars attribute.
- In case multiple password policies are enabled, passwordMaxConsecutiveRepeatedChars will be picked up from the same policy as was used to pick up passwordMaxRepeatedChars. If passwordMaxRepeatedChars is disabled in all policies, then the most restrictive value of passwordMaxConsecutiveRepeatedChars would be picked up.
- If the passwordMaxConsecutiveRepeatedChars attribute is set to 0, then the number of consecutive repeated characters is not checked. If passwordMaxConsecutiveRepeatedChars is set to 1, then a given character cannot be immediately followed by another character of the same type. For instance, if the passwordMaxConsecutiveRepeatedChars attribute is set to 1 then 'aba' is a valid value for a password but 'aab' will be an invalid value.

  Similarly, if the passwordMaxConsecutiveRepeatedChars attribute is set to 2, then the maximum number of times a character can occur consecutively in a password is 2.

6. Attributes in all the group password policy entries are combined to form a union of attributes with the most restrictive attribute values taking precedence. Given below is a table that describes how the most restrictive attribute values are determined:

Table 13. Determining the most restrictive attribute values

Password Policy Attribute	Description	More restrictive value	Valid values	Default values
pwdAttribute	The pwdAttribute attribute specifies the name of the attribute to which the password policy is being applied. This attribute can only be set to the userPassword attribute.	N/A	userPassword	userPassword
pwdMinAge	The pwdMinAge attribute specifies the number of seconds that must pass since the modification of last password, before modifying a password.	Greater	Greater than or equal (GE) to 0	0 - no age limit
pwdMaxAge	The pwdMaxAge attribute specifies the number of seconds after which a password will expire (0 means password does not expire).	Less	GE 0	0 - password does not expire
pwdInHistory	The pwdInHistory attribute specifies the number of passwords that are stored in the pwdHistory attribute.	Greater	0 to 10	0 - no password history
pwdCheckSyntax	The pwdCheckSynatx attribute indicates whether the password will be checked for syntax. The values of the pwdCheckSynatx attribute indicates the following: • '0' means syntax checking will not be enforced • '1' means the server will check the syntax, and if the server is unable to check the syntax (due to a hashed password or other reasons) it will be accepted • '2' means the server will check the syntax, and if the server is unable to check the syntax it returns an error refusing the password	Greater	0, 1, 2 1 – if server not able to check the syntax, then accept password 2 – if server is not able to check the syntax, then reject the password	0

Table 13. Determining the most restrictive attribute values (continued)

Password Policy Attribute	Description	More restrictive value	Valid values	Default values
pwdMinLength	The pwdMinLength attribute specifies the minimum length that must be set for the password string. The server will check the minimum length depending upon the value of the pwdCheckSyntax attribute.	Greater	GE 0	0 – no minimum length
pwdExpireWarning	The pwdExpireWarning attribute specifies the maximum number of seconds before a password is about to expire during which the expiration warning messages will be returned to an authenticating user.	Greater	GE 0	0 – no warnings will be sent
pwdGraceLoginLimit	The pwdGraceLogingLimit attribute specifies the number of times an expired password can be used to authenticate user.	Less	GE 0	0 – no grace login
pwdLockout	The pwdLockout attribute indicates whether a password can be used to authenticate after a specified number of consecutive failed bind attempts.	True	True/False	False
pwdLockoutDuration	The pwdLockoutDuration attribute specifies the number of seconds that the password cannot be used to authenticate due to the specified 'pwdMaxFailure' failed bind attempts.	Greater	GE 0	0 – locked out until reset
pwdMaxFailure	The pwdMaxFailure attribute specifies the maximum number of consecutive failed bind attempts allowed, after which the password will not be considered for authentication. If a value of 0 is set to the pwdMaxFailure attribute then the value of pwdLockout will be ignored.	Less	GE 0	0 – no failure count, no lockout
pwdFailureCountInterval	The pwdFailureCountInterval attribute specifies the number of seconds after which the password failures are removed from the failure counter even if no successful authentication has happened.	Greater	GE 0	0 – no count, reset by successfully authentication
pwdMustChange	The pwdMustChange attribute specifies whether the users must change their passwords when they first bind to the directory after the administrator has reset their passwords.	True	True/False	True/False if cn=noPwdPolicy
pwdAllowUserChange	The pwdAllowUserChange attribute specifies whether the users are allowed to change their own passwords.	True	True/False	True
pwdSafeModify	The pwdSafeModify attribute specifies whether the existing password must be sent when changing a password.	True	True/False	False
ibm-pwdPolicy	The ibm-pwdPolicy attribute specifies whether the password policy is to be turned ON or OFF.	True	True/False	False
passwordMinAlphaChars	The passwordMinAlphaChars attribute specifies the minimum number of alphabet characters that the password string must have. If the server is unable to check the number of alphabetic characters, then the server will continue processing depending on the value of the pwdCheckSyntax attribute.	Greater	GE 0	0 – no min alpha will be enforced

Table 13. Determining the most restrictive attribute values (continued)

Password Policy Attribute	Description	More restrictive value	Valid values	Default values
passwordMinOtherChars	The passwordMinOtherChars attribute specifies the minimum number of numeric and special characters that the password string must have. If the server is unable to check the number of other characters, then the server will continue processing depending on the value of the pwdCheckSyntax attribute.	Greater	GE 0	0 – no min other char
passwordMaxRepeatedChars	The passwordMaxRepeatedChars attribute specifies the maximum number of times a given character can be used in a password. If the server is unable to check the actual password characters, then the server will continue processing depending on the value of the pwdCheckSyntax attribute.	Less	GE 0	0 – no max repeated char
passwordMaxConsecutive RepeatedChars	The passwordMaxConsecutiveRepeatedChars attribute is used to restrict the maximum successive repetitions of a given character in the password.	Less	GE 0	0 – no maximum consecutive repeated character
passwordMinDiffChars	The passwordMinDiffChars attribute specifies the minimum number of characters in the new password that must be different from the characters in the old password, and any passwords stored in the pwdHistory. If the password has been one-way encrypted the server is unable to check actual password characters, then the server will continue processing depending on the value of the pwdCheckSyntax attribute.	Greater	GE 0	0 - no minimum number of different characters between passwords

Based on the rules defined above, a user's composite group policy is determined. To gain a better understanding of how a composite group policy is determined, consider some examples given in the table below:

Table 14. Determining the composite group policy

Group X password policy	Group Y password policy	Group Z password policy	Composite group password policy
pwdMaxAge = 86400 pwdSafeMode = True pwdMaxFailure = 5 ibm-pwdPolicy = True ibm-pwdPolicyStarttime = 20060406200000	pwdMaxAge = 43200 pwdSafeMode = False ibm-pwdPolicy = True ibm-pwdPolicyStarttime = 20060306200000	pwdCheckSytax = 1 ibm-pwdPolicy = True ibm-pwdPolicyStarttime = 20060506200000	pwdMaxAge = 43200 pwdSafeMod = True pwdCheckSytax = 1 pwdMaxFailure = 5 ibm-pwdPolicy = True ibm-pwdPolicyStarttime = 20060306200000
pwdMaxAge = 86400 ibm-pwdPolicy = True	pwdMaxAge = 43200 pwdSafeMode = True	pwdMaxAge = 0 ibm-pwdPolicy = True	pwdMaxAge = 86400 pwdSafeMode = False ibm-pwdPolicy = True **Note:** Group Y's passwd policy is not used in calculating composite group policy, since its ibm-pwdPolicy is not defined and therefore it defaults to FALSE.
pwdMinLength = 10 passwordMinOtherChars = 4 passwordMinAlphaChars= 6 ibm-pwdPolicy = True	pwdMinLength = 12 ibm-pwdPolicy = True		pwdMinLength = 12 ibm-pwdPolicy = True

Table 14. Determining the composite group policy (continued)

Group X password policy	Group Y password policy	Group Z password policy	Composite group password policy
pwdMinLength = 10 passwordMinOtherChars = 4 passowrdMinAlphaChars = 6 ibm-pwdPolicy = True		pwdMinLength = 10 passwordMinOtherChars = 5 passwordMinAlphaChars = 3 ibm-pwdPolicy = True	pwdMinLength =10 passwordMinOtherChars = 5 passwordMinAlphaChars = 3 ibm-pwdPolicy = True
passwordMaxConsecutiveRepeatedChars=0 passwordMaxRepeatedChars=5 ibm-pwdPolicy = True	passwordMaxConsecutiveRepeatedChars=2 ibm-pwdPolicy = True	passwordMaxRepeatedChars=3 ibm-pwdPolicy = True	passwordMaxRepeatedChars=3 passwordMaxConsecutiveRepeatedChars=0 ibm-pwdPolicy = True
passwordMaxConsecutiveRepeatedChars=4 passwordMaxRepeatedChars=0 ibm-pwdPolicy = True	passwordMaxConsecutiveRepeatedChars=1 passwordMaxRepeatedChars=0 ibm-pwdPolicy = True		passwordMaxConsecutiveRepeatedChars=1 passwordMaxRepeatedChars=0 ibm-pwdPolicy = True
passwordMaxConsecutiveRepeatedChars=4 passwordMaxRepeatedChars=2 ibm-pwdPolicy = True	passwordMaxConsecutiveRepeatedChars=2 passwordMaxRepeatedChars=3 ibm-pwdPolicy = True		passwordMaxConsecutiveRepeatedChars=4 passwordMaxRepeatedChars=2 ibm-pwdPolicy = True

Evaluation of a user's Effective Password Policy

A user's effective password policy is evaluated only if the ibm-pwdPolicy attribute is set to TRUE in the global password policy entry. Other password policies, such as individual and group policy, can still be enabled when the global policy is disabled, but these policy rules will have no effect on the user.

The attribute ibm-pwdPolicyStartTime is set to the current system time when ibm-pwdPolicy is set to TRUE. This can be done even if the global password policy entry is set to FALSE. However, the ibm-pwdPolicyStartTime value will not be used for effective policy evaluation unless the global policy is enabled. Once the global policy is enabled, the value of this attribute will be selected from a user's individual, then group and then the global policy. Since ibm-pwdPolicyStartTime exists in every active password policy, the start time of an individual policy, if it exists, will always override any other policy start time as the start time of the user's effective password policy.

Given below is a set of examples that explain how a user's effective password policy is determined.

Table 15. Determining the effective password policy

Individual password policy	Group password policy	Global password policy	Effective password policy
pwdMaxAge = 86400 ibm-pwdPolicy = True pwdMinAge = 21600 pwdLockout = True ibm-pwdPolicyStarttime = 20060406200000	pwdMaxAge =43200 ibm-pwdPolicy = True pwdInHistory = 5 ibm-pwdPolicyStarttime = 20060306200000	ibm-pwdPolicy = True pwdMinAge = 43200 pwdInHistory = 3 pwdCheckSyntax = 0 pwdMinLength = 0 pwdExpireWarning = 0 pwdGraceLoginLimit = 0 pwdLockoutDuration = 0 pwdMaxFailure =0 pwdFailureCountInterval=0 passwordMinAlphaChars=0 passwordMinOtherChars=0 passwordMaxRepeatedChars=0 passwordMinDiffChars=0 pwdLockout=False pwdAllowUserChange=True pwdMustChange=True pwdSafeModify=False ibm-pwdPolicyStarttime = 20060506200000	pwdMaxAge = 86400 ibm-pwdPolicy = True pwdMinAge = 21600 pwdInHistory = 5 pwdCheckSyntax = 0 pwdMinLength = 0 pwdExpireWarning = 0 pwdGraceLoginLimit = 0 pwdLockoutDuration = 0 pwdMaxFailure =0 pwdFailureCountInterval=0 passwordMinAlphaChars=0 passwordMinOtherChars=0 passwordMaxRepeatedChars=0 passwordMinDiffChars=0 pwdLockout=True pwdAllowUserChange=True pwdMustChange=True pwdSafeModify=False ibm-pwdPolicyStarttime = 20060406200000
pwdMaxAge = 86400 ibm-pwdPolicy = True pwdMinAge = 21600 pwdMinLength = 8 pwdLockout = True ibm-pwdPolicyStarttime = 20060406200000	pwdMaxAge =43200 ibm-pwdPolicy = True pwdInHistory = 5 ibm-pwdPolicyStarttime = 20060306200000	ibm-pwdPolicy = True pwdMinAge = 0 pwdInHistory = 3 pwdCheckSyntax = 0 pwdMinLength = 10 pwdExpireWarning = 0 pwdGraceLonginLimit = 0 pwdLockoutDuration = 0 pwdMaxFailure =0 pwdFailureCountInterval=0 passwordMinAlphaChars=4 passwordMinOtherChars=4 passwordMaxRepeatedChars=0 passwordMinDiffChars=0 pwdLockout=False pwdAllowUserChange=True pwdMustChange=True pwdSafeModify=False ibm-pwdPolicyStarttime = 20060506200000	pwdMaxAge = 86400 ibm-pwdPolicy = True pwdMinAge = 21600 pwdInHistory = 5 pwdCheckSyntax = 0 pwdMinLength = 8 pwdExpireWarning = 0 pwdGraceLoginLimit = 0 pwdLockoutDuration = 0 pwdMaxFailure =0 pwdFailureCountInterval=0 passwordMinAlphaChars=0 passwordMinOtherChars=0 passwordMaxRepeatedChars=0 passwordMinDiffChars=0 pwdLockout=True pwdAllowUserChange=True pwdMustChange=True pwdSafeModify=False ibm-pwdPolicyStarttime = 20060406200000

Table 15. Determining the effective password policy (continued)

Individual password policy	Group password policy	Global password policy	Effective password policy
passwordMaxConsecutiveRepeatedChars=1 passwordMaxRepeatedChars=0 ibm-pwdPolicy = True	passwordMaxConsecutiveRepeatedChars=1 passwordMaxRepeatedChars=10 ibm-pwdPolicy = True	passwordMaxRepeatedChars=4 ibm-pwdPolicy = True	passwordMaxConsecutiveRepeatedChars=1 passwordMaxRepeatedChars=0 ibm-pwdPolicy = True

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 "Password policy operational attributes" on page 533

Summary of default settings

The following table shows default password policy settings for user passwords.

Table 16. User password policy settings

Web Administration Tool parameter	Default setting
Password policy enabled: ibm-pwdPolicy	false
Password encryption: ibm-slapdPwEncryption:	AES256
Users must specify old password when changing the password: pwdSafeModify	false
User must change password after reset: pwdMustChange	true
Password expiration: pwdMaxAge	0
Number of grace logins after expiration: pwdGraceLoginLimit	0
Account is locked out after a specified number of consecutive failed bind attempts: pwdLockout	false
Number of consecutive failed bind attempts before locking out the account: pwdMaxFailure	0
Minimum time between password changes: pwdMinAge	0
Amount of time before an account lockout expires or lockouts never expire: pwdLockoutDuration	0
Amount of time before an incorrect login expires or incorrect login is cleared only with correct password: pwdFailureCountInterval	0
Minimum number of passwords before reuse: pwdInHistory	0
Check password syntax: pwdCheckSyntax	0
Minimum length: pwdMinLength	0
Minimum number of alphabetic characters: passwordMinAlphaChars	0
Minimum number of numeric and special characters: passwordMinOtherChars	0
Maximum number of repeated characters: passwordMaxRepeatedChars	0
Maximum number of consecutive repeated characters :passwordMaxConsecutiveRepeatedChars	0
Minimum number of characters that must be different from the old password: passwordMinDiffChars	0

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 17. Administration Password Policy Settings

Administration password requirements	Default setting
Password policy enabled: ibm-slapdConfigPwdPolicyOn	false
Account is locked out after a specified number of consecutive failed bind attempts: pwdLockout	true
Maximum number of incorrect logins until password lockout: pwdMaxFailure	10
Amount of time before an account lockout expires or lockouts never expire: pwdLockoutDuration	300
Amount of time before an incorrect login expires or incorrect login is cleared only with correct password: pwdFailureCountInterval	0
Minimum length: pwdMinLength	8
Minimum number of alphabetic characters: passwordMinAlphaChars	2
Minimum number of numeric and special characters: passwordMinOtherChars	2
Maximum number of repeated characters: passwordMaxRepeatedChars	2
Minimum number of characters that must be different from the old password: passwordMinDiffChars	2

Administration password policy is set to false by default. Turning on the administration password policy, enables the other attributes with the default settings.

Password Guidelines

The following section provides details of the supported values of the IBM Tivoli Directory Server password attribute for user entries in 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.

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 member entries are stored in the directory and are considered as User entries.

Details of the supported password values using IBM Tivoli Directory Server 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)

In 6.0 and later releases, 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 6.3 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:

In 6.0 and later releases, 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) - Primary administrator
   - Members of the Local Administrator Group
   - Master ID for Replication (cn=MASTER)
   - LDAP DB2 users for LDAP DB entry and change log databases (LDAPDB2)

   Note: The administrative password policy applies to all these users except the DB2 user.
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, 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 and later versions of 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=sample   -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 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, issue the following command:

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

where *<filename>* contains:

```
dn: cn=pwdPolicy Admin,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
# Value of pwdlockoutduration is in seconds.
-
replace:pwdfailurecountinterval
pwdfailurecountinterval: 0
-
replace:pwdminlength
pwdminlength: 8
-
replace:passwordminalphachars
passwordminalphachars: 2
-
replace:passwordminotherchars
passwordminotherchars: 2
-
replace:passwordmaxrepeatedchars
passwordmaxrepeatedchars: 2
-
replace:passwordmindiffchars
passwordmindiffchars: 2
```

Unlocking administrative accounts

When an administrator unlocks an account by modifying a local admin group member or master server DN password, the account remains locked until the execution of read configuration exop when new password becomes effective. The password modification for a local admin group member does not take effect until a dynamic configuration update request is made. 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
```

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

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

Note: When the administrator's account is locked, the only way to unlock the account is by logging on to the local console.

Setting the global password policy

The global password policy applies to entries stored in the RDBM backend. To set the global password policy, use one of the following procedures.

Using Web Administration

If you have not done so already, click **Server administration** in the Web Administration navigation area and then click **Manage password policies** in the expanded list. On this panel, you can do the following:

- Add a new password policy in the DIT.
- Edit an existing password policy.
- Create a copy of an existing password policy by providing a new name and location of the policy.
- Delete an exiting password policy.

  Note: The global password policy cannot be deleted.
- View the details of a selected password policy.

To add a password policy: To add a new password policy in the DIT, click the **Add** button or select **Add** from the Select Action list and then click **Go** on the Password policies table. This launches the Policy definition wizard in which the user can define a new password policy by providing a unique password policy name and the required attributes and their values.

Attribute selection: Attribute selection, Password policy settings 1, Password policy settings 2, and Password policy settings 3 panels make up the Policy definition wizard. Users can use these panels of the Policy definition wizard to add a new password policy, edit an existing password policy, and create a copy of an existing password policy.

While adding a new password policy or copying an existing password policy, user must provide a unique name for the password policy on the Attribute selection panel. Users can also provide values for the required attributes by selecting the attributes from the Attribute selection table. While editing an existing password policy, users are not allowed to modify the password policy name but can modify the values of the attributes of the selected password policy.

Note: Based on the selection of the attributes from the Attribute selection table on the Attribute selection panel, user may not be required to traverse though all the panels of the Policy definition wizard while adding a new password policy or editing or copying an existing password policy.

On this panel, you can do the following:

- Enter a unique password policy name in the Policy name field. For Add and Copy operations, users must provide a unique password policy name. In case of the Edit operation, the Policy name field is read-only.
- Select the attributes from the table that you want to include in the password policy overriding the values of these attributes that is in the global password policy.

Password policy settings 1: The controls on the Password policy settings 1 panel are displayed based on the selection of the attributes on the Attribute selection panel. On this panel, you can do the following:

1. To enable the password policy, select the **Enabled (ibm-pwdPolicy)** check box. To disable the password policy, clear the **Enabled (ibm-pwdPolicy)** check box. The attribute ibm-pwdPolicy is associated with this control.
2. To allow user to change their password, select the **User can change password (pwdAllowUserChange)** check box.
3. To ensure that the user change the password after it is reset by the administrator, select the **User must change password after reset (pwdMustChange)** check box.

4. To ensure that the user specify the current password while setting a new password, select the **User must specify current password while changing (pwdSafeModify)** check box.
5. To set the start date and time of password policy, enter date and time in the fields under Password policy start time (ibm-pwdPolicyStartTime). To set date, users can use the calendar by clicking the calendar icon.

   Note: Only administrators and the members of local administrative group can set the start date and time of the password policy.
6. In this group, you can set the number of days after which the password expires. If you select **Days**, you must enter the number of days in the field. Otherwise, to ensure that password never expires, select **Password never expires**.
7. In this group, you can set the minimum age of the password. If you select **Days**, you must enter the number of days in the field after which the password can be changed after the last password change. Otherwise, select **Password can be changed anytime**.
8. In this group, you can set the number of days before the password expires at which to display password expiry warning status. If you select **Days before expiration**, you must enter a value in the field for the number of days before the password expires, in order to warn the user about password expiration. Otherwise, select **Never warn**.
9. In the **Logins** field, enter the number of grace login attempts allowed after the password has expired.

After you have finished, do one of the following:

- Click **Back** to navigate to the Attribute selection panel.
- Click **Next** to navigate to continue with configuring of password policy.
- Click **Cancel** to discard all changes and navigate to the Manage password policies panel.
- Click **Finish** to save all the changes and navigate to the Manage password policies panel.

Password policy settings 2: The Password policy settings 2 panel and the controls on the Password policy settings 2 panel are displayed based on the selection of the attributes on the Attribute selection panel. On this panel, you can do the following:

1. Set the maximum number of failed bind attempts allowed by a user before password locks out. If you select **Attempts**, you must enter a value for maximum number of failed bind attempts allowed before password lockout. To specify the maximum number of failed bind attempts allowed before password lockout as unlimited, select **Unlimited**.
2. Set the duration for which the password authentication will remain locked. To specify the duration, you must select and then enter a value for the duration in the field and select the unit from the combo box. Otherwise, select **Infinite**.
3. Set the duration after which failed bind attempts should be flushed. To specify the duration, you must select and then enter a value for the duration in the field and select the unit from the combo box. Otherwise, select **Infinite**.

Password policy settings 3: The Password policy settings 3 panel and the controls on the Password policy settings 3 panel are displayed based on the selection of the attributes on the Attribute selection panel. On this panel, you can do the following:

1. In the **Minimum number of passwords before reuse (pwdInHistory)** field, enter a value for the minimum number of password to be stored before reusing the old password.
2. Select a check password syntax item from the **Check password syntax (pwdCheckSyntax)** list to specify whether the syntax of password should be checked or not. The items available in the **Check password syntax (pwdCheckSyntax)** list are Do not check syntax, Check syntax (two-way encrypted only), and Check syntax.
3. In the **Minimum length (pwdMinLength)** field, enter a value for the minimum length of the password to be used.
4. In the **Minimum number of alphabetic characters (passwordMinAlphaChars)** field, enter a value for the minimum numbers of alphabetic characters that a password should contain.
5. In the **Minimum number of numeric and special characters (passwordMinOtherChars)** field, enter a value for the minimum numbers of numeric and special characters that a password should contain.
6. In the **Maximum number of times a character can be used in password (passwordMaxRepeatedChars)** field, enter a value for the maximum numbers of repeated characters that is allowed in a password.
7. In the **Maximum number of consecutive repeated characters (passwordMaxConsecutiveRepeatedChars)** field, enter a value for the maximum number of consecutive repeated characters that are allowed in a password.
8. In the **Minimum number of characters different from previous password (passwordMinDiffChars**) field, enter a value for the minimum numbers of characters in a new password that should be different from the previous password.

To edit a password policy: To edit an existing password policy, select the required row and click the **Edit** button or select **Edit** from the Select Action list and then click **Go** on the Password policies table. This launches the Policy definition wizard with the selected password policy. User can edit the selected password policy by modifying the required attributes and their values.

To create a copy of an existing password policy: To create a copy of an existing password policy, select the required row and click the **Copy** button or select **Copy** from the Select Action list and then click **Go** on the Password policies table. This launches the Policy definition wizard with the selected password policy. To copy, user must provide a new password policy name and the location of the policy and is allowed to make changes to the attribute values.

To delete a password policy: To delete an existing password policy, select the required row and click the **Delete** button or select **Delete** from the Select Action list and then click **Go** on the Password policies table.

Note: The global password policy cannot be deleted.

Using the command line

To enable the password policy, issue the following command:

```
idsldapmodify –D <adminDN> –w <adminPW> -p <port> -k
dn: cn=pwdpolicy,cn=ibmpolicies
ibm-pwdpolicy:true
ibm-pwdGroupAndIndividualEnabled:true
```

To define group and individual password policies issue the following commands:

```
idsldapadd -D <adminDN> –w <adminPW>
dn:cn=grp1_pwd_policy,cn=ibmpolicies
objectclass: container
objectclass: pwdPolicy
objectclass: ibm-pwdPolicyExt
objectclass: top
cn:grp_pwd_policy
pwdAttribute: userPassword
pwdGraceLoginLimit: 1
pwdLockoutDuration: 30
pwdMaxFailure: 2
pwdFailureCountInterval: 5
pwdMaxAge: 999
pwdExpireWarning: 0
pwdMinLength: 8
pwdLockout: true
pwdAllowUserChange: true
pwdMustChange: false
ibm-pwdpolicy:true
```

```
idsldapadd -D <adminDN> –w <adminPW>
dn:cn=individual1_pwd_policy,cn=ibmpolicies
objectclass: container
objectclass: pwdPolicy
objectclass: ibm-pwdPolicyExt
objectclass: top
cn:grp_pwd_policy
pwdAttribute: userPassword
pwdGraceLoginLimit: 3
pwdLockoutDuration: 50
pwdMaxFailure: 3
pwdFailureCountInterval: 7
pwdMaxAge: 500
pwdExpireWarning: 0
pwdMinLength: 5
pwdLockout: true
pwdAllowUserChange: true
pwdMustChange: false
ibm-pwdpolicy:true
```

To associate the group and individual password policies with a group or a user, issue the following commands. For instance, to associate a group password policy with a group:

```
idsldapmodify -D <adminDN> -w <adminPW> -k
dn:cn=group1,o=sample
changetype:modify
add:ibm-pwdGroupPolicyDN
ibm-pwdGroupPolicyDN:cn=grp1_pwd_policy,cn=ibmpolicies
```

To associate an individual password policy with a user:

```
idsldapmodify -D <adminDN> -w <adminPW> -k
dn:cn=user1 ,o=sample
changetype:modify
add:ibm-pwdIndividualPolicyDN
ibm-pwdIndividualPolicyDN:cn= Individual1 _pwd_policy,cn=ibmpolicies
```

Changing password when pwdsafemodify is set

When using the Tivoli Directory Server LDAP client, you can use the 'ldapchangepwd' utility to modify a user's password. However, if you are using non-Tivoli Directory Server LDAP client then you can change the userpassword as shown below.

Consider an example where you have a user 'cn=user,o=sample' with the password as 'passw001rd' and you need to update that password to 'passw007rd'. To do this, issue the following command:

```
ldapmodify -p <port> -D <bindDN> -w <bindPassword> -i <input file>

dn: cn=user,o=sample
changetype: modify
delete: userpassword
userpassword: <old password value>
-
add: userpassword
userpassword: <new password value>
```

Setting Kerberos

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 6.3 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 189 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 IBM Tivoli Directory Server 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.

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
krbReamlName-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=Reginal 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.

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, and then select the **DIGEST-MD5** tab. The Digest-MD5 tab is displayed only if any one of the two conditions is satisfied:

- The root DSE search returns the ibm-supportedCapabilities OID 1.3.18.0.2.32.69 for Digest-MD5.
- The root DSE search returns DIGEST-MD5 as value of the supportedsaslmechanisms attribute.

The values of the controls in the Digest-MD5 tab are updated with the Digest-MD5 parameters from the entry "cn=Digest, cn=Configuration" in the configuration file when the tab is loaded.

1. Select the **Enable Digest-MD5** check box to enable the Digest-MD5 mechanism.

   Note: When the **Enable Digest-MD5** check box is selected, other controls related to Digest-MD5 parameters on this tab are enabled and modifications to these controls are allowed.

2. 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.

3. 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.

4. 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.

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 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: <uuid>
ibm-slapdDigestAdminUser: <Adminuser>
ibm-slapdDigestEnabled: true
objectclass:top
objectclass: ibm-slapdConfigEntry
objectclass: ibm-slapdDigest
```

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-slapdDigestRealm
ibm-slapdDigestRealm: <newrealmname>
-
replace: ibm-slapdDigestAttr
ibm-slapdDigestAttr: <newattribute>
-
replace: ibm-slapdDigestAdminUser
ibm-slapdDigestAdminUser: <newAdminuser>
```

Given below is an example of how a user can bind to the server using the Digest MD5 mechanism:

```
idsldapsearch -h <ldaphost> -p ldapport -U <username> -w <password> -m DIGEST-MD5
 -G <realm> -b o=sample cn=gw*
```

Note: To perform a Digest MD5 bind it is necessary to specify the -h <hostname> option. The <hostname> parameter must be the IP address or FQDN (fully qualified domain name) of that Tivoli Directory Server machine, even if the bind is performed from local machine. Specifying localhost or loopback IP address as value of -h may lead to error.

Pass-through authentication

Pass-through authentication (PTA) is a mechanism using which if a client attempts to bind to a directory server and if the user credential is not available locally, then the server attempts to verify the credential from another external directory server or a pass-through server on behalf of the client. The credential here refers to the *userpassword* attribute in Tivoli Directory Server. To gain a better understanding of the pass-through authentication mechanism, consider an example given below:

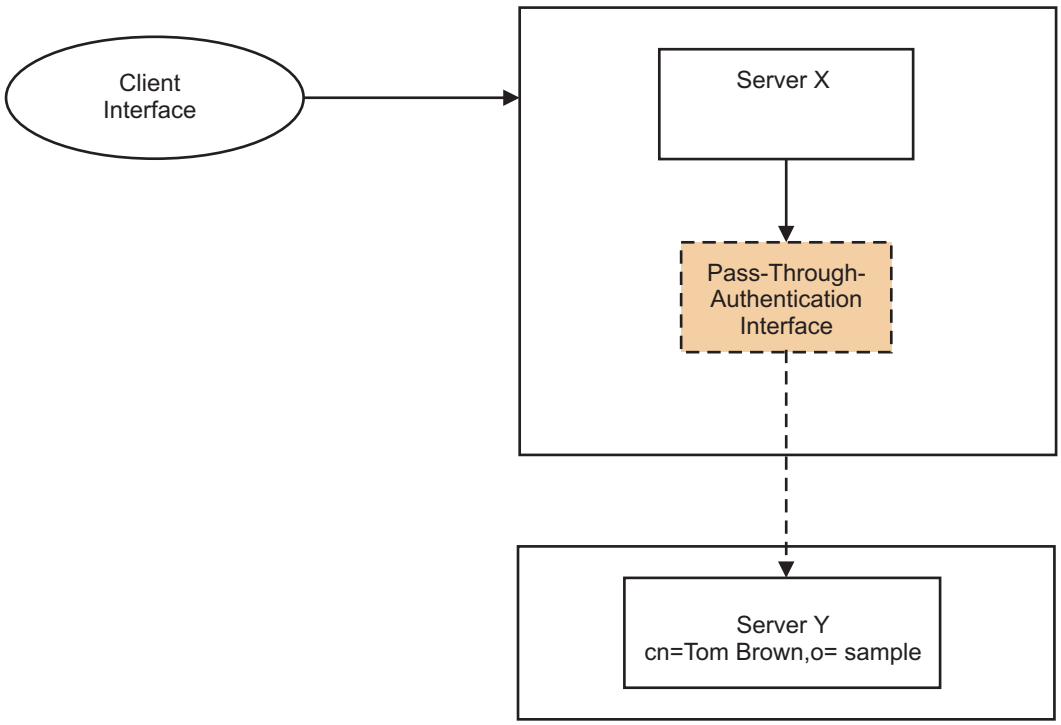


Figure 1. Pass-through authentication

Note: The pass-through server that holds the user credentials can be Active Directory or an LDAP V3 compliant directory (including Tivoli Directory Server).

To illustrate use case of pass-through authentication, let us consider two servers say server X and server Y and a user entry cn=Tom Brown,o=sample stored on server Y. Now, if the user Tom Brown attempts to access server X to perform any operation it has to first bind to server X with its credential for authentication. Since the credential is not present on server X the user will be unable to bind to the server. However, using the pass-through authentication mechanism, server X can verify the credential by contacting server Y. After the credential is validated using server Y, server X presumes that the user is authenticated and hence returns success for the bind operation.

Alternatively, if a user is present on server X while its credential is available on server Y, again server X will contact server Y to verify the credential.

In the above cases it is assumed that the DN's on server Y and server X are identical. However, this may not be true always as the directory structure layout may differ on both the servers. This means that DN “cn=Tom Brown,o=sample” on server X may map to some other DN on server Y. In such situations it is possible that the entries on server X and server Y have some attribute whose value is unique for every entry, say for instance uid. Therefore, an attribute from Tivoli Directory Server can be mapped to another attribute in the pass-through server. This information can then be used to query the pass-through server to fetch the required DN. A bind operation will then be performed for this DN to identify if the userpassword is correct.

Note:

- Any configuration changes done for pass through authentication are not dynamic in nature and hence require a restart.
- It is important to note that all the entries mentioned in the scenarios below must be within a configured subtree.

Scenarios pertaining to pass-through authentication

Let us consider a few scenarios pertaining to pass-through authentication:

Scenario 1: Attribute mapping is configured and entry is present locally

This is a scenario where an attribute in Tivoli Directory Server will match some other attribute in pass-through server. It is not necessary that the name of an attribute is identical in both directories. For instance, *uid=Tom456* in Tivoli Directory Server may map to *userPrincipalName=Tom456* in the pass-through server.

In this scenario, all the entries in Tivoli Directory Server can be directly mapped to the entries in the pass-through server. Now, a search can be performed on the pass-through server to get the actual DN which will then be used to perform a bind operation to verify the user credential.

A sample entry in the configuration would look like:

```
ibm-slapdPtaAttrMapping : uid $ userPrincipalName
uid in TDS is mapped to userPrincipalName in pass-through server.
```

Let us assume that the following entry exists on Tivoli Directory Server:

```
dn: cn=Tom Brown,o=sample
sn: tests
uid: Tom456
objectclass: organizationalPerson
objectclass: person
objectclass: top
objectclass: inetOrgPerson
```

Now, in case of a bind request with a DN “cn=Tom Brown,o=sample”, the pass-through server will be searched using “userPrincipalName=Tom456” search filter. If only one entry is returned then the DN of that entry is used and a bind operation is performed to verify the password. However, if uid is multi-valued in the “cn=Tom Brown,o=sample” entry, the bind operation will fail.

Let us suppose another situation where there is no unique attribute that can be mapped between directories. In such a situation you must introduce an auxiliary class and attach it to the entry where the mapping is required. For instance:

```
dn: cn=Tom Brown,o=sample
sn: test
uid: Tom456
objectclass: organizationalPerson
objectclass: person
objectclass: top
objectclass: inetOrgPerson
objectclass: my-aux-class
uniqueValue: my_value
```

You can create a new auxiliary objectclass (my-aux-class) and associated attribute (uniqueValue) or use any existing objectclass and attribute. Finally, you set *ibm-slapdPtaAttrMapping* as:

```
ibm-slapdPtaAttrMapping : uniqueValue $ userPrincipalName
```

Note: If the mapping attribute *ibm-slapdPtaAttrMapping* is not set to a unique attribute then it is possible that the pass-through server will return more than one entry or a false entry and the PTA interface will return a bind failure and log a message.

Scenario 2: Attribute mapping is configured, entry is present locally, and password migration is enabled

This is similar to scenario 1 except that after the result is sent to the client, the PTA interface will store the userpassword provided by the user during the bind operation in its local entry. Here, password will be stored in the Tivoli Directory Server local directory after the first successful bind and will be present in the directory even after the server is down. Subsequent bind requests from this user will be served completely by local Tivoli Directory Server and will not reach the pass-through server. The *userpassword* will be stored using the encryption scheme configured locally and adhere to the local password policy settings.

Note: The password will also be replicated as per the replication configuration in Tivoli Directory Server.

In this kind of scenario it is important to maintain consistency between the passwords of the pass-through server and the local Tivoli Directory Server. Inconsistencies between the passwords available at the pass-through server and the local Tivoli Directory Server can pose a potential security threat. A system administrator needs to ensure that the integrity of passwords at both the directories is maintained.

Scenario 3: The attribute mapping is not configured and the entry is not present locally.

In this kind of scenario after the bind request fails to locate the entry locally, the PTA interface will check if any pass-through server is configured to service the bind DN. If any pass-through server is configured, then using the bind DN and password supplied by the user, the bind request is send to the pass-through server. If the bind succeeds, the server returns success, otherwise it returns LDAP_INVALID_CREDENTIALS. In this scenario since the entry is not present locally, enabling password migration will not have any effect.

Scenario 4: Auxiliary object class is set for attribute-linking

In this scenario presume that there are two entries in Tivoli Directory Server and one entry in the pass-through server for the user Tom as shown below.

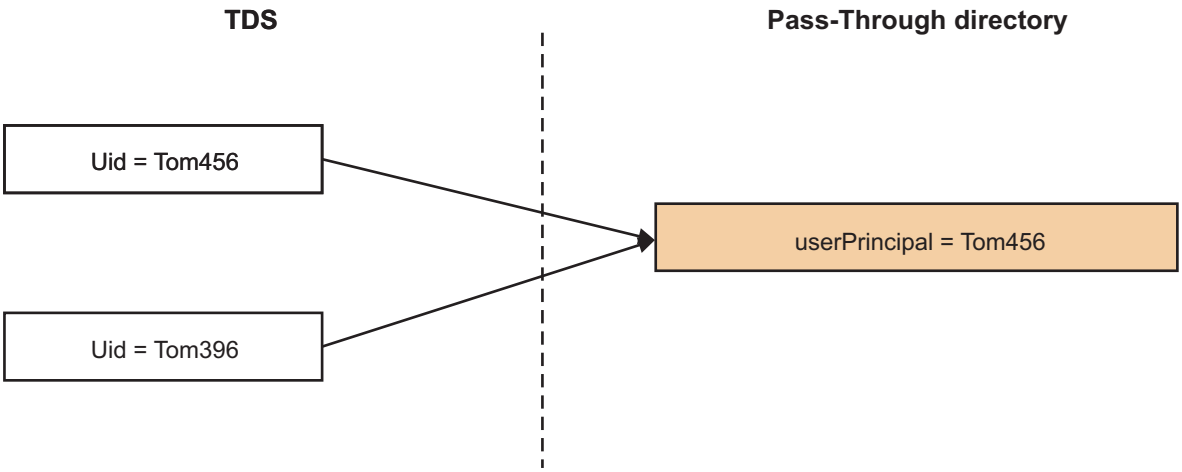


Figure 2. Attribute linking

Now, *uid=Tom456* can be easily mapped to *userPrincipal=Tom456*. However, there is no mapping between the second entry *uid=Tom396* and *userPrincipal=Tom456* since both the values differ even though they actually pertain to the same person. Therefore, if there is a bind request for *uid=Tom396* which has its credentials on the pass-through server, the bind will fail. To solve this problem, you must add an auxiliary class *ibm-ptaReferral*. This will hold two MUST attributes *ibm-PtaLinkAttribute* and *ibm-PtaLinkValue*. This needs to be added to the entry that has no mapping in the pass-through server. Now, whenever there is a bind request for *uid=Tom396*, the PTA interface will first look if the *ibm-ptaReferral* objectclass is present. If it is present then it will fetch the details for the MUST attribute and frame the required search query. The entry will look like:

```
dn:  cn=Tom396,o=sample
objectclass: inetOrgPerson
objectclass: organizationalPerson
objectclass: person
objectclass: top
uid:Tom396
sn: test
objectclass: ibm-ptaReferral
ibm-ptaLinkAttribute:  userPrincipalName
ibm-ptaLinkValue:  Tom456
```

Another case to be considered is when there is no mapping between Tivoli Directory Server and the pass-through server but the administrator is aware of the DN that directly maps between both the directories. In such cases, *ibm-PtaLinkAttribute* must be set to "_DN_" and *ibm-PtaLinkValue* must be set to the actual DN of mapped entry. The entry will look like:

```
dn:  cn=Tom396,o=sample
objectclass: inetOrgPerson
objectclass: organizationalPerson
objectclass: person
objectclass: top
uid:Tom396
sn: test
objectclass: ibm-ptaReferral
ibm-ptaLinkAttribute:  _DN_
ibm-ptaLinkValue:  cn=Tom1000,o=sample
```

By setting these values in the entry, the PTA interface takes the specified DN value and binds using the user supplied credentials. The result will be returned accordingly.

It is important to note that when computing the DN to be passed to the pass-through server, if it is found that an entry is set with *ibm-ptaReferral* auxiliary class, then the attribute mapping configured for the entry will be ignored.

Note: In case you do not want pass-through authentication to be performed for a specific entry, you must set the *ibm-ptaLinkAttribute* to _DISABLE_.

Scenario 5: Configuring pass-through authentication to Active Directory Global Catalog

If a user want to authenticate to an external server, then the user is required to configure attribute mapping for pass-through authentication. This requires providing of information such as the search base (ibm-slapdPtaSearchBase), server URL (ibm-slapdPtaURL), and bind information, etc of the individual server against which the user intend to authenticate. However, if the user want to authenticate to an Active Directory forest instead of an individual external server, then user must specify a search base of **" "**. To search Active Directory Global Catalog, the settings for pass-through authentication in the the configuration file should be of the following format:

```
dn: cn=Passthrough Server1, cn=Passthrough Authentication, cn=Configuration
cn: Passthrough Server1
ibm-slapdPtaAttrMapping: cn $ uid
ibm-slapdPtaBindDN: <valid_DN>
ibm-slapdPtabindPW: <DN_password>
ibm-slapdPtaSearchBase:
ibm-slapdPtaSubtree: <Local subtree>
ibm-slapdPtaURL: ldap://<host>:3268
objectclass: top
objectclass: ibm-slapdConfigEntry
objectclass: ibm-slapdPta
objectclass: ibm-slapdPtaExt
```

Note: The value of the attribute ibm-slapdPtaSearchBase must not be set (that is, should be left empty). Tivoli Directory Server will make a search to Active Directory with search base as " " to make it a Global Catalog search. The search should be routed through Global Catalog port, 3268.

To know more about Active Directory Global Catalog, refer to Microsoft TechNet site:Global Catalog and LDAP Searches.

Using Web Administration Tool

If you have not done so already, expand the **Manage security properties** category under **Server administration** in the navigation area of the Web Administration Tool and click the **Pass-through authentication** tab.

On this panel, you can:

- Enable or disable pass-through authentication by selecting or clearing the **Enable pass-through authentication** check box.
- Configure a pass-through entry for a subtree for pass-through authentication. Clicking **Add** displays the Configure subtree for pass-through authentication wizard that can be used for configuring a pass-through entry for a subtree for pass-through authentication.
- Edit an existing pass-through entry of a subtree for pass-through authentication. Clicking **Edit** displays the Configure subtree for pass-through authentication wizard that can be used for modifying an existing pass-through entry of a subtree for pass-through authentication.

- Delete an existing pass-through entry of a subtree configured for pass-through authentication. For this, select a subtree from the Subtrees configured for pass-through authentication table and click the **Delete** button.
- View pass-through entry details of a configured subtree for pass-through authentication. For this, select a subtree from the Subtrees configured for pass-through authentication table, select View from the Select Action list, and click **Go**.
- After you are finished, do one of the following:
  - Click **OK** to save changes and navigate to the “Introduction” panel.
  - Click **Apply** to save changes and to remain on this panel.
  - Click **Cancel** to discard changes made and navigate to the “Introduction” panel.

To configure a pass-through entry for a subtree for pass-through authentication follow the steps given below:

1. In the Pass-through authentication panel, click **Add**.
2. Next, on the Subtree settings panel you can do the following:
   - Enter a subtree DN in the field and click the **Add** button to add it to the list for storing subtree DN.
   - Enter multiple subtree DNs by clicking the **Browse** button and then selecting the required rows from the Browse entries panel.
   - Remove a subtree DN from the list for storing subtree DN by selecting the subtree DN and clicking the **Remove** button.
   - Specify the host name of the pass-through server in the **Host name** field. This is a required field.
   - Specify the port number of the pass-through server in the **Port** field. This is a required field.
   - Enable SSL encryption on the pass-through server by selecting the **Enable SSL encryption** check box.
   - Specify whether to save the user password on the local directory for all successful bind request processed through the pass-through server by selecting a value from the **Migrate userpassword to this directory server** combo box. The default value of this control is “False”.
   - Specify the number of connections that is required for each pass-through server entry in the **Number of connections to the pass-through server to maintain for Pass-through authentication** field.
   - Specify a timeout value in the **Pass-through authentication timeout** field. The pass-through authentication interface will wait for result from socket till the timeout period before it returns the client request.

     Note:
     - The attribute "ibm-slapdPtaResultTimeout" in the "cn=< pass-through server >, cn=Passthrough Authentication, cn=Configuration" entry is associated with this control.
     - The timeout value is specified in milliseconds. The upper limit for this field is 60000 millisec (60 sec or 1 minute).
   - Click **Next**.

To configure attribute mapping, do the following:

1. Select the **Enable attribute mapping** check box to enable attribute mapping. Selecting the **Enable attribute mapping** check box also enables other controls on the Attribute mapping panel.
2. In the **Bind DN for pass-through server** field, enter a bind DN for binding to the pass-through server.
3. In the **Bind password for pass-through server** field, enter a bind password for binding to the pass-through server.
4. In the **Search base DN** field, enter the search base DN of pass-through server where the entry will be searched, or click the **Browse** button to display Browse entries panel from which the user can select the existing DN from the pass-through server.
5. From the **Attribute for this directory server** combo box, select an attribute that should be mapped to an attribute in pass-through server.
6. From the **Attribute for pass-through directory server** combo box, select an attribute that should be mapped to the Tivoli Directory Server attribute.
7. When you are finished, do one of the following:
   - Click **Back** to navigate to the Subtree settings panel.
   - Click **Finish** to save the changes and to navigate to the Pass-through authentication.
   - Click **Cancel** to discard the changes and to navigate to the Pass-through authentication.

Using command line

To enable PTA using the command line you must modify the configuration file of the directory server. Issue the following command to enable PTA:

```
ldapmodify -h <hosname> -p <port> -D <adminDN> -w <adminpwd> -f <ldif file>

where the ldif file contains

dn: cn=Configuration
ibm-slapdPtaEnabled: true

dn: cn=Passthrough Server1, cn=Passthrough Authentication, cn=Configuration
changetype: add
cn: passthrough Server1
ibm-slapdPtaURL: ldap://<hostname>:<port>
ibm-slapdPtaSubtree: o=sample
ibm-slapdPtaMigratePwd: false
ibm-slapdPtaConnectionPoolSize: 6
objectclass: top
objectclass: ibm-slapdConfigEntry
objectclass: ibm-slapdPta
```

The above command enables PTA, configures a subtree for pass-through authentication, specifies the hostname and port number for a pass through server, and specifies the number of connections that is required for each pass-through server entry. Also, the above command specifies that the user password must not be saved to the local directory for all successful bind requests.

To enable PTA and configure attribute mapping, issue the following command:

```
ldapmodify -h <hostname> -p <port> -D <adminDN> -w <adminpwd> -f <ldif file>

where the ldif file contains

dn: cn=Configuration
ibm-slapdPtaEnabled: true
```

```
dn: cn=Passthrough Server1, cn=Passthrough Authentication, cn=Configuration
changetype: add
cn: passthrough Server1
ibm-slapdPtaURL: ldap://<hostname>:<port>
ibm-slapdPtaSubtree: o=sample
ibm-slapdPtaMigratePwd: true
ibm-slapdPtaAttrMapping: sn $ uid
ibm-slapdPtaSearchBase: ou=austin,o=sample
ibm-slapdPtaBindDN: <bind DN>
ibm-slapdPtabindPW: <bind password>
objectclass: top
objectclass: ibm-slapdConfigEntry
objectclass: ibm-slapdPta
objectclass: ibm-slapdPtaExt
```

In the above example attribute mapping is configured and password migration is also enabled. Here, the attribute 'sn' in the directory server is mapped to the attribute 'uid' in the pass-through server.

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: IBM Tivoli Directory Server's replication supplier DNs can match each other.

A primary administrator must ensure that the archive log attributes are set in both cn=Audit,cn=Log Management,cn=Configuration and cn=Admin Audit,cn=Log Management ,cn=Configuration entries. By setting the archive attributes in the specific entries the primary administrator eliminates the risk of a local administrator changing the default archive log settings that will in turn impact the archiving of audit logs. Without setting the archive log attributes in the audit entries, a local administrative group member may update the default log settings and impact the audit logs. Following are the archive log attributes:

- ibm-slapdLogMaxArchives
- ibm-slapdLogSizeThreshold
- ibm-slapdLogArchivePath

Administrative Roles

While configuring an administrative group member, the primary administrator has to explicitly assign an administrative role to the member. The roles that can be assigned to an administrative member are given below:

- Audit administrator (AuditAdmin) – Members of the administrative group who are assigned the Audit Administrator role have unrestricted access to
  - Audit log

  - Admin Audit log
  - All other server logs
  - Audit log settings (cn=Audit, cn=Log Management, cn=Configuration)
  - Admin Audit log settings (cn=Admin Audit, cn=Log Management, cn=Configuration)
  - Default log management settings (cn=Default, cn=Log Management, cn=Configuration)
- Directory Data Administrator (DirDataAdmin) – Members of the administrative group who are assigned this role will gain unrestricted access to all the entries in the RDBM back-end. However, for setting the password attribute of RDBM entries, members will still have to follow the usual password policy rules that are in effect.
- No administrator (NoAdmin) – If the primary administrator assigns No Administrator role to the configuration file users, then the users will cease to have any administrative privileges. By defining this role the primary administrator can revoke all the administrative privileges of an administrative group member
- Password administrator (PasswordAdmin) – Members of the administrative group who are assigned the Password Administrator role are authorized to unlock other user's accounts or change passwords of users in RDBM back-end. However, they are not authorized to change passwords of Global Administrative Group Member accounts. Also, they are not restrained by password policy constraints that are set on the server. They can also add and delete the userpassword field of entries in RDBM back-end but are not allowed to make changes to users defined in the configuration file. The password changes made by users who are assigned this role are not affected by ACLs. However, when users change their own password, the usual user password policy rules will apply.
- Replication administrator (ReplicationAdmin) – Members of the administrative group who are assigned the Replication Administrator role are authorized to update replication topology objects. The changes made by members with this role are not affected by ACLs or any other configuration file settings.
- Schema administrator (SchemaAdmin) – Members of the administrative group who are assigned the Schema Administrator role have unrestricted access to schema back-end only.
- Server configuration group member (ServerConfigGroupMember) – Members of the administrator group who are assigned the Server Configuration Group Member role have restricted update access to the configuration back-end. This means that Server configuration group members will have restricted update access to entries under cn=Configuration. Users with this role will be unable to perform certain tasks, particularly those related to other local and primary administrators or tasks related to security. For instance, they will be unable to change primary administrator and Admin Group credentials and add or remove members from the administrative group. Also, they will be unable to modify the DN, password, Kerberos ID, or Digest-MD5 ID of any administrative group member entry under cn=AdminGroup, cn=Configuration. They are also not authorized to modify their own DN,Kerberos ID, or Digest-MD5 ID. They are not authorized to add, delete or modify the administrative roles assigned to any of the administrative group members. However, users will be able to modify their own password. In addition, users with this role will be unable to view the password of any other administrative group member or the primary administrator and they are not authorized to add, delete, or modify the audit log setting and admin audit log settings (the entire cn=Audit, cn=Log Management,

cn=Configuration and cn=Admin Audit, cn=Log Management, cn=Configuration entries) or clear the audit log and admin audit log. However, they are allowed to modify default log settings (the cn=Default, cn=Log Management, cn=Configuration entry) and clear all other server logs. Also, users with this role will be unable to add or delete the cn=Kerberos, cn=Configuration or cn=Digest, cn=Configuration entries. However, they are allowed to search all attributes under these entries. The users will be able to modify all attributes under these entries except the Kerberos and Digest-MD5 root administrator bind attributes. They will be unable 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.

- Server start/stop administrator (ServerStartStopAdmin) – Members of the administrative group who are assigned the Server Start/Stop Administrator role are authorized to start or stop the server and the administrator daemon.

Note: See "Global administration group" on page 322 for information on how administrative rights are delegated for the database backend in a distributed directory environment.

The following table gives cross references of various extended operations that administrative group members are allowed to issue.

Table 18. Administrative roles authorized to issue various extended operations

Extended Operations	Audit Admin	Directory Data Admin	Replication Admin	Schema Admin	Server Configuration Group Member	Server Start/Stop Admin	Password Admin	No Admin
Start TLS - Request to start Transport Layer Security. OID = 1.3.6.1.4.1.1466.20037	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Event Registration - Request registration for events in SecureWay™ V3.2 Event support. OID = 1.3.18.0.2.12.1	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Event Unregister - Request Unregister for events that were registered for using an Event Registration Request. OID = 1.3.18.0.2.12.3	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Begin Transaction - Begin a Transactional context for SecureWay V3.2. OID = 1.3.18.0.2.12.5	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
End Transaction - End Transactional context (commit/rollback) for SecureWay V3.2. OID = 1.3.18.0.2.12.6	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Enable and Disable Tracing Dynamically. OID = 1.3.18.0.2.32.14	No	No	No	No	No	No	No	No
Cascading Control Replication - 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. OID = 1.3.18.0.2.12.15	No	Yes	Yes	No	No	No	No	No

Table 18. Administrative roles authorized to issue various extended operations (continued)

Extended Operations	Audit Admin	Directory Data Admin	Replication Admin	Schema Admin	Server Configuration Group Member	Server Start/Stop Admin	Password Admin	No Admin
Control Replication - 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. OID = 1.3.18.0.2.12.16	No	Yes	Yes	No	No	No	No	No
Control Replication Queue - 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. OID = 1.3.18.0.2.12.17	No	Yes	Yes	No	No	No	No	No
Quiesce or Unquiesce Server - 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. OID = 1.3.18.0.2.12.19	No	Yes	Yes	No	No	No	No	No
Clear Log Request – Request to Clear log file. OID = 1.3.18.0.2.12.20	Yes	No	No	No	Yes	No	No	No
Get Lines Request - Request to get lines from a log file. OID = 1.3.18.0.2.12.22	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
Number of Lines Request - Request number of lines in a log file. OID = 1.3.18.0.2.12.24	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
Start, Stop Server Request - Request to start, stop or restart an LDAP server. OID = 1.3.18.0.2.12.26	No	No	No	No	No	Yes	No	No
Update Configuration Request - Request to update server configuration for IBM Tivoli Directory Server. OID = 1.3.18.0.2.12.28	Yes	No	Yes	No	Yes	No	No	No
DN Normalization Request - Request to normalize a DN or a sequence of DNs. OID = 1.3.18.0.2.12.30	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Kill Connection Request - 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. OID = 1.3.18.0.2.12.35	No	Yes	No	No	No	No	No	No

Table 18. Administrative roles authorized to issue various extended operations (continued)

Extended Operations	Audit Admin	Directory Data Admin	Replication Admin	Schema Admin	Server Configuration Group Member	Server Start/Stop Admin	Password Admin	No Admin
User Type Request - Request to get the User Type of the bound user. OID = 1.3.18.0.2.12.37	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Control Server Tracing - Activate or deactivate tracing in IBM Tivoli Directory Server. OID = 1.3.18.0.2.12.40	No	Yes	No	No	No	No	No	No
Group Evaluation – This operation is used in a distributed directory environment to determine all groups that a particular DN is a member of. OID = 1.3.18.0.2.12.50	No	Yes	No	No	No	No	No	No
Topology Replication – This operation is used to replicate the objects that define the topology of a particular replication context, such as the replication agreements for that context. Any user with update rights to the Replication Group Entry of the context is allowed to issue this extended operation. OID = 1.3.18.0.2.12.54	No	Yes	Yes	No	No	No	No	No
Event Update – Request to reinitialize the event notification configuration (this operation can only be initiated by the server, not any user). OID = 1.3.18.0.2.12.31	No	No	No	No	No	No	No	No
Log Access Update – Request to reinitialize the log access plugin configuration (this operation can only be initiated by the server, not any user). OID = 1.3.18.0.2.12.32	No	No	No	No	No	No	No	No
Unique Attributes – Request to get the duplicate values for an attribute. OID = 1.3.18.0.2.12.44	No	Yes	No	No	No	No	No	No
Account Status – This operation is used to determine if an account is locked by password policy. OID = 1.3.18.0.2.12.58	No	Yes	No	No	No	No	No	No
Locate Entry – This operation is used locate details of a given set of DN(s). OID = 1.3.18.0.2.12.71	No	Yes	No	No	No	No	No	No
Proxy Resume Role – Request that a backend server's role is resumed. OID = 1.3.18.0.2.12.65	No	Yes	No	No	No	No	No	No
Get Attributes Type – Request attributes types. OID = 1.3.18.0.2.12.46	No	Yes	No	Yes	No	No	No	No

Table 18. Administrative roles authorized to issue various extended operations (continued)

Extended Operations	Audit Admin	Directory Data Admin	Replication Admin	Schema Admin	Server Configuration Group Member	Server Start/Stop Admin	Password Admin	No Admin
ServerBackupRestore- Requests that the admin server either perform a backup of a directory server's data and configuration or restore a directory server's data and configuration from an existing backup. OID = 1.3.18.0.2.12.81	No	Yes	No	Yes	Yes	Yes	No	No

The following table gives cross references of various objects that different administrative group members are allowed to access.

Table 19. Permissions assigned to Administrative roles for accessing various objects

	Audit Settings / Audit logs		RDBM Backend		Replication Objects		Schema Backend		Configuration Backend		Proxy Backend	Server Start/ Stop
	Read	Write	Read	Write	Read	Write	Read	Write	Read	Write		
Audit Administrator	Yes	Yes	No**	No	No**	No	Yes	No	Yes	No	Note1	No
Directory Data Administrator	No	No	Yes	Yes	Yes	Yes	Yes	No	Yes	No	Note1	No
Replication Administrator	No	No	No**	No**	Yes	Yes	Yes	No	Yes	No	Note1	No
Schema Administrator	No	No	No**	No	No**	No	Yes	Yes	Yes	No	Note1	No
Server Configuration Group Member	Yes	No	No**	No	No**	No	Yes	No	Yes	Yes*	Note1	No
Server Start/Stop Administrator	No	No	No**	No	No**	No	Yes	No	Yes	No	Note1	Yes
Password Administrator	No	No	No**	Yes**	No**	No	Yes	No	Yes	No	Note1	No
No Administrator	No	No	No**	No	No**	No	Yes	No	Yes	No	Note1	No

- * - Server Configuration Group Member will have restricted update access to configuration backend.
- ** - For access to these objects the administrative roles give no special authority, but the user may still have access through normal ACL evaluation.
- Note1 - Proxy will treat the admin group members having any administrative role as anonymous and will accordingly apply access rules.

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 104 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, expand the **Server administration** category in the navigation area and click **Manage administrative group**. Next, 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 181 for information about administration password security restrictions.
3. Enter the member's password again to confirm it.
4. Optionally, enter the member's **Digest-MD5 user name** .
5. 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.
6. Under the Administrative role section, select the **Define roles for admin group member** check box.
7. Select the available administrative roles from the **Available administrative role box** and click **Add**.
8. 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-slapdAdminRole: <role value>
ibm-slapdAdminRole: <role value2>
#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.
- If multiple instances of ibm-slapdAdminRole attribute are specified with different role values, and one of these role values is NoAdmin, then all other role values will be ignored and an administrative group member having NoAdmin role will be added.

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, expand the **Server administration** category in the navigation area and click **Manage administrative group**. 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>

replace: ibm-slapdAdminRole
ibm-slapdAdminRole: <role value>
```

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.

Note: Provide the DN of the entry holding the admin group member and not the bind DN of the admin group member.

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

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

Chapter 13. 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.

Note:

- In a proxy environment, the proxy server will not return referrals. Referral objects will be returned as normal directory entries, such that a client will not chase referrals.
- Referrals are not recommended in a proxy environment.

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 6.0 and above versions are enabled to support IPv6 and IPv4 formats. See Appendix E, “IPv6 support,” on page 517 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.

© Copyright IBM Corp. 2002, 2010

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.

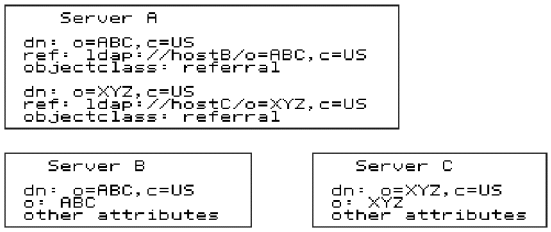


Figure 3. Example of using the referral attribute

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 by specifying the **-M** option. For more information about the command line utilities see the *IBM Tivoli Directory Server Version 6.3 Command Reference*

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=sample
objectclass:  referral
ref:          ldap://9.130.25.51:389/o=sample
```

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 Tivoli Directory Server 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 27 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.

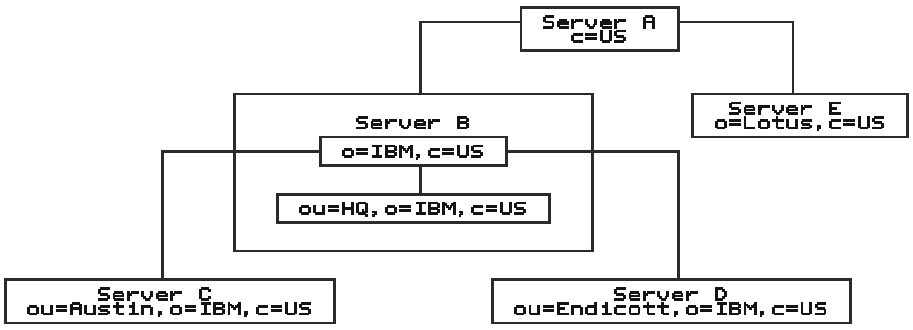


Figure 4. Setting up the servers

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.

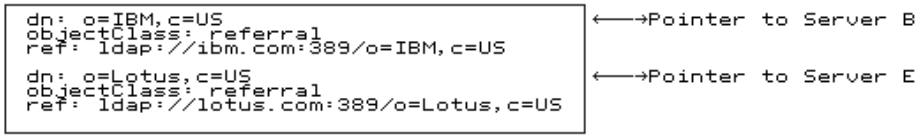


Figure 5. Server A database (LDIF input)

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.

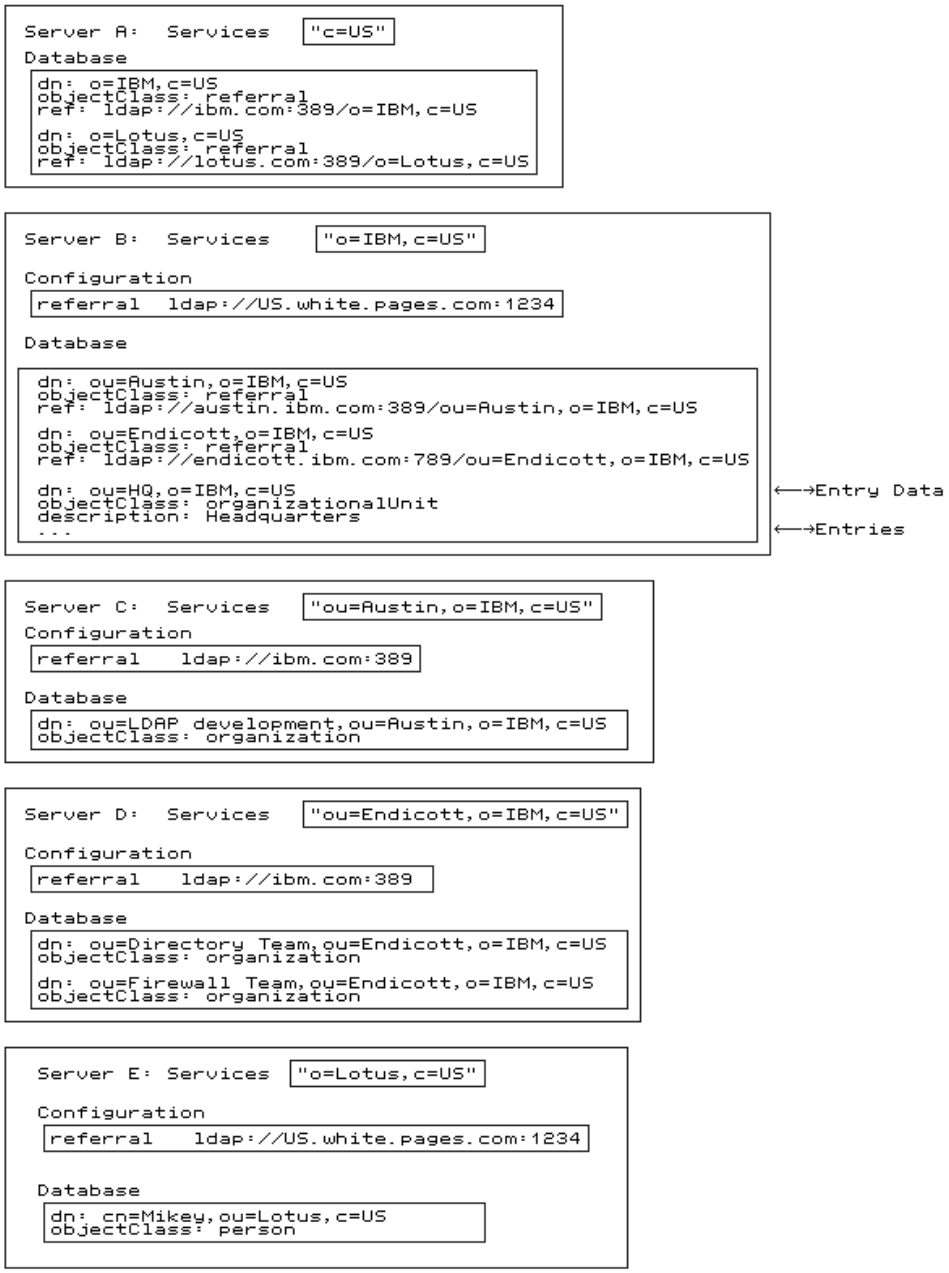


Figure 6. Referral example summary

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. Add a referral entry by selecting the referral object class from the Structural object classes list on the Select object class panel. See “Adding an entry” on page 402 for additional information.
3. On the **Required attributes** tab, click **Manage Referral**.
4. On the Manage referral panel, click **Add** to display the Add referral panel.

   Note: For Admin referrals, the fields related to attributes and filter are not displayed. Admin referrals can be created by adding a referral from the Manage Server Properties panel in the Server Administration category.
5. From the **Server hostname:port** drop down list, select an LDAP server and port or enter a host name and port number of a server in the field in the hostname:port format.
6. Select **Use SSL**, if the referral is to a secure (SSL) server.
7. Enter the base DN in the directory information tree in the target server. For example ou=austin,o=sample.
8. 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**.
9. 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.
   - Select **None** to specify no scope.
10. Specify a search filter. See “Search filters” on page 414 for more information.
11. Click **OK**.
12. Repeat these steps for additional referrals.
13. When you are finished, click **Next** on the Required attributes tab.
14. At the **Optional attributes** tab enter the values as appropriate for the other attributes.
15. Click **Finish** to create the entry.

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=sample** , 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=sample?cn,sn,description?base?objectclass=*
ibm-slapdreferral: ldaps://server2.mycity.mycompany.com:389/
    ou=austin,o=sample?cn,sn,description?base?objectclass=*
objectclass: ibm-slapdReferral
objectclass: ibm-slapdConfigEntry
objectclass: top
```

SeeAppendix E, “IPv6 support,” on page 517 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. On the **Required attributes** tab of the Add an entry panel, click **Manage Referral**.
3. From the **Current referrals** section, select the referral you want to edit.
4. Click **Edit**.
5. You can modify the host name and port for the server to which this referral value is pointing.
6. You can modify **Use SSL**, if the referral is to a secure (SSL) server or not.
7. You can modify the base DN in the directory information tree in the target server. For example ou=austin,o=sample.
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 414 for more information.
11. Click **OK**.
12. Repeat these steps for each referral you want to modify.

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=sample, issue the command:

```
idsldapmodify -D <adminDN> -w <adminPW> -M -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=sample?cn,sn,description?base?objectclass=*
```

Note: When accessing referral entries, you must specify the -M option to treat the entry like a normal entry, otherwise the server will return the referral.

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 14. 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

© Copyright IBM Corp. 2002, 2010

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. IBM Tivoli Directory Server 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 20. Server roles

Master/peer	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. 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. **Notes:** 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 303 for more information.
Forwarding (Cascading)	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.
Gateway	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.
Replica (read-only)	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.

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 228.

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.

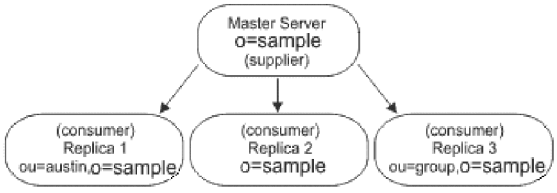


Figure 7. Master-replica replication

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:

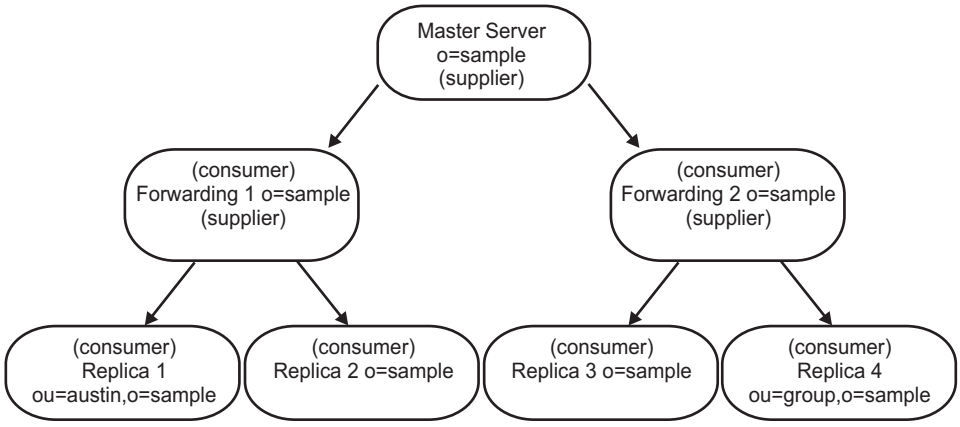


Figure 8. Cascading replication

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 225.

Note: In a Peer-to-peer replication setup with one replica server for each peer-master, if the primary master fails, the proxy server directs the requests to the backup master server. However, the proxy server will not fall back to the primary master until the backup master server fails.

The following is an example of peer-to-peer replication:

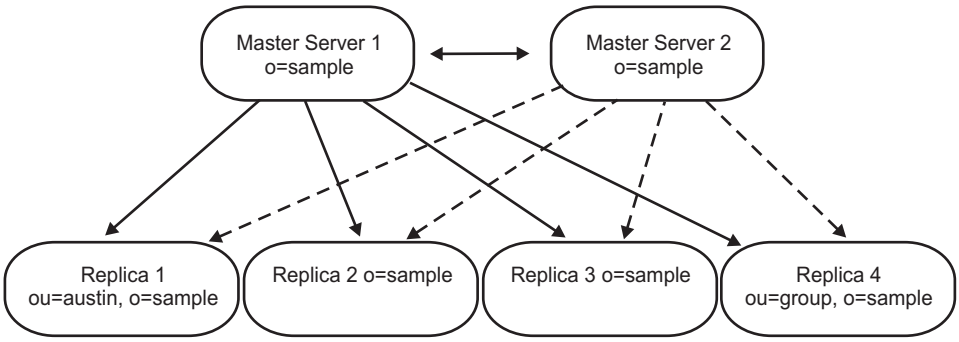


Figure 9. Peer-to-peer replication

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:

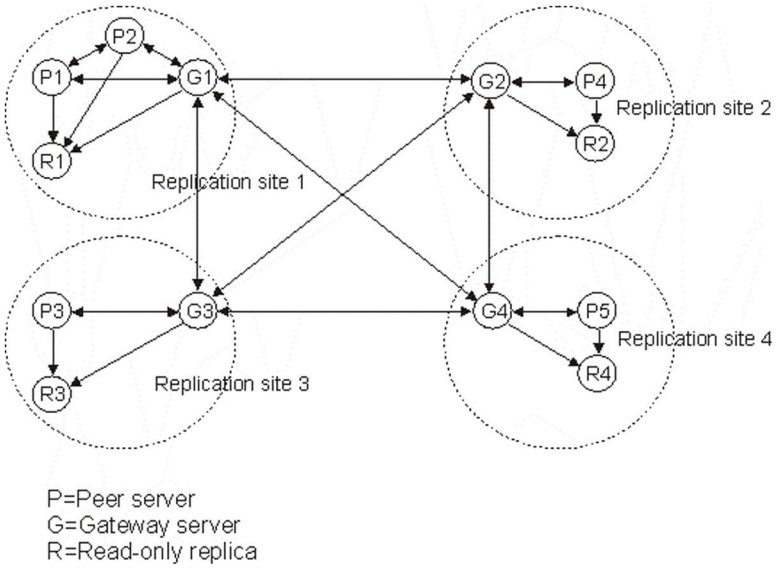


Figure 10. Gateway replication

Partial replication

Partial replication is a replication feature that replicates only the specified entries and a subset of attributes for the specified entries within a subtree. Using partial replication, an administrator can enhance the replication bandwidth depending on the deployment requirements. The attributes that are to be replicated are specified using a replication filter. For more information on partial replication, see "Partial Replication" on page 275

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.

Note: When a replication conflict is detected, an entry is re-added to resolve the replication conflict and the original entry prior to the re-add is written to the lostandfound.log file. This enables restoring any aspect of the original entry. The entire group entry can also be logged in the log file if the conflict is detected in a group entry.

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 17, “Logging Utilities,” on page 359 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 the **ldapdiff** command information in the *IBM Tivoli Directory Server Version 6.3 Command Reference* for information on resynchronizing servers.

To enhance the replication conflict resolution mechanism, the granularity of the timestamps is set to microseconds. Tivoli Directory Server also takes into consideration the fact that changes to the same entry at different times across peers may not converge because of clock skew. Therefore, to ensure convergence, each entry’s timestamp is increased monotonically with every update to ensure that after an update, an entry's timestamp should never be lesser than the timestamp prior to the update operation. This ensures the convergence of entries in spite of clock skew. Therefore, replication conflict resolution will function correctly even if the system clocks of the machines in the replication topology are not in sync.

Note:

- The granularity of timestamp is inconsequential in case of password policy operational attributes even though these attributes hold the timestamp value.
- In Tivoli Directory Server 6.2 and above versions, timestamp granularity of entries added or modified on Windows platform will be in milliseconds. However, in case of non-Windows platform the timestamp granularity will be in microsecond. This means that if an entry is replicated from a non-Windows machine to a Windows machine the timestamp of the entry will have microsecond level granularity. On the other hand, if an entry is replicated from a Windows machine to a non-Windows machine the timestamp of the entry will have millisecond level granularity.

For IBM Tivoli Directory Server 6.0 and above versions, to resolve replication conflict it needs the supplier to provide the entry's timestamp before the entry was updated on the supplier. Downlevel versions of the directory server acting as a supplier do 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 consumer server which is a IBM Tivoli Directory Server version 6.0 server in this case, takes the replicated timestamp and updates and applies it without conflict checking.

Notes:

1. Earlier versions of IBM Tivoli Directory Server do not support time stamp conflict resolution. If your topology contains earlier versions of 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 491 and “OIDs for supported and enabled capabilities” on page 493 to find out how to determine, if your servers support conflict resolution.
2. 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.

3. Replication conflict resolution can be disabled on a consumer if it does not have more than one supplier in the replication topology. In such a replication topology, messages related to the conflicting operation that are logged in the ibmslapd.log file can be considered as simple warning messages. As a work around to stop logging of these messages, you can set the configuration file parameter ibm-slapdNoReplConflictResolution to true using the ldapmodify command.

Setting up a load balancer is one method of resolving data conflict resolution.

A load balancer, such as 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.

When conflict resolution is enabled and changes are made to the membership of a given group on two servers concurrently, conflict resolution will be triggered repeatedly and this may affect the server's performance if the groups are large.

Tivoli Directory Server enables you to selectively enable or disable replication conflict resolution for modifications to group entries by defining the value of the "ibm-slapdEnableConflictResolutionForGroups" attribute present under the "cn=Replication, cn=configuration" entry of the configuration file.

If the attribute "ibm-slapdEnableConflictResolutionForGroups" is set to FALSE then no conflict resolution will be performed even if a conflict is detected for operations on group entries like adding, deleting, or renaming an attribute of type member or uniquemember.

However, a single modify request can target multiple attributes. In such a case, if in a single modify request any other attribute other than member or uniquemember is used, then replication conflict resolution will still be performed even though the attribute "ibm-slapdEnableConflictResolutionForGroups" is set to FALSE.

Use one of the following methods to enable or disable replication conflict resolution for modifications to group entries:

Using Web Administration Tool:

If you have not done so already, click **Server administration** in the Web Administration navigation area and then click **Manage server properties** in the expanded list. Click the **Conflict resolution** tab.

To configure replication conflict resolution for group members, do the following:

1. Select the **Enable replication conflict resolution for group members** check box. By default, this check box is not selected. The "ibm-slapdEnableConflictResolutionForGroups" attribute under the dn "cn=Replication, cn=configuration" in the configuration file is associated with this control. This attribute can be used in all replication topologies to speed up replication by avoiding undesirable conflict resolution to group entries. The default value of the "ibm-slapdEnableConflictResolutionForGroups" attribute is FALSE.
2. When you are finished, do one of the following:

- Click **OK** to apply your changes and exit this panel.
- Click **Apply** to apply your changes and stay on this panel.
- Click **Cancel** to exit this panel without making any changes.

Using command line

To enable or disable replication conflict resolution for group members, issue the following command:

```
ldapmodify -h <ldaphost> -p <ldap port> -D <bindDN> -w <password> -f <file>

where file contains:

dn:cn=Replication, cn=Configuration
changetype: modify
replace: ibm-slapdEnableConflictResolutionForGroups
ibm-slapdEnableConflictResolutionForGroups: <value to be set as either TRUE or FALSE>
```

Disabling replication conflict resolution feature

For Tivoli Directory server 6.1 and above versions, the replication conflict resolution feature can be disabled in the following different ways:-

1. **Manually editing the configuration file**:

   You can manually edit the configuration file and set the attribute "ibm-slapdNoReplConflictResolution" present under the entry "cn=master server, cn=configuration" to TRUE.

   After setting this attribute value to TRUE, the server must be restarted or a readconfig operation must be issued for the changes to take effect.

2. **Using ldapmodify utility**:

   You can set the value of the attribute “ibm-slapdNoReplConflictResolution" to TRUE using the ldapmodify utility as shown below:-

   ```
   ldapmodify -D <admin_dn> -w <admin_pwd>
   dn: cn=master server, cn=configuration
   changetype: replace
   replace: ibm-slapdNoReplConflictResolution
   ibm-slapdNoReplConflictResolution: TRUE
   ```

   The server must be restarted or a readconfig operation must be issued for the changes to take effect.

3. **Using Web Administration Tool**:

   Expand the **Replication management** category in the navigation area and click **Manage replication properties**.

   a. Select **Default credentials and referral** from the **Supplier information** list and click **Edit**.

   b. From the **Replication conflict resolution** combo box, select the value FALSE.

   c. Click **OK** to save your settings.

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 the database. 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 6.3 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 earlier versions of 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.
- 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.

Let us consider an example where you want to switch from a single threaded replication agreement to a multiple threaded replication agreement. Consider an example of a replication agreement on a server with server ID as *wingspread-2389* to a consumer with the LDAP URL *ldap://wingspread:1389*:

```
dn: cn=wingspread-1389,ibm-replicaServerId=wingspread-2389,
   ibm-replicaGroup=default,O=SAMPLE
cn: wingspread-1389
ibm-replicaconsumerid: wingspread-1389
ibm-replicacredentialsdn: cn=simple, cn=replication, cn=localhost
ibm-replicaurl: ldap://wingspread:1389
objectclass: ibm-replicationAgreement
objectclass: top
```

By default, the ibm-replicamethod is 1 (single threaded replication). To change the replication method and specify the number of connections to be used, issue the following ldapmodify command:

```
ldapmodify -D <binddn> -w <password> -p <ldapport> -v -i <file>

where file contains:

dn: cn=wingspread-1389,ibm-replicaServerId=wingspread-2389,
  ibm-replicaGroup=default,O=SAMPLE
ibm-replicamethod: 2
ibm-replicaconsumerconnections: 5
```

To verify the data in the replication agreement, issue the following command:

```
ldapsearch -L -p <ldapport>
-b cn=wingspread-1389,ibm-replicaServerId=wingspread-2389,ibm-replicaGroup=default,
O=SAMPLE objectclass=*
```

The following output is generated:

```
dn: cn=wingspread-1389,ibm-replicaServerId=wingspread-2389,
 ibm-replicaGroup=default,O=SAMPLE
cn: wingspread-1389
ibm-replicaconsumerid: wingspread-1389
ibm-replicacredentialsdn: cn=simple, cn=replication, cn=localhost
ibm-replicaurl: ldap://wingspread:1389
objectclass: ibm-replicationAgreement
objectclass: top
ibm-replicaconsumerconnections: 5
ibm-replicamethod: 2
ibm-replicationonhold: FALSE
```

Here, the replication method (ibm-replicamethod) value of 2 specifies that multiple threaded replication is to be used. The attribute "ibm-replicaconsumerconnections" indicates the number of connections that replication will use for sending the updates to the consumer. This value can range from 1 to 32. In this example, the supplier will establish 5 connections to the consumer to use for replication.

Note: After the replication agreement has been updated, you must restart the server for the changes to take effect.

Now, consider an example where you want to monitor replication status. Issue the following command:

```
ldapsearch -D <binddn> -w <password> -p <ldapport>
-b cn=wingspread-1389,ibm-replicaServerId=wingspread-2389,ibm-replicaGroup=default,
O=SAMPLE objectclass=* +ibmrepl
```

The following output is generated:

```
cn=wingspread-1389,ibm-replicaServerId=wingspread-2389,ibm-replicaGroup=default,O=SAMPLE
ibm-replicationChangeLDIF=N/A
ibm-replicationLastActivationTime=20080707152436Z
ibm-replicationLastChangeId=4855
ibm-replicationLastFinishTime=20080707152436Z
ibm-replicationLastResult=N/A
ibm-replicationLastResultAdditional=N/A
ibm-replicationNextTime=N/A
ibm-replicationPendingChangeCount=0
ibm-replicationState=ready
ibm-replicationFailedChangeCount=0
ibm-replicationperformance=[c=0,l=10,op=0,q=0,d=0,ws=0,s=0,ds=0,wd=0,wr=0,r=0,e=0,ss=1,rs=1]
ibm-replicationperformance=[c=1,l=10,op=0,q=0,d=0,ws=0,s=0,ds=0,wd=0,wr=0,r=0,e=0,ss=1,rs=1]
ibm-replicationperformance=[c=2,l=10,op=0,q=0,d=0,ws=0,s=0,ds=0,wd=0,wr=0,r=0,e=0,ss=1,rs=1]
ibm-replicationperformance=[c=3,l=10,op=0,q=0,d=0,ws=0,s=0,ds=0,wd=0,wr=0,r=0,e=0,ss=1,rs=1]
ibm-replicationperformance=[c=4,l=10,op=0,q=0,d=0,ws=0,s=0,ds=0,wd=0,wr=0,r=0,e=0,ss=1,rs=1]
```

In this example there are 5 connections to the consumer. The attribute names appear to the left of the equal signs. Some replication status information attributes are only used with single threaded replication, (shown here with the value 'N/A') while others are only for multiple threaded replication. Using "+ibmrepl" you can easily request all replication status information attributes. Use "++ibmrepl" to show all the attributes including the pending changes and logged replication errors.

The "ibm-replicationperformance" attribute is applicable to replication agreements only using multiple threaded replication. For single threaded replication, this attribute will have the value 'N/A'. You can interpret the ibm-replicationperformance data in the following manner:

c=0 This is the connection number. In this example there are 5 connections. The first connection will show the most traffic. The workload will determine how often the other connections are used.

l=10 This is the size limit for each queue. There are two queues for each connection and both have the same length. There is a queue for updates to be sent on the connection (called the send queue) and a queue for updates that have been sent but no response has been received from the consumer (called the receive queue). As updates are sent, they are moved from the send queue to the receive queue. When the receive queue hits its size limit, no more updates will be sent until some responses from the consumer are received. When the send queue hits its size limit, no more updates can be assigned to the connection. When the size limit for all of the connections send queues is reached, the supplier has to wait for the consumer to process the backlog.

op=0 The replication ID of the last operation assigned to the send queue of the connection. Replication IDs are assigned to any update that is to be replicated to a consumer.

q=0 The current size of the send queue.

d=0 The count of dependent updates (an add of an entry followed by a modify of the same entry counts as a dependency and dependent updates must be assigned to the same connection so they can be applied in the correct order).

ws=0 The number of times the send queue hit the size limit.

ds=0 The number of dependent updates sent.

wd=0 The number of times the send queue waited for a dependent update before sending additional updates.

wr=0 The number of times the receive queue hit the size limit.

r=0 The number of replicated updates waiting for a response from the consumer.

e=0 The number of replication errors reported by the consumer.

ss=1 The session count for the sender thread (incremented when the connection to the consumer is established).

rs=1 The session count for the receiver thread.

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 305 for details. These replication queues should be checked periodically for errors. Read "Viewing server errors" on page 300 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 308 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 225.

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 J, "Synchronizing two-way cryptography between server instances," on page 577 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.

When you are configuring a replica server, ensure that you set the master DN to an ID that is not the same as the admin DN on the replica server. If the DN IDs are the same, it is possible to bind to the replica with the combined adminDN-masterDN ID and make updates directly to the replica. The replica servers can become unsynchronized with the master server. This causes errors on the master and leads to data inconsistencies on the servers. All changes to the replica must be made by the master server binding with the masterDN. Attempted changes to a replica server are referred to the master server for the update process to take place.

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. To know more about replication of password policy attributes, see "Replicating password policy operational attributes" on page 536.

Note: If you are using a proxy server, password policy updates are replicated. Tivoli Directory Server also supports replication of schema updates to the consumer servers in a replication topology if the replication is setup for the CN=IBMPOLICIES context among all the backend servers that the proxy server is serving to. Global administrative group members can send request for schema updates through a proxy server to its backend servers. For more information on updates to schema, see "Schema updates in a distributed directory" on page 332.

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 keep schema synchronized, if the cn=ibmpolicies entry is replicated among all the servers in the directory topology. If you have distributed directory setup, then user must ensure that the schema updates are made through the Proxy server.

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:

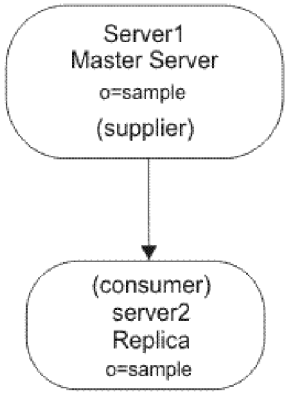


Figure 11. Basic master-replica topology

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.

Note: While setting up a new replica, when the data is exported to the replica , you must also copy the password policy entry.

The following sections explain how to accomplish the tasks mentioned above.

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 430 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.x 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 6.3 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 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
```

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 o=sample.

   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**.
8. Select the subtree from the **Replicated subtrees** table and click **Show topology**. The topology is displayed under the **Topology for selected subtree** heading. By default, if subtrees are available in the Replicated subtrees table, the topology of the first subtree in the table is displayed under the **Topology for selected subtree** heading.

   Depending on the selection of the node in the topology tree, the operations allowed on the node vary. Some of the operations are applicable only when a node other than the root is selected. Also, some operations are specific to the type of node, such as master server, forwarding server, replica server, and gateway server.

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 288 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:
   - From the **Server hostname:port** drop-down list, select an LDAP server for the replica server.

     If you want to provide another server as replica server, which is not registered on the console server, select Use entry from below item from the **Server hostname:port** drop-down list and then enter the host name and port number for the replica server in the field in the hostname:port format. The default port is 389 for non-SSL and 636 for SSL.
   - 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. Click **OK**.

     See "Adding credentials" on page 288 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 294.
   c. Keep the **Replication method** set to **Single threaded**.

      Note: Only IBM Tivoli Directory Server 6.0 and above versions 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 294.
   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 server port number. Make sure you have the correct admin server port number for the specific instance used. If the wrong admin server port number is specified, the admin server 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 294 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 J, "Synchronizing two-way cryptography between server instances," on page 577 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 515 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 288.
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 301 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 305 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=sample
objectclass: organization
objectclass: ibm-replicationContext
o: sample
```

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 229. 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=sample**:

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-slapdServerId** 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-slapdServerId** 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=sample
objectclass: organization
objectclass: ibm-replicationContext


###Copy the following to servers at v5.x and above.

###Replica Group
dn: ibm-replicaGroup=default, o=sample
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=sample
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=sample
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=sample
     -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 Appendix J, "Synchronizing two-way cryptography between server instances," on page 577, for information about cryptographic synchronization of servers.240.
   See the **idsdb2ldif** command information in the *IBM Tivoli Directory Server Version 6.3 Command Reference* 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>
    ```

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 requests are accepted in the order received without conflict resolution. See "Creating replication schedules" on page 303 for more information about conflict resolution.

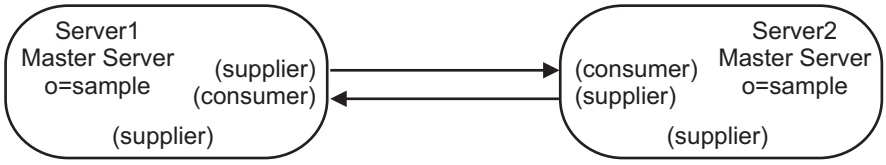


Figure 12. Basic peer-to-peer topology

This section shows how to set up a replication topology between two servers only.

Using Web Administration

Before you start, be sure that:

1. Both servers are running.
2. The servers are cryptographically synchronized if necessary. See Appendix J, "Synchronizing two-way cryptography between server instances," on page 577, in the *IBM Tivoli Directory Server Version 6.3 Administration Guide*.
3. In the Web Administration Tool, be sure that you are logged in to one of the servers. (This procedure assumes that you are logged in to the first of the two servers, server1.)

To set up two peer masters:

1. In the Web Administration Tool, expand the **Replication management** category in the navigation area and click **Manage topology**
2. Select the subtree that you want to replicate and click **Show topology**.

   If you want to view the existing topology, click the box next to the existing servers to expand the list of supplier servers.
3. Click **Replication topology** to highlight it, and then click **Add master**.
4. On the **Server** tab of the Add master window:
   a. Select **Server is a gateway** to make this server a Gateway server or select **Supplier gateway** and then select a server from the drop-down list to add the server as a master server.
   b. From the **Server hostname:port** drop-down list, select an LDAP server for the master server.

      If you want to provide another server as master server, which is not registered on the console server, select Use entry from below item from the **Server hostname:port** drop-down list and then enter the host name and port number for the master server in the field in the hostname:port format.

      Note: The default port is 389 for non-SSL and 636 for SSL.

c. Select the **Enable SSL encryption** check box to enable SSL communications.
d. In the **Peer master name** field, enter the server name or leave this field blank to use the host name.
e. 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**.
f. Optionally, enter a description of the server.
g. You must specify the credentials that the server uses to communicate with the master server. Click **Select** beside the **Credential object** field. The Select credential window is displayed. On the Select credential window:
   1) Select the location for the credentials you want to use. Preferably this is cn=replication,cn=localhost.

      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 replicated to the servers.

        The location cn=replication,cn=IBMpolicies is available only 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.
   2) If you have already created a set of credentials:
      a) Click **Show credentials**. A list of existing credentials is shown in the **Select credentials** field.
      b) Expand the list of credentials and select the one you want to use.
   3) If you do not have preexisting credentials, click **Add credentials** to create the credentials. See "Adding credentials" on page 288, in the *IBM Tivoli Directory Server Version 6.3 Administration Guide* for additional information about agreement credentials.
   4) Click **OK**.

5. On the **Additional** tab:
   a. If you want to use an existing replication schedule, select the replication schedule from the drop-down list.

      If you want to create a new replication schedule:
      1) Click **Add**.
      2) See "Creating replication schedules" on page 303, in the *IBM Tivoli Directory Server Version 6.3 Administration Guide* for information about replication schedules.

         When you return to the Add master panel, select the schedule you created from the list of schedules.
   b. From the **Capabilities replicated to consumer** list, 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 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.

c. Check the **Add credential information on consumer** check box. This selection automatically updates the supplier credentials in the configuration file of the consumer server. This enables the topology information to be replicated to server2.
   - Type the Administrator DN for the consumer server (server2); for example, `cn=root`.

     Note: If the administrator DN that was created during the server configuration process was `cn=root`, then enter the full administrator DN. Do not use only `root`.
   - Type the Administrator password for the consumer server; for example, `secret`.

d. Click **OK**.

e. On the `Create additional supplier agreements` panel, supplier and consumer agreements are listed between the new master server and any existing servers. Clear the check boxes for any agreements that you do not want to be created.

f. Click **Continue**.

g. If a message is displayed asking if you want to restart server2, click **Yes**. Other 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**.

h. Add the appropriate credentials to configure agreements from server2 to server1:
   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:
      a) Click **Show credentials**. A list of existing credentials is shown in the **Select credentials** field.
      b) Expand the list of credentials and select the one you want to use.
   3) Click **OK**.

   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 288, in the *IBM Tivoli Directory Server Version 6.3 Administration Guide* for information about credentials.

i. Click **OK** to create the peer-master.

j. 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, have a configured database, and have been started at least once for initialization purposes. (Be sure to read Appendix J, "Synchronizing two-way cryptography between server instances," on page 577, in the *IBM Tivoli Directory Server Version 6.3 Administration Guide* before you start the server instances.)

Note: The subtree you want to create is the following:

```
dn: o=sample
objectclass: organization
objectclass: ibm-replicationContext
o: sample
```

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=sample**:

1. Start servers server1 and server2 in configuration only mode. On each of the servers issue the command:

   ```
   idsslapd -I <LDAPinstance> -a
   ```

2. If the administration server (**idsdiradm**) is not running for any 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-slapdReplication
   cn: Master Server
   ibm-slapdMasterDN: cn=any
   ibm-slapdMasterPW: secret
   ```

   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:

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

where <*instancename*> is the name of the directory server instance you want to stop.

```
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 computer 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
objectclass: container

###Bind Credentials/method to peer server - replication agreement
###points to this.
dn: cn=simple,cn=replication,cn=IBMpolicies
objectclass:ibm-replicationCredentialsSimple
cn:simple
replicaBindDN:cn=any
replicaCredentials:secret
description:Bind method of the peer master (server1)to the peer (server2)
```

7. Issue the command:

```
idsldif2db -r no -i  <mycredentialsfile> -I <instance_name>
```

8. Copy <*mycredentialsfile*> to the computer where server2 is located and issue the command:

```
idsldif2db -r no -i  <mycredentialsfile> -I <instance_name>
```

9. At the computer where server1 is located create a file, <*mytopologyfile*>, where <*mytopologyfile*> includes the following:

   Note: Replace all occurrences of <*server1-uuid*> in the following files with the value of the **ibm-slapdServerId** 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 an AIX, Linux, or Solaris system. Similarly, all occurrences of the <*serverx-uuid*> (where x represents 1 or 2) must be replaced with the value of the **ibm-slapdServerId** attribute from the respective server's **cn=Configuration** entry.

```
dn: o=sample
o: sample
objectclass: top
objectclass: container
objectclass: ibm-replicationContext

dn: ibm-replicaGroup=default, o=sample
objectclass: top
objectclass: ibm-replicaGroup
ibm-replicaGroup: default

dn: ibm-replicaServerId=<server1-uuid>,ibm-replicaGroup=default,o=sample
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=sample
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=sample
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(master) agreement

#server2 to server1 agreement
dn: cn=server1,ibm-replicaServerId=<server2-uuid>,
    ibm-replicaGroup=default,o=sample
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: 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=sample" -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 synchronized. See the **idsdb2ldif** command information in the *IBM Tivoli Directory Server Version 6.3 Command Reference* for more information.

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 Appendix J, "Synchronizing two-way cryptography between server instances," on page 577, 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 515, in the *IBM Tivoli Directory Server Version 6.3 Administration Guide* 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.

13. Restart server1.
14. Copy the <*mymasterfile.ldif*> file to the computer where server2 is located.
15. On the computer where server2 is located, issue the following command:
    ```
    idsldif2db -r no -i <mymasterfile.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:

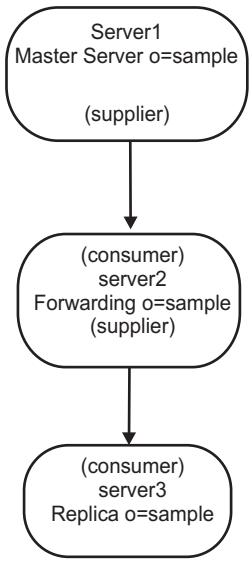


Figure 13. Master-forwarding server-replica topology

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 234.
2. Create a new replica server for the original replica. See "Adding a replica server" on page 294.
3. Copy data to the replicas. See "Copying data to the replica" on page 239.

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 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 must 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:
   - From the **Server hostname:port** drop-down list, select an LDAP server for the replica server.

     If you want to provide another server as replica server, which is not registered on the console server, select Use entry from below item from the

Server hostname:port drop-down list and then enter the host name and port number for the replica server in the field in the hostname:port format. The default port is 389 for non-SSL and 636 for SSL.

- 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 288 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 239 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 301 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 in 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, have a configured database, and have been started at least once for initialization purposes. (Be sure to read Appendix J, "Synchronizing two-way cryptography between server instances," on page 577, in the *IBM Tivoli Directory Server Version 6.3 Administration Guide* before you start the server instances.)

Note: The subtree you want to create is the following:

```
dn: o=sample
objectclass: organization
objectclass: ibm-replicationContext
o: sample
```

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=sample**:

1. At the computer where the master server is located, create a file to contain the agreement information; for example *myreplicainfofile* where *myreplicainfofile* contains the following:

   Note: Replace all occurrences of <*server1-uuid*> in the following files with the value of the **ibm-slapdServerId** 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 by using the **grep** command on the ibmslapd.conf file, if you have an AIX, Linux, or Solaris system. Similarly, all occurrences of the <*server2-uuid*> and the <*server3-uuid*> must be replaced with the value of the **ibm-slapdServerId** attribute from the respective server's **cn=Configuration** entry.

```
dn: cn=replication,cn=IBMpolicies
objectclass: container

dn: o=sample
objectclass: organization
```

```
objectclass: ibm-replicationContext

dn: ibm-replicaGroup=default, o=sample
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=sample
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=sample
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=sample
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=sample
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, by using the following command:

   ```
   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=sample
    -k <key seed> -t <key salt>
   ```

   Note: You must use the `-I` option if there is more than one directory server instance. You must use the `-k` and `-t` options if keys on the server are not synchronized. See the **idsdb2ldif** command information in the *IBM Tivoli Directory Server Version 6.3 Command Reference* for more information.

   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 Appendix J, "Synchronizing two-way cryptography between server instances," on page 577, in the *IBM Tivoli Directory Server Version 6.3 Administration Guide* 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 515, in the *IBM Tivoli Directory Server Version 6.3 Administration Guide* 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.
5. Copy the <*masterfile.ldif*> file to the computer where server2 is located.
6. Start the forwarder, server2, in configuration only mode.

   ```
   idsslapd -I <LDAPinstance> -a
   ```
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:

   ```
   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 the <*masterfile.ldif*> file to the computer 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=server3 BindCredentials".

13. Stop server3.

    ```
    ibmdirctl -h server3 -D <adminDN> -w <adminPW> -p <port> stop
    ```

14. Save the **ibmslapd.conf** file.
15. At the computers 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 in order to replicate global updates such as cn=schema.

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 that you are adding must have correct ACLs, if it is not a suffix in the server.

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>
```

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 287).

Using the forwarding topology created in "Changing the replica to a forwarding server" on page 251, 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 294.
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 292.

   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 239.
5. Start replication. See "Managing queues" on page 305.

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, have a configured database, and have been started at least once for initialization purposes. (Be sure to read Appendix J, "Synchronizing two-way cryptography between server instances," on page 577, in the *IBM Tivoli Directory Server Version 6.3 Administration Guide* before you start the server instances.)

Note:

```
dn: o=sample
objectclass: organization
objectclass: ibm-replicationContext
o: sample
```

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:

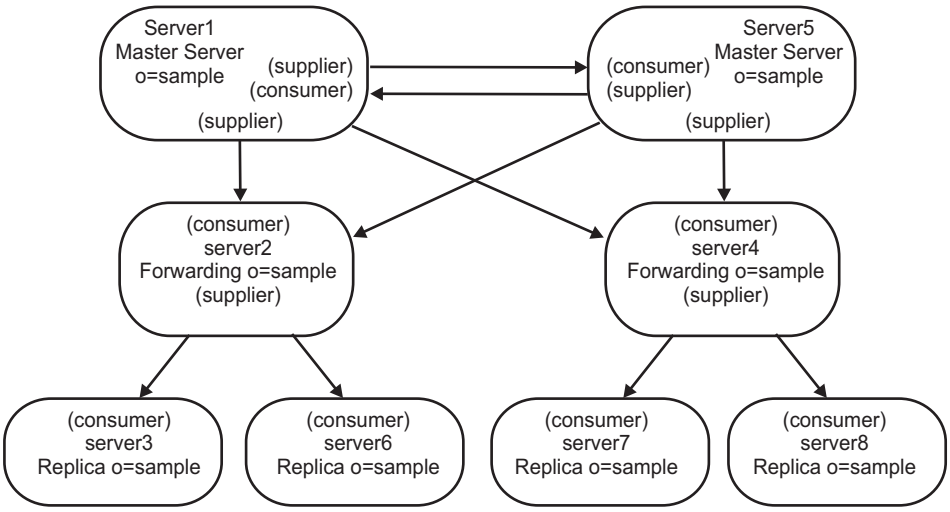


Figure 14. A peer-to-peer topology

- 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=sample**:

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=any
   ibm-slapdMasterPW: secret
   ```

   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 that you have a backup of the ibmslapd.conf file.
5. At the computer where the master server, server1, is located, create a file to contain the agreement information; for example, *mycredentialsfile*, where *mycredentialsfile* contains the following:

   ```
   dn: cn=replication,cn=IBMpolicies
   objectclass: container

   ###Bind Credentials/method to peer/forwarder server - replication agreement
   ###points to this.
   dn: cn=simple,cn=replication,cn=IBMpolicies
   objectclass:ibm-replicationCredentialsSimple
   cn:simple
   replicaBindDN:cn=any
   replicaCredentials:secret
   description:Bind method of the master to the peer/forwarder
   ```
6. Issue the following 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 the <*mycredentialsfile*> file to the computers where server5, server2, and server4 are located and issue the following command on each computer:

   ```
   idsldif2db -r no -i  <mycredentialsfile> -I <instance_name>
   ```
9. At the computer 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-slapdServerId** 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 an AIX, Linux, or Solaris system. Similarly, all occurrences of the <*serverx-uuid*> (where x represents a number) must be replaced with the value of the **ibm-slapdServerId** attribute from the respective server's **cn=Configuration** entry.

   ```
   dn: o=sample
   o: sample
   objectclass: top
   objectclass: organization
   objectclass: ibm-replicationContext

   dn: ibm-replicaGroup=default, o=sample
   objectclass: top
   objectclass: ibm-replicaGroup
   ibm-replicaGroup: default

   dn: ibm-replicaServerId=<server1-uuid>,ibm-replicaGroup=default,o=sample
   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=sample
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=sample
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=sample
objectclass: top
objectclass: ibm-replicaSubentry
ibm-replicaServerId: <server4-uuid>
ibm-replicationServerIsMaster: false
cn: server4
description: server4 (forwarder) ibm-replicaSubentry

#server1 to server5 agreement
dn: cn=server5,ibm-replicaServerId=<server1-uuid>,
    ibm-replicaGroup=default,o=sample
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=sample
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=sample
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=sample
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=sample
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=sample
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=sample
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=sample
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=sample
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=sample
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=sample" -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 synchronized. See the **idsdb2ldif** command in the *IBM Tivoli Directory Server Version 6.3 Command Reference* for more information.

    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 Appendix J, "Synchronizing two-way cryptography between server instances," on page 577, in the *IBM Tivoli Directory Server Version 6.3 Administration Guide* 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 515, in the *IBM Tivoli Directory Server Version 6.3 Administration Guide* 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.
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 the <*mymasterfile.ldif*> file to the computers where server2, server3, server4, server5, server6, server7, and server8 are located.
18. At each of these computers, issue the following command:

    ```
    idsldif2db -r no -i <mymasterfile.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>
    ```

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

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:

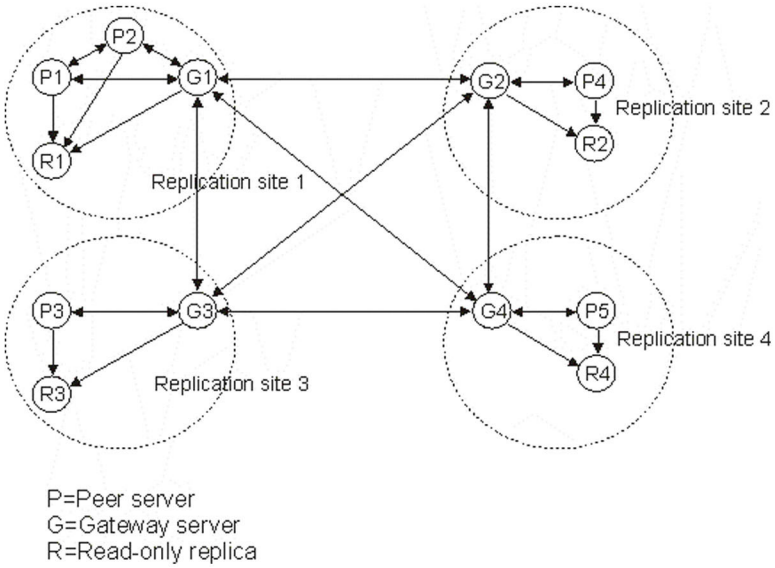


Figure 15. A replicating network with Gateway servers

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 ibmslapdcfg.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=sample, you need to export the subtree ibm-replicagroup=default,o=sample.

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:

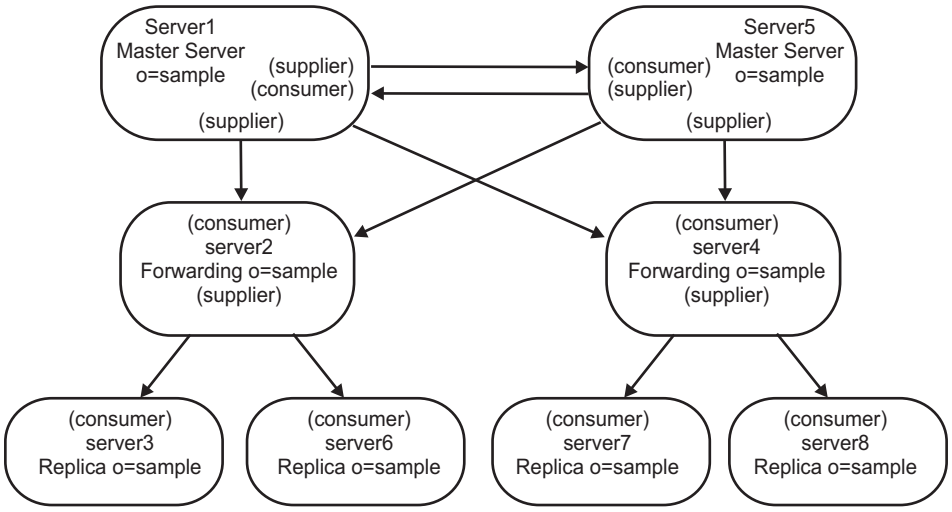


Figure 16. Initial peer-to-peer topology

- 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 292 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
	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 288.

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-gatway)
  – server1 (master-gateway)

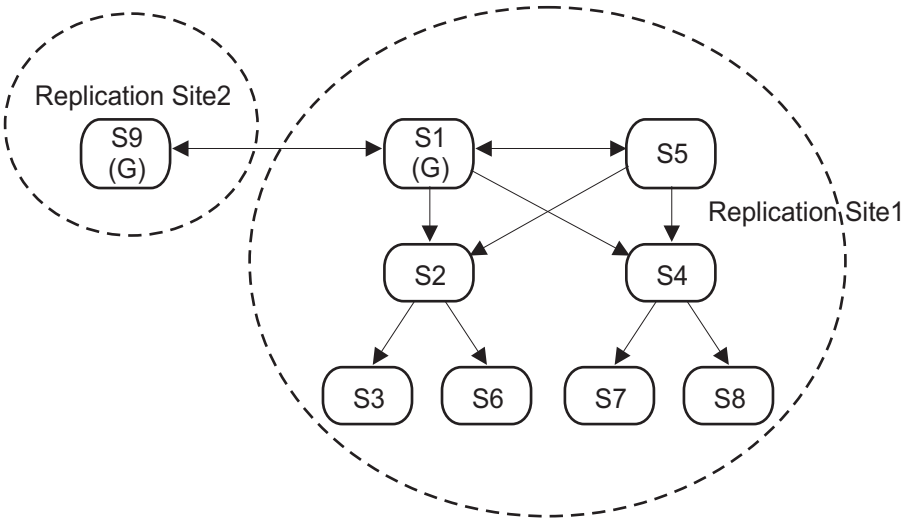


Figure 17. A gateway topology with two replication sites

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 239 and "Adding the supplier information to a replica" on page 301 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.

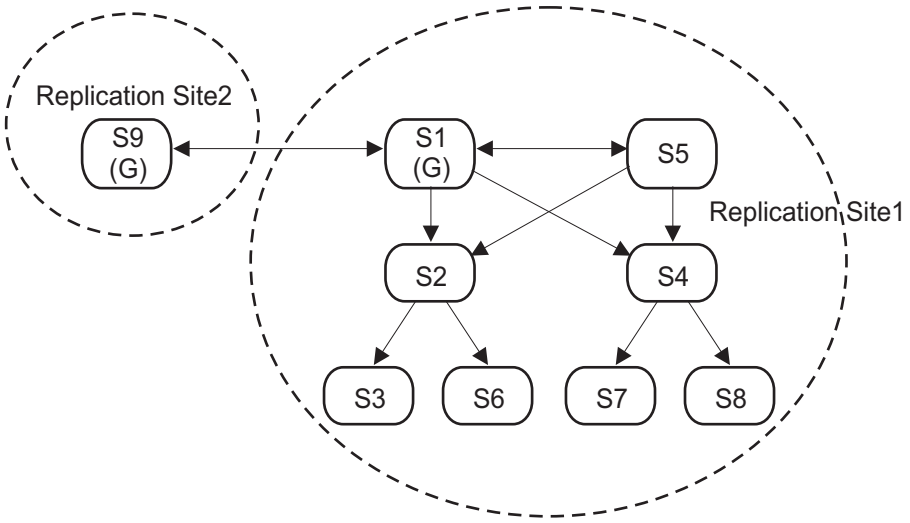


Figure 18. A gateway topology with two replication sites

1. Create server9. Create an instance (see "Creating and administering instances" in the *IBM Tivoli Directory Server Version 6.3 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=sample
   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=sample
   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=sample"
```

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=sample
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=sample
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=sample"
```

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=sample"
```

10. Perform an idsdb2ldif command to create an LDIF file on server1:

```
idsdb2ldif  -s "ou=test,o=sample" -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 272.

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 275.

12. Unquiesce server1:

```
idsldapexop -D <adminDN> -w <admin_password> -h server1 -p <port> -op
 quiesce -end -rc "ou=test,o=sample"
```

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=sample"
```

    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=sample
o: sample
objectclass: top
objectclass: organization
objectclass: ibm-replicationContext

dn: ibm-replicaGroup=default,ou=test,o=sample
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=sample
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=sample
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=sample
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=sample
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=sample
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=sample
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=sample
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=sample
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=sample
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=sample
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=sample
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=sample
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=sample
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=sample
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=sample
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=sample
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=sample
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: sample
objectclass: top
objectclass: container
objectclass: ibm-replicationContext

dn: cn=simple,cn=replication,cn=ibmpolicies
objectclass: ibm-replicationCredentialsSimple
cn: simple
replicaBindDN: cn=any
replicaCredentials: secret
```

Partial Replication

Partial replication is a replication feature that replicates only the specified entries and a subset of attributes for the specified entries within a subtree. The entries and attributes that are to be replicated are specified by the LDAP administrator. Using partial replication, an administrator can enhance the replication bandwidth depending on the deployment requirements. For instance, an administrator may choose the entries of the object class person with cn, sn, and userPassword attributes to be replicated and description attribute not to be replicated.

The attributes that are to be replicated are specified using a replication filter. A replication filter may be associated with a particular replication agreement and will be based on object classes. A set of attributes pertaining to an object class constitutes a replication filter. The list of attributes selected for an object class can either be a part of an inclusion list or an exclusion list. An inclusion list is list of attributes that will be selected for replication while an exclusion list is list of attributes that will not be selected for replication. However, the administrator must ensure that the MUST attributes of an object class should not be excluded. If MUST attributes are excluded for an object class, then the replication of entries containing that object class might fail, which will set the replication state as retrying. This might block the replication for that particular replication agreement. For example, consider the following:

```
dn: cn=replicationfilter1,cn=localhost
objectclass: ibm-replicationfilter
ibm-replicationFilterattr: (objectclass=person) : !(sn)
ibm-replicationFilterattr: (objectclass=*) : (*)
```

In this case, if the filter `ibm-replicationFilterattr: (objectclass=person) : !(sn)` is given, then entries with object class as `person` will fail to replicate and block the replication because `sn` is a MUST attribute for the person object class.

The following attributes are always replicated, irrespective of their presence in the exclusion list

- Object class attributes of an entry

- Naming attribute
- All operational attributes

For information about known limitations of partial replication, see Chapter 10, "General Information, Known Limitations and General Troubleshooting" in the *IBM Tivoli Directory Server Version 6.3 Problem Determination Guide*

The partial replication feature can be managed using the web administration tool or from the command line.

Using Web Administration Tool

If you have not done so already, expand the **Replication management** category in the navigation area of the Web Administration Tool and click **Manage filters**. This panel is available only if the server supports the filter-based replication capability.

On this panel you can:

- View subtrees where replication filters are stored
- Add filters
- Edit filters
- Delete filters
- Copy filters
- View filters

Add filters

To add a replication filter, you first select a subtree from the Select a subtree box on the Manage filters panel and then click **Add** to display the Add Replication Filter panel.

Add Replication Filter- General: This panel contains controls for adding details for a replication filter.

To add a replication filter:

1. In the Filter name box, enter a name for the filter. For example, myfilter1.
2. From the Available object classes box, select the object classes on which you want to create filter.
3. Click **Add** to populate the Selected object classes box with the object classes from the Available object classes box.
4. Select the **Define filter for remaining object classes** check box.
5. To continue with adding a replication filter for filtered attributes, click **Next**.

Add Replication Filter- Filtered Attributes: This panel provides the facility to choose the attributes to be replicated for the selected object classes. This panel is invoked on clicking the **Next** button on the Add Replication Filter- General panel.

To specify the attributes to be replicated for an object class:

1. Click the **Select** column of the object class row for which you want to specify attributes to be replicated.
2. Click the **Manage filter attribute** button or select Manage filter attribute from the Select Action list and then click **Go**.

Manage filter attributes: The Manage filter attributes panel is used for specifying object class attributes for replication filter.

To specify attributes for replication filter:

1. Clear the **Select all attributes as filtered attributes** check box.

   Note: If you want to specify all the attributes of the selected object class in a replication filter, select the **Select all attributes as filtered attributes** check box.
2. Select the required attributes in the Available attributes box.
3. Click **Add** to move the selected attributes from Available attributes to Filtered attributes.
4. To include the attributes in the Filtered attributes box in the replication filter, click **Include selected filtered attributes**.
5. To exclude the attributes in the Filtered attributes box from the replication filter, click **Exclude selected filtered attributes**.
6. Click **OK**.
7. To save the replication filter, click **Finish** on the Add Replication Filter- Filtered Attributes panel.

Delete filters

To delete a replication filter, select a replication filter in the Filters for selected subtree box on the Manage filters panel and then click **Delete**.

Edit filters

To edit a replication filter, you select a filter from the Filters for selected subtree box on the Manage filters panel and then click **Edit**.

Edit Replication Filter- General: This panel contains controls for modifying the content of a selected filter.

To edit a replication filter:

1. From the Available object classes box, select the object classes that you want to add to the filter.
2. To edit the existing filter:
   a. Click **Add** to populate the Selected object classes box with the object classes from the Available object classes box.
   b. Click **Remove** to remove a selected object class from the Selected object classes box.
3. Select the **Define filter for remaining object classes** check box.
4. To continue editing the replication filter for filtered attributes, click **Next**.

Edit Replication Filter- Filtered Attributes: This panel provides the facility to choose the attributes to be replicated, when the filter is selected. This panel is invoked on clicking the Next button on the Edit Replication Filter- General panel.

To specify the attributes to be replicated for an object class:

1. Click the **Select** column of the object class row for which you want to edit the existing attributes list for the selected object class in the replication filter.
2. Click the **Manage filter attribute** button or select Manage filter attribute from the Select Action list and then click **Go** to display the Manage filter attributes panel.
3. In the Manage filter attributes panel, specify the attributes that are to be included or excluded in the replication filter definition.

Copy Filters

To copy the details of a replication filter to another replication filter, you first select a subtree from the Select a subtree box and then select a filter stored under that subtree from Filters for selected subtree on the Manage filters panel and then click **Copy**.

Copy Replication Filter- General: To copy a replication filter:

1. From the Filter location box, select the subtree under which you want to copy the selected replication filter.
2. In the Filter name box, enter a name for the filter. For example, myfilter2.
3. From the Available object classes box, select the object classes that you want to add to the existing filter.
4. Click **Add** to populate the Selected object classes box with the object classes from the Available object classes box.
5. Select the **Define filter for remaining object classes** check box.
6. To continue with copying of the filter for filtered attributes, click **Next**.

Copy Replication Filter- Filtered Attributes: This panel provides the facility to choose the attributes to be replicated for the selected object classes. This panel is invoked on clicking the Next button on the Copy Replication Filter- General panel.

To specify the attributes to be replicated for an object class:

1. Click the **Select** column of the object class row for which you want to specify attributes to be replicated.
2. Click the **Manage filter attribute** button or select Manage filter attribute from the Select Action list and then click **Go** to display the Manage filter attributes panel.
3. In the Manage filter attributes panel, specify the attributes that are to be included or excluded in the replication filter definition.

Using command line

Issue the following command to add a replication filter:

```
ldapadd –D cn=root –w root

dn: cn=replicationfilter,cn=localhost
objectclass: ibm-replicationfilter
ibm-replicationFilterAttr: (objectclass=person):(cn,sn,description)
ibm-replicationFilterAttr: (objectclass=printer):!(cn,color)
ibm-replicationFilterAttr: (objectclass=*): (*)
```

The above example states that for entries of type “person”, the attributes cn, sn, and description will be sent to the replica. The rest of the attributes present in the entry will not be sent. For entries of type “printer”, all attributes except cn and color will be sent. For the remaining entries, all attributes will be sent.

Now, modify the replication agreement to add the DN of the filter entry. To do this, issue the following command:

```
ldapmodify –D cn=root –w root

dn: cn=replica1,ibm-replicaServerId=master-uuid,ibm-replicaGroup=default,o=sample
changetype: modify
add: ibm-replicationFilterDN
ibm-replicationFilterDN: cn=replicationfilter,cn=localhost
```

Examples of replication filter

Given below are some examples that explain the usage of replication filter-:

Note: Tivoli Directory Server 6.2 and above versions do not support alternate names in replication filter.

Example 1

```
dn: cn=replicationfilter, cn=localhost
objectclass: ibm-replicationFilter
ibm-replicationFilterAttr: (objectclass=person):(*)
ibm-replicationFilterAttr: (objectclass=*): !(*)
```

The first filter attribute in this example specifies that all attributes of entry type "person" will be replicated. The second filter attribute specifies that no other entries except those of type "person" will be replicated. This means that only entries of type "person" will be replicated and no other entries will be replicated.

Example 2

```
dn: cn=replicationfilter, cn=localhost
objectclass: ibm-replicationFilter
ibm-replicationFilterAttr: (objectclass=person):(cn,sn,userPassword)
ibm-replicationFilterAttr: (objectclass=managerOf):(managerOfDept)
ibm-replicationFilterAttr: (objectclass=*): !(managerOfDept)
```

For this example, consider an entry "cn=Ricardo Garcia,o=sample" of type "person". A new auxiliary objectclass "managerOf" is attached to the above entry. Therefore the entry "cn=Ricardo Garcia,o=sample" will contain both "person" and "managerOf" object classes.

The first filter attribute specifies that attributes cn, sn, and userpassword of entry type "person" will be replicated. The second filter attribute specifies that attribute managerOfDept of entry type "managerOf" will be replicated. The third filter attribute specifies that attribute managerOfDept will not be replicated for any other entry except those of type "person" or "managerOf".

Therefore, for an entry type person, the attribute cn, sn, and userPassword will be replicated. For the entry "cn=Ricardo Garcia,o=sample", containing objectclass person and managerOf, the attributes cn, sn, userPassword, and managerOfDept will be replicated. For any other entry that is not of type "person" or "managerOf", all attributes except managerOfDept will be replicated.

Example 3

```
dn: cn=replicationfilter, cn=localhost
objectclass: ibm-replicationFilter
ibm-replicationFilterAttr: (objectclass=person):(cn,sn,userPassword)
ibm-replicationFilterAttr: (objectclass=inetOrgPerson):!(userPassword,employeeNumber)
ibm-replicationFilterAttr: (objectclass=*): !(*)
```

For this example, consider an entry "cn=Ricardo Garcia,o=sample" of type "person" and another entry "cn=Jane Smith,o=sample" of type "inetOrgperson". The entry "cn=Jane Smith,o=sample" will contain both "person" and "inetOrgPerson" object classes.

The first filter attribute specifies that attributes cn, sn, and userpassword of entry type "person" will be replicated. The second filter attribute specifies that attributes userPassword and employeeNumber of entry type "inetOrgPerson" will not be

replicated. The third filter attribute specifies that any attribute for any other entry except that of type “person” or “inetOrgPerson” will not be replicated.

Therefore, for the entry “cn=Ricardo Garcia,o=sample”, the attributes cn, sn, and userPassword will be replicated. For the entry “cn=Jane Smith,o=sample”, which matches the first and second replication filters, only attributes cn and sn will be replicated. The attribute userPassword being present in both the inclusion and exclusion list, will be eliminated as exclusion takes precedence over inclusion. For any other entry, that is not of type “person” or “inetOrgPerson” no attributes will be replicated.

Excluding replication topology information

In the IBM Tivoli Directory Server configuration, replication topology information is present in the DB2 database of every directory server instance participating in replication. In this replication environment, there may be a situation where you may want to export the contents of the DB2 database of a directory server instance to an LDIF file but exclude the replication topology related data. Tivoli Directory Server provides a file named replfilterdn.ldif at the location <IDS_LDAP_HOME>/examples. The entries in this file can be used to suppress replication topology information in the resulting ldif file. Given below is an example of the replfilterdn.ldif file

```
dn: cn=replicationfilter,cn=localhost
objectclass: ibm-replicationfilter
ibm-replicationFilterAttr: (objectclass=ibm-replicaGateway):!(*)
ibm-replicationFilterAttr: (objectclass=ibm-replicaGroup):!(*)
ibm-replicationFilterAttr: (objectclass=ibm-replicaSubentry):!(*)
ibm-replicationFilterAttr: (objectclass=ibm-replicationAgreement):!(*)
ibm-replicationFilterAttr: (objectclass=ibm-replicationCredentials):!(*)
ibm-replicationFilterAttr: (objectclass=ibm-replicationCredentialsExternal):!(*)
ibm-replicationFilterAttr: (objectclass=ibm-replicationCredentialsKerberos):!(*)
ibm-replicationFilterAttr: (objectclass=ibm-replicationCredentialsSimple):!(*)
ibm-replicationFilterAttr: (objectclass=ibm-replicationDailySchedule):!(*)
ibm-replicationFilterAttr: (objectclass=ibm-replicationWeeklySchedule):!(*)
ibm-replicationFilterAttr: (objectclass=*): (*)
```

To suppress the replication topology information, you must first create an entry in the directory server instance from which you want to export the data. This entry specifies the filter properties to use during the export. The ibm-replicationFilterAttr values state which entries to exclude and include.

Let us consider an example where you want to exclude all "ibm-replicagroup" entries. These entries are identified by the value "ibm-replicaGroup" present for the objectclass attribute. This exclusion is achieved by the second value of the ibm-replicationFilterAttr as shown above. The last value for the ibm-replicationFilterAttr indicates that all attributes for any other entry, which does not meet the criteria of being a replication topology related entry, must be included.

Create a file, filterdn.ldif, with the entries given above and issue an ldapadd command to add the entry to the directory server instance:

```
idsldapadd -D <binddn> -w <password> -f filterdn.ldif
```

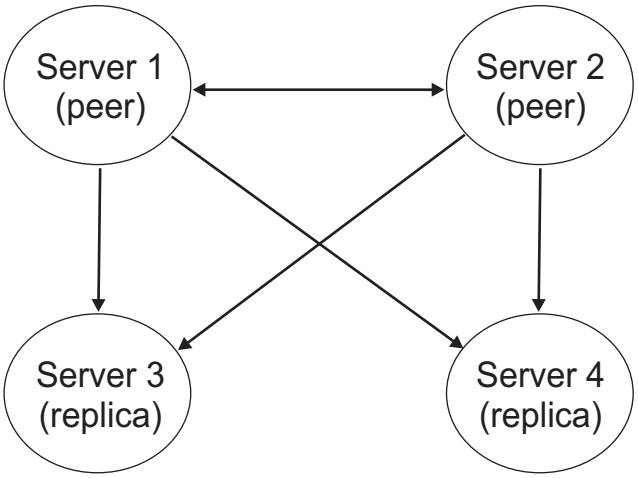
To export the DB2 database information from the directory server instance and exclude the replication related data, specify the DN of the newly created filter entry, cn=replicationfilter,cn=localhost using the –n option.

```
idsdb2ldif -I <instance_name> -o <output_file> -n "cn=replicationfilter,cn=localhost"
```

The resulting output file will not contain any of the replication topology related entries.

Recovery procedures

The following procedures are based on a system topology 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 ibmslapdcfg.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 restore 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 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 ibmslapdcfg.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=sample
   ```
5. Wait for server 1 to replicate all pending updates to server 2, when ibm-replicationpendingchangecount is zero.

   ```
   idsldapsearch –D <admin_dn> -w <admin_pw> -h <server1> -b
    <dn of agreement with server2> -s base
    objectclass=* ibm-replicationpendingchangecount
   ```
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=sample
   ```
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 IBM Tivoli Directory Server Version 6.3 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 282.
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-replicationpendingchangecount 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 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 values 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 Version 6.3 Programming Reference*.

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 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
```

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 430 for information on how to edit ACLs and see Chapter 19, "Access control lists," on page 419 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 286 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 187 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. Select the **Enable PKCS#11 interface support** check box to enable PKCS#11 support of crypto hardware.
  g. Click **Finish**.

  See "Secure Sockets Layer" on page 141 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:

- In the JAVA_HOME\jre\lib\security\java.security file, check if the following two entries to register JCE provider and CMS provider are present. If the entries do not exist, add this entry in the java.security file by entering the following:

  ```
  security.provider.X=com.ibm.crypto.provider.IBMJCE
  security.provider.X+1=com.ibm.security.cmskeystore.CMSProvider
  ```

  where, **X** is the next number in the order.
- GSKit must be installed and <*install_location*>gsk8\lib or <*install_location*>gsk8\lib64 depending on the platform 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.
  a. In the Edit credential panel you can modify:
     - The **Key file**
     - The **Password**
     - The **Key label**
     - The **Description** of the credential
  b. Select the **Enable PKCS#11 interface support** check box to enable PKCS#11 support of crypto hardware.

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 ALCs for and click **Edit ACL**.
3. See "Working with ACLs" on page 430 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:

- Select **Server is a gateway** to make this server a Gateway server or select **Supplier gateway** and then select a server from the drop-down list to add the server as a master server.
- From the **Server hostname:port** drop-down list, select an LDAP server for the master server.

  If you want to provide another server as master server, which is not registered on the console server, select Use entry from below item from the **Server hostname:port** drop-down list and then enter the host name and port number for the master server in the field in the hostname:port format.

  Note: The default port is 389 for non-SSL and 636 for SSL.
- Select the **Enable SSL encryption** check box to enable SSL communications.
- 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 288 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 303
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 420) and password policy ("Setting password policy" on page 170), 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 288.

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 241.

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:

- In the JAVA_HOME\jre\lib\security\java.security file, check if the following two entries to register JCE provider and CMS provider are present. If the entries do not exist, add this entry in the java.security file by entering the following:

  ```
  security.provider.X=com.ibm.crypto.provider.IBMJCE
  security.provider.X+1=com.ibm.security.cmskeystore.CMSProvider
  ```

  where, **X** is the next number in the order.
- Restart IBM WebSphere Application Server.
- GSKit must be installed and gsk8\lib or gsk8\lib64 depending on the platform 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:

- From the **Server hostname:port** drop-down list, select an LDAP server for the replica server.

  If you want to provide another server as replica server, which is not registered on the console server, select Use entry from below item from the **Server hostname:port** drop-down list and then enter the host name and port number for the replica server in the field in the hostname:port format. The default port is 389 for non-SSL and 636 for SSL.
- 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 288 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 303
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 420) and password policy (“Setting password policy” on page 170), 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 239 or "Adding the supplier information to a replica" on page 301 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 294.

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 291.

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 form 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 288.
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 256 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**.

Promoting replica server to master when master server is down

To promote a replica server to a master server when the master server is down, follow the steps given below:

Using Web Administration: First, using the web administration tool, login to the replica server that you want to change to a master.

1. Expand the **Replication management** category in the navigation area of the Web Administration Tool and click **Manage topology**.
2. To edit the role of an existing replica, select the required row and select **Edit subtree** from the Select Action list and then click **Go**.
3. Click the **Make server a master** button to change the role of the server to a master.
4. Click **OK** to save your settings.

Using the command line: To promote a replica server to a master you must first create an ldif record as shown below. In the ldif record, you must ensure that the value of the attribute **ibm-replicaServerId** is the same as the server-id of the

replica or consumer server. This value can be obtained from the *ibmslapd.conf* file of the replica or by issuing a rootDSE search against the replica. Then, issue an ldapadd command as shown below to add this to the replica/consumer.

```
ldapadd -h <ldaphost> -p <port> -D cn=root -w root -f promote.ldif -k

where promote.ldif file contains :
dn: cn=<any_name>,ibm-replicaGroup=default,o=sample
objectclass: top
objectclass: ibm-replicaSubentry
ibm-replicaServerId: <server id of replica or consumer server>
ibm-replicationServerIsMaster: true
cn: master
description: master server
```

After promoting a replica to master if you want to demote it again, you must remove the previously added entry.

Managing gateway servers

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 265 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 288.

On the **Additional** tab you can change:

- Replication schedules - see "Creating replication schedules" on page 303.
- 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 303 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** to display the view server panel.

The View Server panel displays the following information:

Server name
This field displays the name of the server on which the directory server instance is running. This information is displayed in the hostname:port format.

Host Name
This field displays the host name of the machine on which the directory server instance is running.

Port This field displays the nonsecure port on which the server is listening.

Server ID
This field displays the unique ID assigned to the server at the first startup of the server. This ID is used in replication topology to determine a server's role.

Role This field displays the configured role of the server in a replication topology.

Configuration mode
This field 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.

Instance name
This field displays the name of the directory server instance running on the server.

Security
This field displays the secure SSL port the server is listening on.

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 305 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 in not needed.
   - **SSL w/ EXTERNAL bind** - Specify the subject DN for the certificate and no password

   See “Adding credentials” on page 288.
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 305 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 that a server allows while replicating updates. To do this, click **Error** and enter a numeric value in the field. Otherwise, to set the maximum number of replication errors a server allows while replicating updates to a consumer as unlimited, click **Unlimited**.

  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.
- Select a value from the Restrict access to replication topology combo box to specify whether the access to replication topology is restricted or not.
- 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 288.
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. To specify whether the server supports replication conflict resolution, select a value from the **Replication conflict resolution** combo box.
5. 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).
6. Enter and confirm the credential password.
7. 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"ffor 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
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.

Note: The default timeout for any change to be completed through replication is 60 seconds. If replication updates involve large amount of changes, such as adding a large group entry, the update operation may require more than 60 seconds for the operation to finish. If any single update (add, delete, modify, or modifydn) operation through replication takes more than 60 seconds, then the supplier server times out that update operation and retries again sending the same update through replication. Inorder to extend the timeout duration for update operations in replication, you can use the IBMSLAPD_REPL_UPDATE_EXTRA_SECS environment variable. To know more about using the environment variable, see the *IBM Tivoli Directory Server Version 6.3 Problem Determination Guide*.

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 the **ldapdiff** command information in the *IBM Tivoli Directory Server Version 6.3 Command Reference* 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=sample
   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 the **ldapexop** command information in the *IBM Tivoli Directory Server Version 6.3 Command Reference* for more information.

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.

  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.

  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.

error xxxx
An error has occurred where xxxx is the id of the message that describes the error.

- **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 **ibmreplicationPendingChangeCount** 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 229 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=sample
objectclass: top
objectclass: organization
objectclass: ibm-replicationContext

dn: ibm-replicagroup=default,o=sandbox
objectclass: top
objectclass: ibm-replicaGroup
ibm-replicagrpoup: default

dn: ibm-replicaServerId=<serverid>,ibm-replicagroup=default,o=sandbox
objectclass: top
objectclass: ibm-replicaSubentry
objectclass: ibm-replicaGatewayibm-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 413.

Chapter 15. 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

The 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. IBM Tivoli Directory 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 routes read and write requests differently based on the configuration. Write requests for a single partition are directed to the single primary write server. Peer servers are not used to avoid conflicts. Read requests are routed in a round robin manner to balance the load. However, if high consistency is enabled read requests are routed to the primary write server.

The proxy server also provides support for ACL's to be defined based on groups defined on a different partition, and support for partitioning of flat namespaces. The proxy server can also be used as an LDAP-aware load balancer.

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 global admin group, local admin group with dirData authority, or the primary administrator.

Before deploying a proxy server, you must verify that all the operations required in your environment are supported. For more information, see “OIDs for supported and enabled capabilities” on page 493, “OIDs for extended operations” on page 501, and “OIDs for controls” on page 505

Note: If you specify an administrative control for any operation on proxy, the proxy server will propagate the administrative control to the backend server.

The proxy server routes new requests targeting a backend server only through a free backend connection. If there are no free backend connections available, Proxy will temporarily suspend reading requests from clients. Proxy will resume reading from clients only when the backend connection becomes free. Also, if there are pending requests from a client to a backend, any new request from the client will be routed through the same backend connection used by earlier requests.

© Copyright IBM Corp. 2002, 2010

Note: The *ibm-slapdProxyMaxPendingOpsPerClient* attribute included in the ibm-slapdProxyBackendServer objectclass can be used to configure the threshold limit for pending requests from a client connection in a backend connection. On reaching this threshold limit, requests from the client connection will not be read until the pending requests in the backend connection reduces to a value below the specified threshold limit. If this attribute is not specified, the maximum pending client operations will default to 5.

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: The server uses the same default configuration file whether it is configured as a directory server or a proxy server. However, when the server is configured as a proxy server, the configuration settings for the features that the proxy server does not support are ignored. Given below is a list of entries in the configuration file that are ignored by the proxy server:

- cn=Event Notification, cn=Configuration
- cn=Persistent Search, cn=Configuration
- cn=RDBM Backends, cn=IBM Directory, cn=Schemas, cn=Configuration
- cn=Replication, cn=configuration
- cn=Bulkload, cn=Log Management, cn=Configuration
- cn=DB2CLI, cn=Log Management, cn=Configuration

For the entry "cn=Front End, cn=configuration", environment variables set under this entry will be supported by proxy. The environment variables supported by the proxy server include the following:

Table 21. Environment variables supported by proxy server

Variable	Description
PROXY_CACHE_GAG_PW	Specifies if password caching is enabled or disabled. The proxy server has the ability to locally cache the passwords of global administrators. If password policy is enabled, caching of the Global Admin Group Member passwords is disabled. If password policy is disabled, the caching of Global Admin Group Members is enabled. PROXY_CACHE_GAG_PW environment variable can override this default behavior. PROXY_CACHE_GAG_PW set to YES will enable password caching. PROXY_CACHE_GAG_PW set to any other value will disable password caching. When the env variable is unset the default behavior is governed by the password policy setting.
PROXY_GLOBAL_GROUP_PERIOD	Specifies the interval after which the proxy interval thread wakes up. The default value for this variable is 30 seconds.
PROXY_USE_SINGLE_SENDER	Specifies if a single sender thread is used for the operations. By default this is false.

Table 21. Environment variables supported by proxy server (continued)

Variable	Description
PROXY_RECONNECT_TIME	Specifes the interval after which the proxy tries to reconnect to a backend server that has gone down. By default this is 5 seconds.
LDAP_LIB_WRITE_TIMEOUT	Specifies the time (in seconds) to wait for a socket to be write ready
FLOW_CONTROL_SLEEP_TIME	In Flow control, when there are no free backend connections available, the proxy server temporarily suspends reading from socket. It then checks periodically to see if there is a free backend connection that became available. The frequency with which this check is done is determined by the environment variable "FLOW_CONTROL_SLEEP_TIME". This must be set to an integer value and will specify in milliseconds the frequency with which the check is done by the proxy. If the environment variable is not set, it defaults to 5.

The proxy server supports some features of Tivoli Directory Server while at the same time there are some features that are not supported by proxy. The list of features that are supported by the proxy server are given below:

- Log access extended operations.
- Dynamic configuration of the supported attributes
- Server start stop
- TLS
- Unbind of a bound dn
- Dynamic trace
- Attribute type extended operation
- User type extended operation
- Auditing of source ip control
- Server administration control
- Entry check sum
- Entry uuid
- Filter acls
- Admin group delegation
- Denial of service prevention
- Admin server auditing
- Dynamic groups
- Monitor operation counts
- Monitor logging counts
- Connection monitor active workers
- Monitor tracing
- SSL Fips mode
- Modify dn as long as the entry rename does not move the entry across partitions.

- Multiple instances
- AES password encryption
- Admin password policy
- Locate entry extended operation
- Resume role extended operation
- ldap get file
- Limit number of attribute values
- Audit performance - Performance auditing is supported for proxy. The following performance info fields for each audit record are valid for proxy. The RDBM lock wait time will always be 0 for a proxy server:
  - Operation response time
  - Time spent on work Q
  - Client I/O time
- Digest MD-5 Binds
- Admin roles
- Preoperation plugins
- Global Admin Group
- Paged and Sorted Searches
- ibm-allmembers search
- Transactions

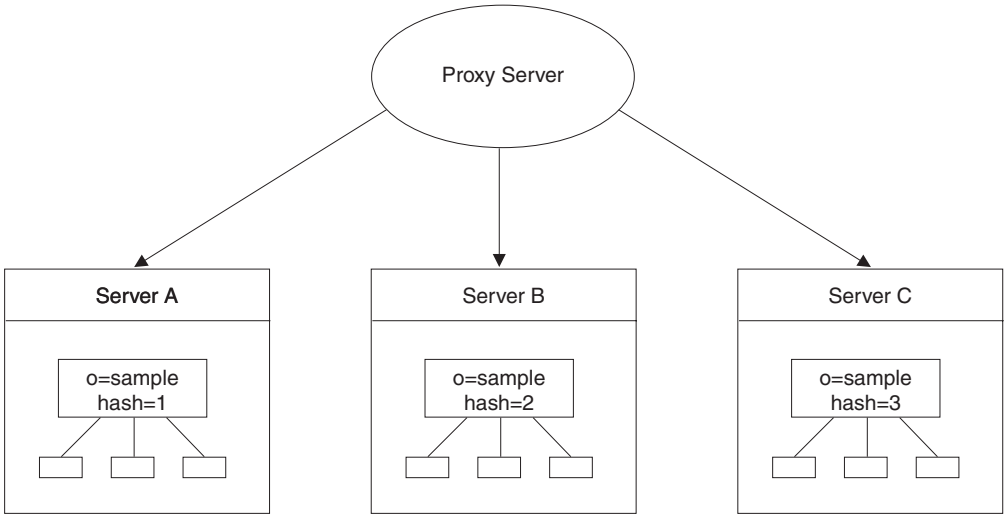
  Note: Transactions are supported but only if all the entries that are part of the transaction request reside on a single directory server.

The list of features not supported by the proxy server are given below:

- Event notification
- Replication management extended operations
- Group evaluation extended operation
- Account Status extended operation
- Subtree delete
- Proxy authorization control
- Group authorization control
- Omit group Referential integrity
- Unique Attributes
- Effective password policy
- Online backup extended operation
- Password prebind extended operation
- Password post bind extended operation
- Post Operation plugins
- Null based search

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=sample is split across three servers. This means that the proxy server is configured to hash RDN values immediately after o=sample among 3 servers, or "buckets". This also means that RDN values more than 1 away from o=sample will map to the same server as values immediately after o=sample. For example, cn=test,o=sample and cn=user1,cn=test,o=sample 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=sample. The proxy server then uses the configuration information (specifically that there are 3 partitions with a base at o=sample) 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 normalized string is used for the hash, not just the value. For example, if our split point is o=sample and this is split into three partitions, then the following occurs:

- cn=example,o=sample hashes to a single server, let's say serverA. This is determined by hashing cn=example into one of three partitions.
- dc=example, o=sample hashes to a different server, let's say serverB. This is determined by hashing dc=example.
- cn=foo,cn=example,o=sample hashes to serverA. This is because only cn=example is used for the hash algorithm. All entries beneath cn=example,o=sample resolve to the same server as cn=example,o=sample.

Note: It is essential to note that when using a 6.1 and above version of the proxy server with 6.0 backend servers, the cn=pwdpolicy subtree must be configured as a split point. However, a 6.1 and above version of proxy server using 6.1 and above backend servers should not have the cn=pwdpolicy subtree.

DN Partition plug-in

Tivoli Directory Server provides the option to load customer written partitioning function during server runtime. The existing hash algorithm that is used to partition data is statically linked by Tivoli Directory Server. However, with DN partitioning function implemented as a plug-in, the hash algorithm can be easily replaced resulting in Tivoli Directory Server being more flexible and adaptive.

The existing hash algorithm however remains as the default partitioning plug-in. It is loaded during server startup if no customized code is available. This feature incorporates an attribute called ibm-slapdDNPartitionPlugin in the objectclass ibm-slapdProxyBackend. It is a required and single-valued attribute which means that only one DN partitioning plug-in is allowed for a Proxy Server Back-end. The value of the attribute consists of a path using which a customized DN partitioning module is loaded and an initialization function using which a user provided DN partitioning function is registered.

The initialization function is called when the DN partitioning plug-in is loaded during Proxy Server startup time. By loading the dynamically loadable plug-in module, the functions defined in the module get assigned with function addresses by the loader. By executing this initialization function, the address of the partitioning function registered in the initialization function gets stored in the Proxy Server Back-end. The registered DN partitioning function, later on, is called by Proxy Router to route requests to target servers.

Note:

- The DIT that is populated by Proxy Server using one partitioning algorithm will be inaccessible by the Proxy Server using a different partitioning algorithm. Once the DIT is populated, the partition plug-in should not be changed. If you need to change the partition plug-in, then the data should be reloaded. Data loaded for Tivoli Directory Server

Version 6.0 and earlier versions will not work with a DN partitioning plug-in in Version 6.1 and above unless the default plug-in is used in Version 6.1 and above.

- It is essential to note that to you use a customized plugin, it must be set before running the ddsetup command.

Using the command line

To modify the ibm-slapdDNPartitionPlugin attribute and to add a customized plug-in, issue the following command:

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

where <*filename*> contains:

```
dn: cn=ProxyDB, cn=Proxy Backends, cn=IBM Directory, cn=Schemas, cn=Configuration
changetype: modify
replace: ibm-slapdDNPartitionPlugin
ibm-slapdDNPartitionPlugin: <customized DN partitioning plug-in library>
                            <plug-in initialization function>
```

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 ddsetup tool uses the proxy server's ibmslapd.conf file to partition entries. The data is split using the partition algorithm specified in ibm-slapdDNPartitionPlugin attribute of the configuration file.

Note: The ddsetup tool does not enforce objectclass schema check since it is designed for optimal performance.

Adding and partitioning the data

Entries are added using either the Web Administration Tool (see "Adding an entry" on page 402 for additional information or the **idsldapadd** and **idsldapmodify** command information in the *IBM Tivoli Directory Server Version 6.3 Command Reference*).

If you have an existing database with a large number of entries, you need to export the entries to an LDIF file. See the **idsdb2ldif** command information in the *IBM Tivoli Directory Server Version 6.3 Command Reference* for more information on how to do this:

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

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

2. Issue the command:

   ```
   ddsetup –I proxy -B "o=sample" -I mydata.ldif
   ```

   where

   proxy: Is the proxy server instance

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 J, "Synchronizing two-way cryptography between server instances," on page 577 for information about synchronizing directory server instances.

3. 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 will not able to retrieve the entries.

   For more information about the ddsetup utility, see the *IBM Tivoli Directory Server Command Reference*.

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=ibmpolicies 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 322 for more information.

Notes:

1. The global policies are not replicated to the proxy server.
2. Changes to cn=schema is not replicated to the proxy server.

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

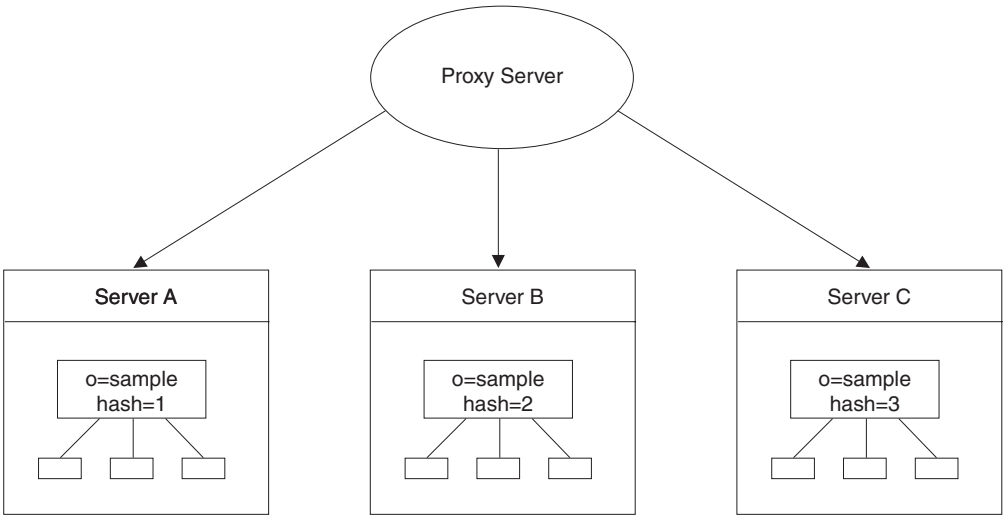
See Appendix J, "Synchronizing two-way cryptography between server instances," on page 577 for information about synchronizing directory server instances.

Partition entries

Entries that exist as the base of a partition, for example, o=sample, cannot be modified 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 proxy server

The following scenario shows how to set up the proxy server and a distributed directory with three partitions for the subtree o=sample.



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, click **Server administration** in the Web Administration navigation area and then click **Manage server properties** in the expanded list. Next, click the **Suffixes** tab.
2. Enter the Suffix DN, **o=sample**.
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 127.

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 do not have access to the audit log. Therefore, local administrators can use the audit log to monitor global administrative group member activity for security purposes.

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 can send request for schema updates through a proxy server to its backend servers. In this case, after the schema updates are applied to the proxy server, the changes will then be propagated to the backend servers. To know more about this, see “Schema updates in a distributed directory” on page 332.

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 402 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** filed.
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**.

   Note: The Current location field displays the current level of an entry in DIT tree in URL format. The suffix node in the DIT is displayed in the ldap://hostname:port format. The next level is displayed when you click a RDN from the RDN column in the Manage entries table. This displays DIT at that level. To go up at any level in the displayed DIT tree, click the required URL in the Current location field.

2. Select the radio button for cn=ibmpolicies and click **Expand**.

   Note: An expandable entry indicates that the entry has child entries. Expandable entries have a plus '+' sign next to them in the Expand column. You can click the ‘+’ sign next to the entry to view the child entries of the selected entry.

3. Select the radio button for globalGroupName=GlobalAdminGroup and from the **Select Action** drop-down menu select **Manage members** and click **Go**.

4. Specify the maximum number of members to return for a group. If you click Maximum number of members to return, you must enter a number. Otherwise, click **Unlimited**.
5. To load the members into the table, click **Load** or select Load from Select Action and click **Go**.
6. Type **cn=manager,cn=ibmpolicies** in the member field and click **Add**.
7. 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**.
8. cn=manager is displayed in the table. Click **Ok**. cn=manager is now a member of the global administration group.

Using the command line

Adding the suffix to the backend servers: For information about adding the suffix to the backend servers using command line see "Adding and removing suffixes" on page 127.

Creating and adding a user entry for membership in the global administrators group: Issue the commands:

```
idsldapadd -h <ServerA> -D <admin_dn> -w <admin_pw> -f <LDIF1>
idsldapmodify -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 **Proxy administration** .
4. Click **Manage proxy properties**.
5. Select 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=sample** and click **Add**.
8. To enable all groups processing, select the **Enable distributed groups** check box. By default, this check box is selected. The attribute ibm-slapdProxyEnableDistGroups under the ibm-slapdProxyBackend object class in the configuration file is associated with this control.

   Note: A distributed group is a group where group entries and member DN's are located in different partitions. When all group processing is disabled, the proxy server will not perform any distributed group evaluation. This is helpful if distributed directories do not contain any groups or distributed groups as the proxy server can avoid additional group processing in such cases. However, if groups are disabled at the proxy server level and the data on the backend servers contain distributed groups, the behavior is not supported and is undefined. The proxy server will be unable to detect this, so no warnings or errors will be issued.
9. To enable dynamic groups processing, select the **Enable distributed dynamic groups** check box. By default, this check box is selected. The attribute ibm-slapdProxyEnableDistDynamicGroups under the ibm-slapdProxyBackend object class in the configuration file is associated with this control.

   Note: Distributed dynamic groups are dynamic groups that are defined when some or all of the members reside in a different partition. If distributed dynamic groups do not exist, dynamic group processing can be avoided. Dynamic groups must be enabled for this setting to have an impact. By selecting or clearing the Enable dynamic group check box you can enable or disable dynamic group processing.
10. 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. Expand **Proxy administration** from the navigation area and 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:

   - Do not set the value in the **Connection pool size** field to be less than **5**.
   - Number of connections to the back-end server should be less than or equal to the number of workers configured on the back-end server.
6. Enter duration in seconds to schedule health check runs by the server.

   Note: This edit box is displayed only for proxy server with version 6.1 and above.

7. In the **Maximum pending client operations per connection** field, enter a numeric value for the maximum pending client operations per connection. The ibm-slapdProxyMaxPendingOpsPerClient attribute of the ibm-slapdProxyBackendServer objectclass is associated with this field. This attribute is used to configure the threshold limit for pending requests from a client connection in a backend connection. The default value for the ibm-slapdProxyMaxPendingOpsPerClient attribute is 5. If a value of "0" is assigned to the ibm-slapdProxyMaxPendingOpsPerClient attribute, then number of client operations per connection that is pending can be unlimited.

   Note: In the Maximum pending client operations per connection field, only positive numeric values should be assigned. If negative values are assigned, an appropriate error message will be displayed.
8. The authentication method for the back-end directory server is set to "Simple", by default. Verify that the **Enable SSL encryption** checkbox is not selected.
9. Select the **Enable health check outstanding limit** check box to check the number of outstanding health check requests the server is waiting on.
10. Enter a value for health check outstanding limit.
11. Click **Next**.
12. Specify the administrator DN, the DN of a member of the local administrator, or a member of a global admin group in the **Bind DN** field. For example, **cn=root**.
13. Specify and confirm the administration password, in the **Bind password** fields. For example, **secret**.
14. Click **Finish**.
15. Repeat steps 2 through 10 for ServerB and ServerC.
16. When you are finished, click **Close** to save your changes and return to the **Introduction** panel.
17. 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 server starts in configuration mode only. This is true unless you set up server groups. See "Server groups" on page 339.

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. Global administrative group members can send request for schema updates through a proxy server to its backend servers. To know more about schema update, see "Schema updates in a distributed directory" on page 332.

1. From the navigation area, click **Manage partition bases**.
2. On the **Partition bases** table, click **Add**.
3. Enter a split name in the **Split Name** field.

   Note: This value represents the split name provided for a split point that splits a partition base DN into partitions. The ibm-slapdProxySplitName attribute in the ibm-slapdProxyBackendSplitContainer object class is associated with this split name. The value of the ibm-slapdProxySplitName attribute must be unique within a proxy server's configuration file and must only contain alphanumeric values. For example, if a directory is split at DN "o=sample" into two partitions, the split name is associated with the o=sample split and the two partitions. To uniquely identify a split

partition you must use the ibm-slapdProxySplitName and ibm-slapdProxyPartitionIndex attributes.

4. Enter **cn=ibmpolicies** in the **Partition base DN** field.
5. Enter **1** in the **Number of partitions** field.

   Note: A value greater than **1** for cn=ibmpolicies is not supported.
6. To enable auto fail-back, select the **Auto fail-back enabled** check box.
   - To enable Auto fail-back queue, select the **Auto fail-back queue enabled** check box. The ibm-slapdProxyFailbackBasedOnQueueEnabled attribute in the ibm-slapdProxyBackendSplitContainer objectclass is associated with this control.

     When the **Auto fail-back queue enabled** check box is selected, fail-back is based on replication queue size. If this check box is not selected, then the auto fail-back queue threshold size value is ignored.
   - Enter the auto fail-back queue threshold size in the **Auto fail-back queue threshold size** field. The ibm-slapdProxyFailbackQueueThresholdSize attribute in the ibm-slapdProxyBackendSplitContainer objectclass is associated with this control.

     The default value of auto fail-back queue threshold size is 5. The auto fail-back queue threshold size denotes the size of the replication queue which determines if the replication state is stable. A value of 0 indicates that the replication queue is considered stable only if there are no pending changes. Negative values are not allowed.

   Note: If a backend server is restarted and if autofailback is enabled, the proxy server will automatically start using that backend server.
7. To enable proxy high consistency, select the **Proxy high consistency enabled** check box. For more information see "High consistency and failover when high consistency is configured" on page 337
8. Click **OK**.
9. Select the radio button for cn=ibmpolicies and click **View servers**.
10. Verify that cn=ibmpolicies is displayed in the **Partition base DN** field.
11. In the **Back-end directory servers for partition base** table, click **Add**.
12. From the **Back-end directory server** menu, select ServerA.
13. Enter **1** in the **Partition index** field.
14. From the **Server role** combo box, select a role for the back-end directory server.

    Note: The available roles that you can assign for a back-end directory server are primarywrite and any. Primary write server should be set to a master or peer server where write requests should be sent.
15. From the **Proxy tier** combo box, select a priority that you want to assign. For more information, see "Weighted Prioritization of backend servers" on page 337.
16. Click **OK**.

Dividing the data into partitions: These steps divide the data in the subtree o=sample into three partitions.

1. On the **Partition bases** table, click **Add**.
2. Enter a split name in the **Split Name** field.
3. Enter **o=sample** in the **Partition base DN** field.

4. Enter **3** in the **Number of partitions** field.
5. To enable auto fail-back, select the **Auto fail-back enabled** check box.
   - To enable Auto fail-back queue, select the **Auto fail-back queue enabled** check box. The ibm-slapdProxyFailbackBasedOnQueueEnabled attribute in the ibm-slapdProxyBackendSplitContainer objectclass is associated with this control.

     When the **Auto fail-back queue enabled** check box is selected, fail-back is based on replication queue size. If this check box is not selected, then the auto fail-back queue threshold size value is ignored.
   - Enter the auto fail-back queue threshold size in the **Auto fail-back queue threshold size** field. The ibm-slapdProxyFailbackQueueThresholdSize attribute in the ibm-slapdProxyBackendSplitContainer objectclass is associated with this control.

     The default value of auto fail-back queue threshold size is 5. The auto fail-back queue threshold size denotes the size of the replication queue which determines if the replication state is stable. A value of 0 indicates that the replication queue is considered stable only if there are no pending changes. Negative values are not allowed.
6. To enable proxy high consistency, select the **Proxy high consistency enabled** check box.
7. 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=sample and click **View servers**.
2. Verify that o=sample 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. From the **Server role** drop-down menu, select the appropriate server role.

   Note: This value represents the role of a back-end directory server in a particular partition. The ibm-slapdProxyServerRole attribute in the ibm-slapdProxyBackendSplit object class is associated with this value. The values that can be assigned to this attribute are primarywrite or any.
7. From the **Proxy tier** combo box, select a priority that you want to assign.
8. Click **OK**.
9. In the **Back-end directory servers for partition base** table, click **Add**.
10. From the **Back-end directory server** drop-down menu, select ServerB.
11. 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.
12. Click **OK**.
13. In the **Back-end directory servers for partition base** table, click **Add**.
14. From the **Back-end directory server** drop-down menu, select ServerC.

15. Ensure that **3** is displayed in the **Partition index** field.
16. From the **Server role** drop-down menu, select the appropriate server role.

    Note: This value represents the role of a back-end directory server in a particular partition. The ibm-slapdProxyServerRole attribute in the ibm-slapdProxyBackendSplit object class is associated with this value. The values that can be assigned to this attribute are primarywrite or any.
17. From the **Proxy tier** combo box, select a priority that you want to assign.
18. Click **OK**.
19. When you are finished, click **Close**.
20. Restart the proxy server for the changes to take effect.

Viewing partition bases: Do the following to view partition bases:

1. From the navigation area, click **View partition bases**.
2. Select a split from the **Select a split combo box**.
3. Click **Show partitions**. This populates the **Partition entries** table with the available partitions for the selected spilt.

Do the following to view server entries for a partition:

1. Select a partition entry from the **Partition entries table**.
2. Click **Show servers**. This populates the **Server entries** table with the server information associated with the selected partition of a split.

Viewing entry location: If you have not done so already, click **Proxy administration** in the Web Administration navigation area and then click **View entry location** in the expanded list. In this panel, the Location details table is populated with the location details of a DN entry or DN entries in a distributed directory. To populate the Location details table with information, the locate entry extended operation is called.

To view the location of a DN entry in a distributed directory, do the following:

1. To search the location of a DN entry in a distributed directory, select **Entry DN** and then enter a valid DN in the field or click the **Browse** button and specify the location of the entry DN.
2. Click the **Show entry details** button. This will populate the Location details table with the location information of the specified entry DN.
3. Click the **Close** button to navigate to the Introduction panel.

To view the locations of multiple DN entries in a distributed directory, do the following:

1. To search the locations of multiple DN entries in a distributed directory, select **Select a file containing multiple DNs**.
2. Enter the absolute path of the text file containing the multiple DN entries in the **File name** field or click the **Browse** button and specify the location of the text file that contains DN entries.
3. Click the **Submit file** button.
4. Click the **Show entry details** button to populate the Location details table with the location information of the DN entries.
5. Click the **Close** button to navigate to the Introduction panel.

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: o=sample
-
replace: ibm-slapdProxyEnableDistDynamicGroups
ibm-slapdProxyEnableDistDynamicGroups: true
-
replace: ibm-slapdProxyEnableDistGroups
ibm-slapdProxyEnableDistGroups: true
```

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-slapdProxyMaxPendingOpsPerClient: <value to be set in numerals>
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-slapdProxyMaxPendingOpsPerClient: <value to be set in numerals>
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-slapdProxyMaxPendingOpsPerClient: <value to be set in numerals>
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
ibm-slapdProxySplitName: ibmpolicysplit
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
ibm-slapdProxyBackendServerRole: any
objectclass: top
objectclass: ibm-slapdConfigEntry
objectclass: ibm-slapdProxyBackendSplit

dn: cn=o\=sample split, cn=ProxyDB, cn=Proxy Backends, cn=IBM Directory,
 cn=Schemas, cn=Configuration
cn: o=sample split
ibm-slapdProxyNumPartitions: 3
ibm-slapdProxyPartitionBase: o=sample
ibm-slapdProxySplitName: samplesplit
objectclass: top
objectclass: ibm-slapdConfigEntry
objectclass: ibm-slapdProxyBackendSplitContainer

dn: cn=split1, cn=o\=sample 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
ibm-slapdProxyBackendServerRole: any
objectclass: top
objectclass: ibm-slapdConfigEntry
objectclass: ibm-slapdProxyBackendSplit

dn: cn=split2, cn=o\=sample 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
ibm-slapdProxyBackendServerRole: any
objectclass: top
objectclass: ibm-slapdConfigEntry
objectclass: ibm-slapdProxyBackendSplit

dn: cn=split3, cn=o\=sample 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
ibm-slapdProxyBackendServerRole: any
objectclass: top
objectclass: ibm-slapdConfigEntry
objectclass: ibm-slapdProxyBackendSplit
```

Schema updates in a distributed directory

When schema updates are requested by a global administrator group member, the schema updates are first applied to the Tivoli Directory Proxy server and then the updates are propagated to its backend servers. Additionally, global admin group member can request for schema update directly on the backend servers. However, if the schema updates are requested on the proxy server by the primary administrator, a local admin group member having SchemaAdmin role, or the Master Server DN, then the schema updates are applied only to the proxy server.

To enforce schema updates to all the backend servers that the proxy server is serving to, the global policies must be synchronized. Tivoli Directory Server supports replication of schema updates to the consumer servers in a replication topology if the replication is setup for the CN=IBMPOLICIES context among all the backend servers that the proxy server is serving to. To implement this, you need to setup replication on the CN=IBMPOLICIES context amongst all the backend servers served by the proxy server. In order to ensure that the schema updates are made properly even in case of a failure of the primary write server, directory administrators should always include at least one other write server in the replication topology of the CN=IBMPOLICIES context. Proxy will re-route the schema update to the next available write server if the primary write server fails. Once the primary write server is restored back, the schema updates received in its absence will be pushed to it by the second write server.To create this setup, you must consider the following:

1. Set up a distributed directory with proxy server. See “Setting up the proxy server” on page 324.
2. Create replication topology for the cn=ibmpolicies subtree. For more information about setting up replication, see Chapter 14, “Replication,” on page 219 and see “Setting up a topology for global policies” on page 342.

   Note: If all the write server are offline, then the proxy server will return an appropriate error message to the LDAP client.

An example extract of proxy server configuration to propagate the schema updates:

```
cn=ProxyDB, cn=Proxy Backends, cn=IBM Directory, cn=Schemas, cn=Configuration
cn=Proxy Backend
ibm-slapdDNPartitionPlugin=libldapdnhash.so dnHashInit
ibm-slapdPagedResAllowNonAdmin=TRUE
ibm-slapdPagedResLmt=3
ibm-slapdPlugin=database libback-proxy.so proxy_backend_init
ibm-slapdPlugin=extendedop  libback-proxy.so  initResumeRole
ibm-slapdProxyEnableDistDynamicGroups=true
ibm-slapdProxyEnableDistGroups=true
ibm-slapdSuffix=o=sample
ibm-slapdSuffix=cn=ibmpolicies
objectclass=top
objectclass=ibm-slapdConfigEntry
objectclass=ibm-slapdProxyBackend
```

```
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-slapdProxyHealthCheckOlimit=24
ibm-slapdProxyMaxPendingOpsPerClient=5
ibm-slapdProxyPW={AES256}LM3NvpMrOFvYhTnEdmeTbw==
ibm-slapdProxyTargetURL=ldap://ServerA:389
ibm-slapdServerID=8c440640-6e1f-102e-88a8-ff9133d50edd
ibm-slapdStatusInterval=5
objectClass=top
objectClass=ibm-slapdProxyBackendServer
objectClass=ibm-slapdConfigEntry

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-slapdProxyHealthCheckOlimit=24
ibm-slapdProxyMaxPendingOpsPerClient=5
ibm-slapdProxyPW={AES256}LM3NvpMrOFvYhTnEdmeTbw==
ibm-slapdProxyTargetURL=ldap://ServerB:389
ibm-slapdServerID=aaaa01c0-6e1f-102e-8ea9-8d957fd1611f
ibm-slapdStatusInterval=5
objectClass=top
objectClass=ibm-slapdProxyBackendServer
objectClass=ibm-slapdConfigEntry

cn=cn\=ibmpolicies split, cn=ProxyDB, cn=Proxy Backends, cn=IBM Directory,
 cn=Schemas, cn=Configuration
cn=cn=ibmpolicies split
ibm-slapdProxyAutoFailBack=true
ibm-slapdProxyFailbackBasedOnQueueEnabled=true
ibm-slapdProxyFailbackQueueThreshold=5
ibm-slapdProxyHighConsistency=true
ibm-slapdProxyNumPartitions=1
ibm-slapdProxyPartitionBase=cn=ibmpolicies
ibm-slapdProxySplitName=ibmpoliciessplit
objectclass=ibm-slapdConfigEntry
objectclass=ibm-slapdProxyBackendSplitContainer
objectclass=top

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-slapdProxyBackendServerRole=primarywrite
ibm-slapdProxyPartitionIndex=1
ibm-slapdProxyTier=1
objectclass=top
objectclass=ibm-slapdConfigEntry
objectclass=ibm-slapdProxyBackendSplit

cn=split2, cn=cn\=ibmpolicies 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-slapdProxyBackendServerRole=any
ibm-slapdProxyPartitionIndex=1
```

```
ibm-slapdProxyTier=1
objectclass=top
objectclass=ibm-slapdConfigEntry
objectclass=ibm-slapdProxyBackendSplit
```

Password policy in a distributed directory

Password Policy in a distributed directory is enforced on the backend servers with some additional overhead in the proxy server. There are two kinds of user password policies: Global and multiple password policies. Multiple password policies is supported in the distributed directory environment only if all the groups, members, and policy data is local to a single partition. On the other hand, global password policy is supported, even when users and groups are distributed.

In order for the proxy server to support password policy it must be enabled on all backend servers. The proxy server will send password policy controls on all necessary requests. The majority of password policy enforcement is done locally on the backend servers, and therefore will function the same way as it does in a non-distributed environment. In some cases additional checking must be done at the proxy server level to ensure consistent password policy enforcement.

Notes:

1. If an administrator enables or disables password policy, the proxy server must be restarted.
2. The proxy server does not support effective password policy extended operation.

The proxy server uses two extended operations to enable password policy enforcement for external binds. The extended operations are Password Policy Initialize and Verify Bind Extended operation and the Password Policy Finalize and Verify Bind Extended operation. For further information about these two extended operations refer the *IBM Tivoli Directory Server Version 6.3 Programming Reference* .

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 server configured as the primary write server is the primary master. If no primary write server is configured the first master or peer server is the primary write server. If the primary write server is down, the proxy server is capable of failing over to a backup server (one of the other master or peer 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 be cryptographically synchronized. See Appendix J, “Synchronizing two-way cryptography between server instances,” on page 577 for information about synchronizing directory server instances.
- Compare operations are not load balanced.

The proxy server performs load balancing on read requests when high consistency is disabled. On the other hand, when high consistency is enabled all read and

write requests are sent to the primary write server until a failover occurs. See "High consistency and failover when high consistency is configured" on page 337 for further information

If a backend server is unavailable, the operation will error out. All subsequent operations will fail over to the next available server.

Auto failback

Tivoli Directory Server provides an option to enable and disable auto failback. When auto failback is enabled, the proxy server starts using a server as soon as it becomes available. However, when auto failback is disabled, servers must be restored using the resume role extended operation, except in the following cases where auto failback is always enabled:

Cases that always invoke auto failback and the action taken:

- All back-end servers go down in a partition.
- Action Taken:
  - If a read server is the first server to come back online, the proxy server will auto restore that server. Since read servers cannot handle write operations, the first write server to come back online will also be restored.
  - If a write server is the first server to come back online the proxy server will auto restore that write server. Since write servers can handle both read and write requests no additional servers will be automatically restored.
- All Writeable Backend Servers go down in a partition.
- Action Taken
  - The first write server to come back online will be auto restored by the proxy server.

Note:

- Autofailback can be enabled or disabled by setting the value of the attribute ibm-slapdEnableAutoFailBack to true or false.
- The default value of ibm-slapdEnableAutoFailBack is true.

Tivoli Directory Server also provides you with an option to enable failback based on a configurable replication queue size. This feature enables failback to be done automatically only when the replication queue size from the current write server to the server being failed back is less than or equal to the configured replication queue size.

To enable failback based on a configurable replication queue size using web administration tool see "Dividing the data into partitions" on page 327

To enable failback based on a configurable replication queue size using the command line, do the following:

- Set the value of ibm-slapdProxyFailbackBasedOnQueueEnabled attribute to TRUE. To do this, issue the following command:

```
ldapmodify -D <admin DN of proxy server> -w <admin PW of proxy server> \
-p <port of proxy server> -i modify.ldif

where modify.ldif contains
dn: <RDN of Backend Split Container>, cn=ProxyDB, cn=Proxy Backends,
cn=IBM Directory, cn=Schemas, cn=Configuration
```

```
changetype: modify
replace: ibm-slapdProxyFailbackBasedOnQueueEnabled
ibm-slapdProxyFailbackBasedOnQueueEnabled : <value to be set as either TRUE or FALSE>
```

- Set the value of the ibm-slapdProxyFailbackQueueThreshold attribute to a desired value. To do this, issue the following command:

```
ldapmodify -D <admin DN of proxy server> -w <admin PW of proxy server> \
-p <port of proxy server> -i modify.ldif

where modify.ldif contains
dn: <RDN of Backend Split Container>, cn=ProxyDB, cn=Proxy Backends,
cn=IBM Directory, cn=Schemas, cn=Configuration
changetype: modify
replace: ibm-slapdProxyFailbackQueueThreshold
ibm-slapdProxyFailbackQueueThreshold : <value to be set in numerals>
```

Health Check Feature

The proxy server back-end uses a thread named health check to identify the servers that are available and the servers that are down. The health check thread runs a health check by initiating a root DSE search for the ibm-slapdisconfigurationmode attribute against each of the back-end servers. If the Root DSE search against any server fails, either because the server is down or if the server is in configuration only mode, the thread begins the failover process and marks the server as unavailable. After a server is identified as unavailable, an appropriate error message is also written to the error log.

The health check feature has the ability to detect when back-end servers become unresponsive. To enable this feature, you set the ibm-slapdProxyHealthCheckOlimit attribute. The value of this attribute indicates the threshold for the number of outstanding health check requests before the proxy server determines that a back-end server is unresponsive.

Let us consider an example where the health check interval is set to 5 seconds and the olimit is set to 5. In this case, if the back-end server does not respond to the health check searches within 25-30 seconds, the proxy server will mark the back-end server as disconnected and will failover to the next available server. Subsequently, a message is also logged at this time (GLPPXY044E).

When such a message is logged it could be either because the back-end server is overloaded and the server needs performance tuning or hardware upgrade or there could be some kind of error condition in the back-end server that needs to be addressed. The proxy server updates the state of the back-end server to ready when the back-end server can successfully respond to root dse searches. If auto failback is enabled, the server is restored at this time. If auto failback is disabled an administrator can use the resume role extended operation to resume use of the server.

Note: Caution should be used when configuring the ibm-slapdProxyHealthCheckOlimit attribute. This attribute is used to specify the olimit on healthcheck. If the proxy server is under heavy load and the olimit value set is too small, the proxy may falsely report that the back-end server is unresponsive. To correct this problem, the olimit should be increased. However, the value of olimit must be atleast 3 less than the value of connection pool size.

Health Check Status Interval Configuration

The ibm-slapdStatusInterval attribute represents the time interval between health check runs scheduled by the server. This attribute is not a dynamic attribute and the default value is set to 0. The value 0 disables the health check. An administrator can modify the value of this attribute to best suit the environment.

High consistency and failover when high consistency is configured

Sometimes, high consistency is required by applications. For instance, an application may write some data then immediately perform a search to ensure the update was correct. In a high consistency environment, the proxy server does not round robin read operations. Instead, the proxy sever directs all read and write operations for a single partition to a single back-end server.

High Consistency is configurable on a per split basis. To enable high consistency, you need to set the attribute ibm-slapdProxyHighConsistency to true.

The sample entry below specifies that High consistency is enabled for the split container having partition base o=sample.

```
Sample Entry
dn: cn=o\=sample split, cn=ProxyDB, cn=Proxy Backends, cn=IBM Directory,
cn=Schemas, cn=Configuration
cn: o=sample split
ibm-slapdProxyNumPartitions: 1
ibm-slapdProxyPartitionBase: o=sample
ibm-slapdProxySplitName: samplesplit
ibm-slapdEnableAutoFailBack: true
ibm-slapdProxyHighConsistency: true
objectclass: top
objectclass: ibm-slapdConfigEntry
objectclass: ibm-slapdProxyBackendSplitContainer
```

All read and write operations within a single partition are directed to a single back-end. When the primary back-end server goes down the proxy will failover to a secondary server that is configured. All read and write operations will then be directed towards that server until the primary server is restored.

Weighted Prioritization of backend servers

The proxy server prioritizes back-end servers into 5 possible tiers. At a given time the proxy server will only use servers in one tier. When all the write servers within a tier fail the proxy server will failover to the second tier. When the second tier fails it will failover to the third tier, so on and so forth.

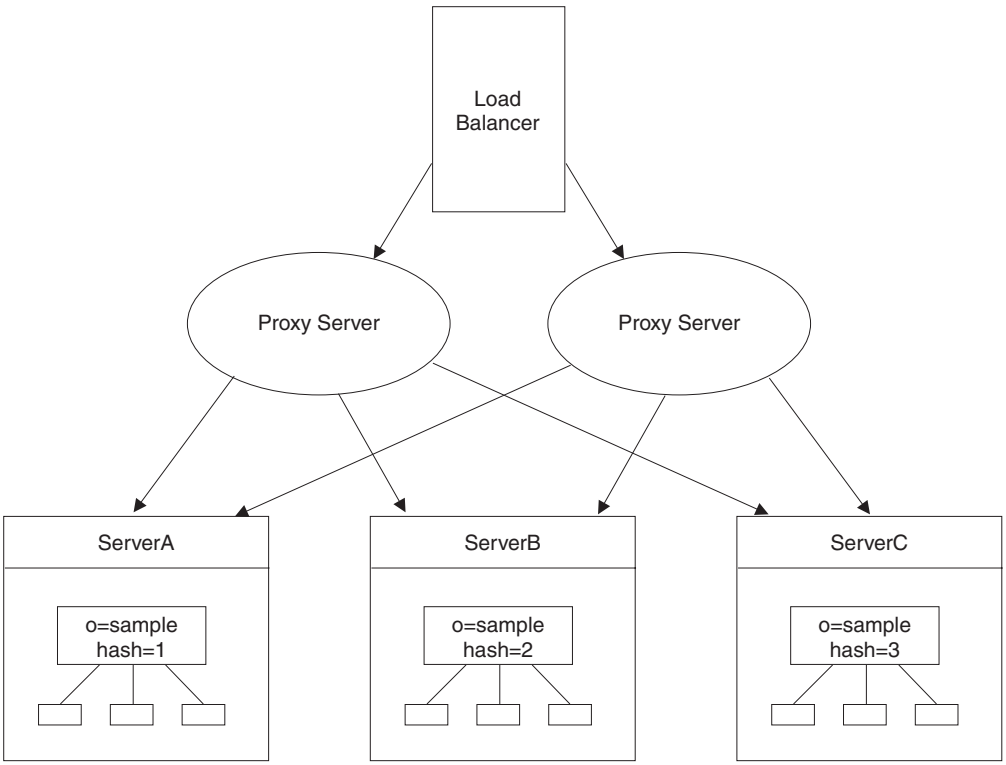
Weighted prioritization is configurable for each back-end server within a split. This is done by setting a value for the attribute ibm-slapdProxyTier. The default value for this attribute is 1 and if the attribute is not present the proxy will treat the back-end server as a tier one server. Valid values for this attribute range from 1 to 5.

During startup, all servers in all tiers will be contacted. If the administrator wants the proxy server to start up even if some of the back-end servers in different tiers are not available, then server groups can be used. For more information about server groups, see “Server groups” on page 339.

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.

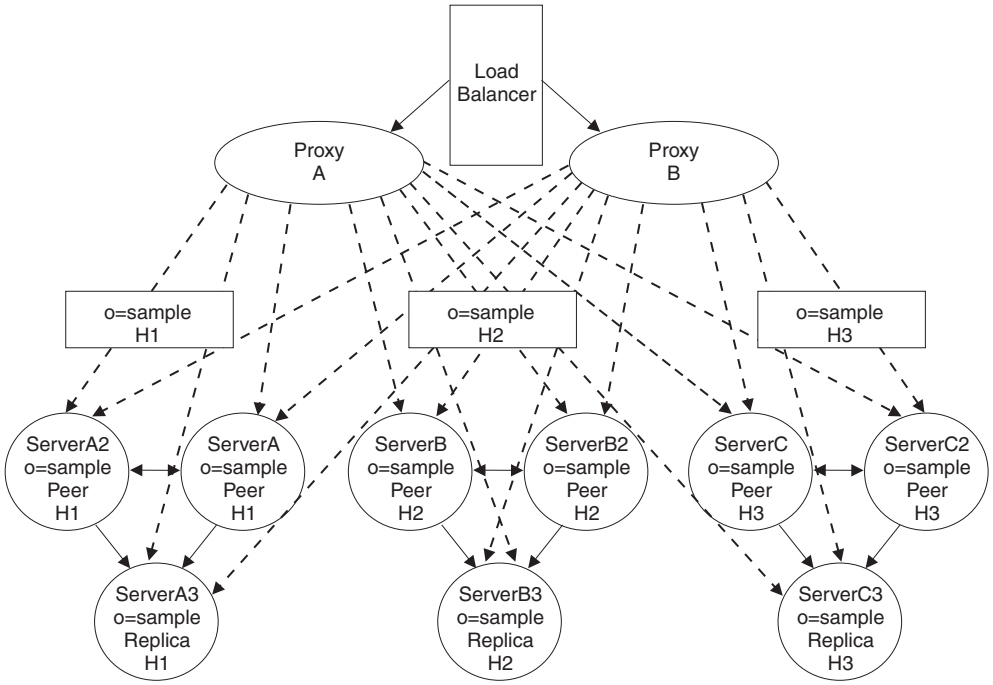
A load balancer, such as IBM WebSphere Edge Server, has a virtual host name that applications use when sending updates to the directory. The load balancer is configured to send those updates to only one server. If that server is down, or unavailable because of a network failure, the load balancer sends 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=sample 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 340
2. Create a replication topology for the data subtree. See "Setting up the replication topology" on page 341.
3. Create a second replication topology for the cn=ibmpolicies subtree. See "Setting up a topology for global policies" on page 342.
4. Set up the proxy servers. See "Setting up proxy servers" on page 343
5. Partition existing data. See "Partitioning the data" on page 343.
6. Load the data. See "Loading the partitioned data" on page 343.
7. Start replication. See "Starting replication" on page 347

For more information about setting up replication, see Chapter 14, "Replication," on page 219.

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.

The directory administrator must define the 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=sample if they currently reside on a server:

- Issue the command:

```
idsdb2ldif  -o mydata.ldif -s o=sample  -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 Appendix J, "Synchronizing two-way cryptography between server instances," on page 577 for information about cryptographic synchronization of servers.
- If all the backend servers in a distributed directory environment are not configured for the SHA-2 family of algorithms (SHA-224, SHA-256, SHA-384, and SHA-512) or salted version of the SHA-2 family of algorithms (SSHA-224, SSHA-256, SSHA-384, and SSHA-512) then data encrypted using these family of algorithms should not be added through the Proxy server. This is because if data encrypted using these family of algorithms are added to the backend servers that is not configured for these family of algorithms, then the server will assume the data to be in clear text and consequently data corruption might occur.

See the **idsdb2ldif** command information in the *IBM Tivoli Directory Server Version 6.3 Command Reference* for more information.

Setting up the replication topology

Ensure that you understand replication concepts and terms before attempting to create this scenario. See Chapter 14, "Replication," on page 219, 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=sample. Doing this makes ServerA a master server for o=sample. See "Adding a subtree" on page 286.
2. Create a set of credentials for the topology. See "Adding credentials" on page 288.
3. Add ServerA2 as a peer-master server. See "Adding a peer-master or gateway server" on page 292.
4. Add ServerA3 as a replica. Ensure that the supplier agreement with ServerA2 is selected. See "Adding a replica server" on page 294.

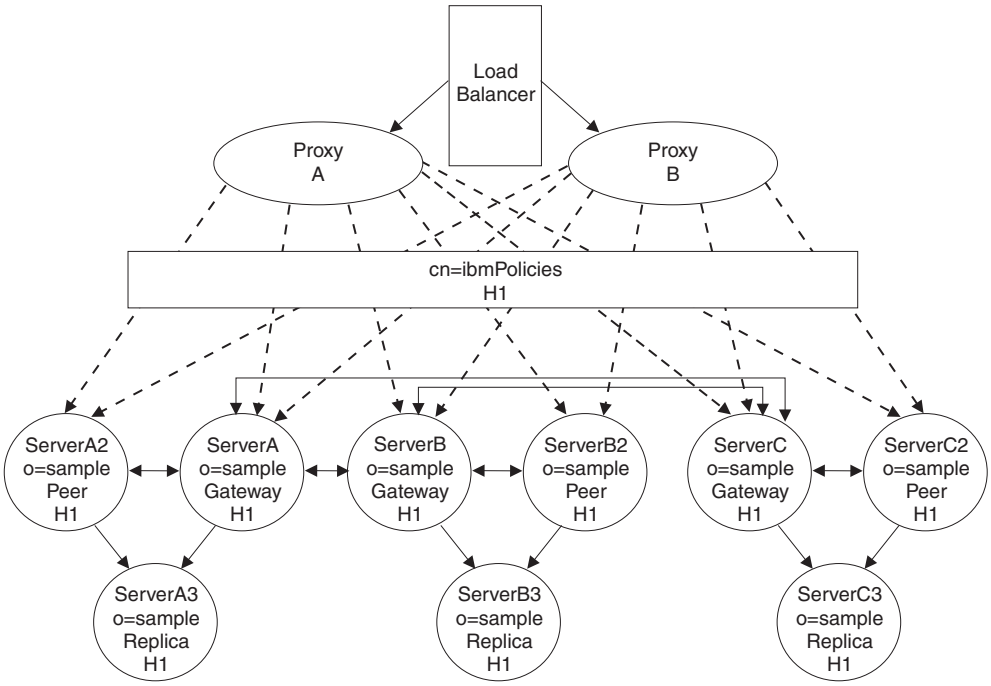
   Note: You can either log on to ServerB and ServerC to create similar topologies as you did with ServerA or continue to create the topology from ServerA. Remember that if you continue to add the topology from ServerA, you must deselect any agreements that the Web Administration Tool tries to create that 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=sample. See "Adding a peer-master or gateway server" on page 292. 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 292. 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 294.
8. Add ServerC as a master server for the subtree o=sample. See “Adding a peer-master or gateway server” on page 292. 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 292. 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 294.

For more information about setting up replication, see Chapter 14, “Replication,” on page 219.

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=sample 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 265 and “Managing gateway servers” on page 298 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 proxy servers

1. Set up a proxy server, Proxy A:

   Follow the directions is "Setting up the proxy server" on page 324 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 IBM WebSphere Edge Server.

Partitioning the data

To partition the data contained in the mydata.ldif file you created for the subtree o=sample, issue the following command:

```
ddsetup –I ProxyA -B "o=sample" –i mydata.ldif

where
ProxyA:Is the proxy server instance
```

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

Monitor Search

Administrators can use monitor search to determine the current status of the proxy server. Monitor search does not actively query for status but it simply reports the current status that is available to the proxy server. This implies that if a back-end server is down and the proxy server has not discovered it yet then it will not be reported in the search result. A monitor search for "cn=partitions, cn=proxy, cn=monitor" will return one entry for each split point, partition, and server in each partition.

Note:

- On a proxy server the cn=monitor search will show operations as completed before they are really completed. If operation counts are needed to detect actual completed operations, the cn=proxy,cn=monitor search must be used.
- In a proxy server environment a single request from a client can map to multiple different kinds of requests in the proxy environment. For example, a bind maps to a compare, search, and a series of extended operations to evaluate group membership.

An example of monitor search for the searchbase "cn=partitions, cn=proxy, cn=monitor" is given below:

```
idsldapsearch -D <adminDN> -w <adminpw> -h <servername> -p <portnumber>
–b cn=partitions,cn=proxy,cn=monitor -s base objectclass=*
```

This command returns the following information:

```
Split Point Entry:
ibm-slapdProxySplitName= <configured name>, cn=partitions, cn=proxy, cn=monitor
ibm-slapdProxyPartitionBase= <configured base>
ibm-slapdProxyHighConsistencyEnabled = <true|false>
ibm-slapdProxyCurrentTier = <tier number> the current tier that the proxy
server uses to process operations.

Partition Entry:
ibm-slapdProxyPartitionIndex= <index value>,ibm-slapdProxySplitName= <configured name>,
cn=partitions,cn=proxy, cn=monitor
ibm-slapdProxyPartitionStatus : (active, readonly, unavailable)
ibm-slapdProxyPartitionIndex= <index value>

Server Entry:
ibm-slapdPort= <port> + ibm-slapdProxyBackendServerName= <server URL>,
ibm-slapdProxyPartitionIndex= <index value> ibm-slapdProxySplitName= <configured name>,
cn=partitions, cn=proxy, cn=monitor
ibm-slapdServerStatus: (active, unavailable)
ibm-slapdProxyCurrentServerRole: (primarywriteserver, readonlyserver, writeserver, notactive)
ibm-slapdProxyConfiguredRole: (primarywriteserver, readonlyserver, writeserver)
ibm-slapdProxyNumberofActiveConnections: <connection count>
```

where

- ibm-slapdProxyPartitionStatus:
  - active: If atleast one write server is active.
  - readonly: If no write servers are active, but atleast one read server is active.

  - unavailable: No servers are active in the partition.
- ibm-slapdServerStatus:
  - active: The server is up and the proxy server has established connections to the server.
  - unavailable: The server is either started in configuration mode, or the proxy server is unable to establish a connection to the server with the proper authority.
- ibm-slapdProxyCurrentRole:
  - primarywriteserver: The server is active and receiving all the write requests. If high consistency is enabled the server is also receiving all the read requests.
  - readonlyserver: The server is active and available for read only requests. the server will only be used if high consistency is disabled, or all the write servers are down.
  - writeserver The server is active and available. If high consistency is enabled, this server will not be used until failover. If high consistency is disabled, this server will be used as a read server until a failover situation.
  - notactive: This means that the server is currently not being used in this partition. This can mean one of two things: the server is unreachable, or the server is up, but has not been restored in this partition.
- ibm-slapdProxyConfiguredRole: This is the role that the server was configured as. If no roles were specifically configured this value is set based on the proxy server's own discovery algorithm at start up.
- ibm-slapdProxyNumberofActiveConnections: This is the actual number of connections that are open to the backend server.

Note: If the connection is secure, ibm-slapdSecurePort attribute will be used instead of ibm-slapdPort.

A monitor search for cn=proxy,cn=monitor will provide counters for each kind of operation requested and completed by the proxy back-end. The filter supported by this search is objectclass=*. The counters related to all the back-end servers configured in the proxy server is given as an output of the monitor search. Following counters are returned by the proxy backend monitor search:

- ops_requested – The number of operations requested by the Proxy Backend.
- ops_completed - The number of operations completed by the Proxy Backend.
- search_requested - The number of search operations requested by the Proxy Backend.
- search_completed - The number of search operations completed by the Proxy Backend.
- binds_requested - The number of bind operations requested by the Proxy Backend.
- binds_completed - The number of bind operations completed by the Proxy Backend.
- unbinds_requested - The number of unbind operations requested by the Proxy Backend.
- unbinds_completed - The number of unbind operations completed by the Proxy Backend.
- adds_requested - The number of add operations requested by the Proxy Backend.
- adds_completed - The number of add operations completed by the Proxy Backend.

- deletes_requested - The number of delete operations requested by the Proxy Backend.
- deletes_completed - The number of delete operations completed by the Proxy Backend.
- modrdns_requested - The number of modrdn operations requested by the Proxy Backend.
- modrdns_completed - The number of modrdn operations completed by the Proxy Backend.
- modifies_requested - The number of modify operations requested by the Proxy Backend.
- modifies_completed - The number of modify operations completed by the Proxy Backend.
- compares_requested - The number of compare operations requested by the Proxy Backend.
- compares_completed - The number of compare operations completed by the Proxy Backend.
- abandons_requested - The number of abandons operations requested by the Proxy Backend.
- abandons_completed - The number of abandons operations completed by the Proxy Backend.
- extops_requested - The number of extended operations requested by the Proxy Backend.
- extops_completed - The number of extended operations completed by the Proxy Backend.
- unknownops_requested - The number of unknown operations requested by the Proxy Backend.
- unknownops_completed - The number of unknown operations completed by the Proxy Backend.
- total_connections - The number of connections between the proxy backend and backend servers configured for the proxy server.
- total_ssl_connections - The number of ssl connections between the proxy backend and backend servers configured for the proxy server.
- used_connections - The number of used connections between the proxy backend and backend servers configured for the proxy server.
- used_ssl_connections - The number of used ssl connections between the proxy backend and backend servers configured for the proxy server.
- total_result_sent – The number of results sent by the proxy backend to the client since the proxy server was started.
- total_entries_sent - The number of entries sent by the proxy backend to the client since the proxy server was started.
- total_success_result_sent - The number of success results sent by the proxy backend to the client since the proxy server was started.
- total_failed_result_sent - The number of failed results sent by the proxy backend to the client since the proxy server was started.
- total_references_sent - The number of references sent by the proxy backend to the client since the proxy server was started (related to referrals).
- transactions_requested - The number of transaction operations requested by the Proxy Backend.
- transactions_completed - The number of transaction operations completed by the Proxy Backend.

- transaction_prepare_requested - The number of prepare transaction operations requested by the Proxy Backend.
- transaction_prepare_completed - The number of prepare transaction operations completed by the Proxy Backend.
- transaction_commit_requested - The number of commit transaction operations requested by the Proxy Backend.
- transaction_commited - The number of commit transaction operations completed by the Proxy Backend.
- transaction_rollback_requested - The number of rollback transaction operations requested by the Proxy Backend.
- transaction_rollbacked - The number of rollback transaction operations completed by the Proxy Backend.

Transactions in proxy server

Transactions enable an application to group a set of entry updates. The proxy server can process concurrent transaction requests where all operations target a single backend server.

The proxy server utilizes the backend servers' s transaction functionality to complete the transaction requests. Transactions are enabled on the proxy server only if they are enabled on the backend servers. A message is logged at startup if the backend servers have transactions enabled. In addition, the prepare transaction extended operation is enabled only if it is enabled on the backend servers. A message is logged at start up if the backend servers do not support the prepare transaction request.

For best results, the maximum number of transactions configured on the proxy server must be at least one less than the number of connections available to each backend server. For example, if the connection pool value is set to 10, then the maximum number of transactions should be set to 9 or less. Also, if the backend servers have a small timeout value, then the proxy server's transactions will get rolled back on the smaller transaction timeout value.

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 287 and “Managing queues” on page 305 for information on how to do those tasks.

Chapter 16. Directory Server backup and restore

Tivoli Directory Server provides methods for backing up and restoring directory server instance information. There are methods that back up the complete information for a directory server instance, and methods that back up only the data in the database. Use the information in "Methods that back up complete directory server instance information" and "Methods that back up database information only" on page 350 to help you choose a backup and restore method.

Methods that back up complete directory server instance information

Tivoli Directory Server provides two mechanisms for backing up and restoring complete directory server instance information: basic and enhanced. These mechanisms can back up not only the directory server instance data (stored in a DB2 database), but also the associated configuration and schema files for the directory server instance.

You can find information about the basic method in the *IBM Tivoli Directory Server Version 6.3 Installation and Configuration Guide* under the sections *Creating a new instance for which you specify all settings* and *Backing up the directory server instance*. Also, you can find information about the basic method in the *IBM Tivoli Directory Server Version 6.3 Command Reference* (see the information about the **idsdbback** and **idsdbrestore** commands).

Information about the enhanced method is contained in this chapter and in the *IBM Tivoli Directory Server Version 6.3 Command Reference* (see the information about the **ldapexop** utility with the extended operations option **-op backuprestore**).

Both methods provide the option to perform *online* or *offline* backups. (Online backups can be performed while the server is running or stopped; offline backups must be performed while the server is stopped.) The backups are always stored on the server where they are taken. However, there are differences in where and how you can request the backup.

With either of these two methods, the backups do not back up the following files, which you must back up separately:

- idsinstances.ldif
- SSL related files: keys, key stash files, CRL files
- Tivoli Directory Integrator solution files

After investigating these methods, choose one and use it exclusively. Do not mix the two methods.

The following table compares the two methods.

Table 22. Comparison of basic and enhanced backup and restore methods

Feature	Basic method	Enhanced method
Request from	Local server	Remote or local server
Interface used	Instance Administration Tool or the **idsdbback** and **idsdbrestore** commands	Web Administration Tool or **ldapexop** utility

© Copyright IBM Corp. 2002, 2010

Table 22. Comparison of basic and enhanced backup and restore methods (continued)

Feature	Basic method	Enhanced method
Backup location	Can be taken to a different location each time; overwrites the previous backup only if the backup is performed to the same location.	Provides a way to configure the backup location and method that will be used for all the backups requested through this mechanism.
Store one or multiple backups	Multiple backups	Stores only one backup at a time and overwrites previous backups when the new backup is successfully taken
Restores	Administrator can choose from any backup location on the disk	Allows a restore only from the most current backup taken
Scheduling	One time request that backs up or restores to a specific location specified at the time of the backup	Provides the option to schedule backups one time, daily or weekly.
Online or offline	Can perform online or offline backups.	Can perform online or offline backups.
Backs up directory server data and associated configuration and schema files	Provides option to back up only configuration files	Backs up data and associated configuration and schema files
Administrator management	More required. Administrators must better manage their disk space.	Less required. Only one backup location.
Backs up and restores DB2 parameters	Backs up and restores DB2 configuration parameters and database optimization parameters	Backs up and restores DB2 configuration parameters and database optimization parameters

Methods that back up database information only

As an alternative to Tivoli Directory Server complete backup and restore mechanisms, there are two other methods that you can use to back up and restore only the directory server instance data that is stored in the DB2 database. These methods back up the DB2 data but not Tivoli Directory Server-specific configurations such as the schema. One method also preserves DB2 configurations. The two methods are described in the following list:

- You can use the Tivoli Directory Server LDAP LDIF export and import commands, **idsdb2ldif** and **idsldif2db**, to export the data into an LDIF file and restore it from the LDIF file. See the section *Importing LDIF data with the Configuration Tool* in the *IBM Tivoli Directory Server Version 6.3 Installation and Configuration guide* for information about using the Configuration Tool, or the *IBM Tivoli Directory Server Version 6.3 Command Reference* for information about the commands. These commands do not preserve DB2 configurations. They work across all dissimilar hardware platforms, but they are relatively slow.
- You can use DB2 backup and restore commands to back up and restore the data. This method preserves the DB2 configurations and is fast. This method works across some dissimilar hardware and platforms, depending on whether DB2 supports it. See Appendix M, "IBM Tivoli Directory Server backup and restore," on page 591 for more information.

For best results, use either the basic or enhanced method described in "Methods that back up complete directory server instance information" on page 349 unless there are special circumstances you must address, such as backing up and restoring data across dissimilar hardware platforms.

Enhanced backup

The enhanced backup method enables you to back up the directory server instance data and the associated configuration and schema files for the directory server instance. The enhanced backup method provides options to perform both online and offline backups.

Note:

- Online Backup configuration can be done either during the initial database configuration or from the database backup tool.
- If online backup is configured in the server's configuration file and the administrator changes the backup location path, the server needs to be stopped for the first backup following the change. Subsequent backups can be performed with the server online.
- For servers configured for online backups it is important to schedule recurring backups or logs will grow too large for the file system.
- Removal of online backup configuration can be done using the database configuration tool.
- Backed up database and server files are replaced after each successful backup. However, if the backup operation fails, the previous backup is still available.
- Proxy servers must be backed up using the basic method. For more information, see *IBM Tivoli Directory Server Version 6.3 Installation and Configuration Guide*.
- Changelog data can be backed up if desired.
- To backup or restore to multiple paths you must use the Instance administration tool. For more information, see *IBM Tivoli Directory Server Version 6.3 Installation and Configuration Guide*.
- For all backup operations, ensure that the administration server is running.

To configure the directory server for backup and restore, use one of the following methods:

Using Web Administration

If you have not done so already, click **Server administration** in the Web administration navigation area, and then click **Manage backup/restore** in the expanded list. On the Manage backup/restore panel, the **Backup/Restore status** tab is selected by default.

The Backup/Restore status tab displays the following information:

Backup enabled
Specifies whether backup is enabled for a directory server instance or not. The value of this field can be "true" or "false". The backupenabled attribute is associated with this field.

Backup change log enabled
Specifies whether backup is configured for change log or not. The value of this field can be "true" or "false". The backupchangelog attribute is associated with this field.

Backup type
Specifies whether an online backup or offline backup is configured for a directory server instance. If the backup type is online, the value of this field will be "ONLINE". If the backup type is offline, the value of this field will be "OFFLINE".

Backup frequency
Specifies the frequency at which the backup will be performed for a directory server instance. The value of this field can be "one time", "daily", "weekly", or "one time and recurring" depending on the type of schedule that a user had configured on the **Schedule directory server backup** tab. If a user does not choose any options, then "none" is displayed in the field.

Backup status
Specifies the current status of the backup. The status of the backup can be one of the following:

- SCHEDULED
- NOT SCHEDULED
- BACKUP IN PROGRESS

The backupstatus attribute is associated with this field.

Previous successful backup
Specifies the date and time when the last successful backup was performed in YYYY-MM-DD-hh:mm format. If backup was never been done for a directory server instance, then "none" is displayed. The backuplastdone attribute is associated with this field.

Previous backup location
Specifies the configured path where the last backup was performed. If backup is not configured for a directory server instance, then "none" is displayed in this field.

Next scheduled backup
Specifies the date and time when the next backup is scheduled in YYYY-MM-DD-hh:mm format. If backup is not configured for a directory server instance, then "none" is displayed in this field. The backupnextscheduled attribute is associated with this field.

Next backup location
Specifies the location where the next backup is to be performed. If backup is not configured for a directory server instance, then "none" is displayed in this field.

Restore status
Specifies the current restore status. The restore status can be one of the following:

- RESTORE IN PROGRESS
- RESTORE COMPLETED yyyy-MM-dd-hh:mm
- none

The restorestatus attribute is associated with this field.

You can click Refresh to refresh the information on this panel

Configure directory server backup

If you have not done so already, click **Server administration** in the Web administration navigation area, and then click **Manage backup/restore** in the expanded list. Click the **Configure directory server backup** tab.

Note: If the directory server is not running and the Web administration tool is connected to the administration server, then a message such as "Connected to administration server. Not all values are available." will be displayed.

On this tab, you can do the following:

- Enable or disable backup for a directory server
- Enable or disable backup for change log
- Set the backup type
- Set a path for backup and restore

Directory server backup can be configured by the following users :

- Primary directory administrator
- Local administration group member having all of the following roles: DirDataAdmin, ServerStartStopAdmin, ServerConfigGroupMember, and SchemaAdmin

To configure backup and restore settings for a directory server instance, do the following:

1. Select the **Enable backup of directory server** check box to enable backup for the selected directory server instance.
2. Select the **Enable backup of changelog** check box to enable backup for the changelog database.

   Note: This check box will be available only if the changelog is configured for the directory server instance.
3. To specify a backup type, select one of the following:
   - Click **Online backup** to enable online backup for a directory server instance.
   - Click **Offline backup** to enable offline backup for a directory server instance.

   Note: Online backups can be performed while the server is running or stopped and offline backups must be performed while the server is stopped.
4. Specify a path for backup and restore operations in the **Backup/Restore location** field. If the specified location does not exist on the computer, then the path will be created.

   Note:
   - The instance owner must have write permission on the specified backup location.
   - When specifying a path for directory server backup, you must ensure that the specified path has adequate space for two directory backups since the previous backup is retained until the current backup is completed successfully. If online backups are scheduled, you must ensure that there is adequate space for up to a week's worth of inactive archive log files. If online backups are not scheduled, a directory administrator must monitor the space used by inactive logs and remove them periodically.
5. When you are finished, do one of the following:
   - Click **OK** to apply your changes and exit this panel.

- Click **Apply** to apply your changes and stay on this panel.
- Click **Cancel** to exit this panel without making any changes.

Perform directory server backup

If you have not done so already, click **Server administration** in the Web administration navigation area, and then click **Manage backup/restore** in the expanded list. Click the **Perform directory server backup** tab.

This tab will be available only if the server returns the following:

- Server capability OID 1.3.18.0.2.32.87 on the root DSE search, which suggests that the server is capable of configuring the backup and restore configuration entry.
- The ServerBackupRestore LDAP extended operation OID 1.3.18.0.2.12.81 in the supportedextension for the administration server on the root DSE search.

The administration server uses the idsdbback command to process backup requests. The idsdbback command is used to backup directory server instance's data and configuration files. If you are performing backup for a directory server instance for the first time, the directory server instance must be stopped before performing backup. For a first time online backup, you must stop the directory server if the database is not configured for online backup. This is because online backups require changes to the database configuration. After the initial backup, the directory server instance state can be running or stopped when performing an online backup. However, for an offline backup the directory server instance must be stopped. The web administration tool will guide you through stopping the server.

The backup operation can only be performed by the following users :

- Primary directory administrator
- Local administration group member having all of the following roles: DirDataAdmin, ServerStartStopAdmin, ServerConfigGroupMember, and SchemaAdmin

The **Perform directory server backup** tab will not be displayed to any other user.

The Perform directory server backup tab displays the following information:

Backup type
Specifies the type of backup configured for a directory server instance. Depending on the type of backup configured, the value of this field can be "ONLINE" or "OFFLINE". The backuponline attribute is associated with this field.

Backup status
Specifies the current status of the backup. The status of the backup can be one of the following:

- SCHEDULED
- NOT SCHEDULED
- BACKUP IN PROGRESS

The backupstatus attribute is associated with this field.

Previous successful backup
Specifies the date and time when the last successful backup was performed in YYYY-MM-DD-hh:mm format.

Backup location

Specifies the path where the backup would be stored. The backuplocation attribute is associated with this field. If a backup location is not configured, then the Perform directory server backup tab will not be available.

Do the following:

- To take a first time online backup when database is not configured for online backup, click **Stop server and backup now**.
- To take online backup of a directory server instance when database is configured for online backup, click **Backup now**.
- To take an offline backup while server is running, click **Stop server and backup now**.
- To take an offline backup while server is stopped, click **Backup now**.
- To view logs related to the backup operation, click **View logs**.

Note: The web administration tool will only display one of the above mentioned options depending on the current state.

You can click **Refresh** to refresh the information on this panel.

Schedule directory server backup

If you have not done so already, click **Server administration** in the Web administration navigation area, and then click **Manage backup/restore** in the expanded list. Click the **Schedule directory server backup** tab.

Note: Scheduling an offline backup will cause the server to stop the backup to be taken and the server will be restarted. However, scheduled online backups do not stop the server.

On this tab, you can configure a schedule to perform the backup operation for a directory server instance. To configure the scheduling of backup, do the following:

1. To take backup once for a directory server, select the check box under the section One time and specify a date and time. You can select a date using the calendar icon.

   Note:

   - User can select One time, Recurring, or both of the options to schedule backup operations.
   - If the backup type is online but the database is not configured for online backup, then the controls under the sections One time and Recurring will be disabled and scheduling of backup will not be allowed until you perform the first backup using the **Perform directory server backup** tab.
2. To take directory server backup after a specific interval of time in a recurring manner, select the check box under the section Recurring and specify the duration. Here, the duration can be daily or a day of the week.
3. When you are finished, do one of the following:
   - Click **OK** to apply your changes and exit this panel.
   - Click **Apply** to apply your changes and stay on this panel.
   - Click **Cancel** to exit this panel without making any changes.

Perform directory server restore

If you have not done so already, click **Server administration** in the Web administration navigation area, and then click **Manage backup/restore** in the expanded list. Click the **Perform directory server restore** tab.

The administration server uses the idsdbrestore command to process restore requests. This command restores directory server instance's data and configuration files. To perform restore operation, the directory server instance must be stopped.

The restore operation can only be performed by the following users:

- Primary directory administrator
- Local administration group member having all of the following roles: DirDataAdmin, ServerStartStopAdmin, ServerConfigGroupMember, and SchemaAdmin

The **Perform directory server restore** tab will not be displayed to any other user.

The Perform directory server restore tab displays the following information:

Restore status
Specifies the status of the restore operation.

Restore location
Specifies the configured path from where the backup should be restored. The backuplocation attribute is associated with this field.

Restore from backup
Specifies the date and time in yyyy-MM-dd-hh:mm format of the previous backup.

Do the following:

- To restore a directory server instance while server is running, click **Stop server and restore now**.
- To restore a directory server instance while server is stopped, click **Restore now**.
- To view logs related to the restore operation, click **View logs**.

Note: The web administration tool will display only one of the above mentioned options depending on the server state.

You can click **Refresh** to refresh the information on this panel.

Using the command line

To display backup status, issue the following command:

```
idsldapsearch -h <ldaphost> -p <admin port> -D <binddn> -w <password>
-s base -b cn=backup,cn=monitor objectclass=*
```

To configure backup, issue the following command:

```
idsldapmodify -h <ldaphost> -p <ldap port> -D <binddn> -w <password> –i backup.ldif

Where backup.ldif contains:

dn: cn=RDBM Backup, cn=Configuration
ibm-slapdBackupAt: 2008-04-14-16:55
ibm-slapdBackupChangelog: <value to be set as either TRUE or FALSE>
ibm-slapdBackupEnabled: <value to be set as either TRUE or FALSE>
```

```
ibm-slapdBackupEvery: 6-01:17
ibm-slapdBackupLocation: <specify the desired backup location>
ibm-slapdBackupOnline: <value to be set as either TRUE or FALSE>
```

In this example, one time as well as recurring backups is scheduled by setting the ibm-slapdBackupAt and the ibm-slapdBackupEvery attribute respectively. The attributes ibm-slapdBackupAt and the ibm-slapdBackupEvery must be set in the following format:

- ibm-slapdBackupAt : <YYYY-MM-DD-hh:mm>
- ibm-slapdBackupEvery: <D-hh:mm> , where 0=Sunday, 6=Saturday, and 7=Every day

Note: If backups are configured to be done online, the first backup must be performed with the directory server offline. You must not schedule an online backup before performing the first backup offline or the backup will fail.

To notify the admin server about the changes in the server configuration, issue the following command:

```
idsldapexop -h <ldaphost> -p <admin port> -D <binddn> -w <password>
–op readconfig -scope subtree 'CN=RDBM BACKUP, CN=CONFIGURATION
```

To initiate backup of a directory server instance request remotely, issue the following command:

```
idsldapexop -h <ldaphost> -p <admin port> -D <binddn> -w <password>
-op backuprestore -action backup
```

Note: The type of backup and the backup location are determined by how the server is configured for backups. The configuration must be done before issuing the idsldapexop command.

To restore a directory server instance remotely, issue the following command:

```
idsldapexop -h <ldaphost> -p <ldap port> -D <binddn> -w <password>
-op backuprestore -action restore
```

Note: For more information about backing up and restoring directory server instance information using **ldapexop** utility with the extended operations option **-op backuprestore**, see *IBM Tivoli Directory Server Version 6.3 Command Reference*.

Chapter 17. Logging Utilities

IBM Tivoli Directory Server provides several logging utilities that can be viewed either through the Web Administration Tool or the system command line.

- "Modifying global log settings" on page 361
- "Modifying administration server log settings" on page 362
- "Enabling the administration server audit log and modifying administration audit log settings" on page 364
- "Enabling the server audit log and modifying server audit log settings" on page 368
- "Modifying bulkload log settings" on page 378
- "Modifying tools log settings" on page 380
- "Modifying DB2 log settings" on page 381
- "Modifying lost and found log settings" on page 382
- "Modifying the server log" on page 384

Notes:

1. In the Web Administration Tool the **Logfiles** link in each task title bar accesses the Web Administration console log files. 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.

© Copyright IBM Corp. 2002, 2010

Log management tool

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 **idslogmgmt** command information in the *IBM Tivoli Directory Server Version 6.3 Command Reference* for more information.

Note: The log management tool requires that you have IBM Tivoli Directory Integrator installed.

Specifying custom location for an instance's idslogmgmt.log file

By default, the idslogmgmt tool when run as a instance owner with the -I <instance> option or as root without -I option logs information to the following file:

On UNIX-based systems

```
/var/idsldap/V6.3/idslogmgmt.log
```

On Windows systems

```
<TDS install_directory>\var\idslogmgmt.log
```

However, when the tool is run by a user with root permission it overrides the file permissions of the idslogmgmt.log file to read-only for other users, thereby not allowing an instance owner to log information in the idslogmgmt.log file specific to an instance.

To specify a custom location for the idslogmgmt.log file to log information when using the `idslogmgmt -I <instance>` command, user must set the environment variable IDSLMG_LOG_PATH. Do the following to specify a custom location for the idslogmgmt.log file:

1. Create a directory where you want to store the idslogmgmt file. Ensure that the instance owner has the required access rights on the directory.
2. Log in to the system using the instance owner's credentials.

   Note: On UNIX, if a user has logged with credentials other than instance owner, run `su - <instance_owner>`.
3. Set the environment variable, IDSLMG_LOG_PATH, and export the variable. For example:

   On UNIX-based systems

   ```
   export IDSLMG_LOG_PATH=/directoryForLog
   ```

   On Windows systems

   ```
   set IDSLMG_LOG_PATH=C:\directoryForLog
   ```
4. From the same console, start the idslogmgmt tool. For example:

   ```
   idslogmgmt -I <instance_name>
   ```

   This will create the idslogmgmt.log file under the location specified in the IDSLMG_LOG_PATH environment variable. However, Tivoli Directory Integrator AssemblyLine startup messages will be logged in the <instance_home>/idsslapd-<instance>/etc/logmgmt/idslogmgmt.log file.

If the environment variable is not set and the idslogmgmt command is run with the -I <instance> option, the information will be logged in the <instance_home>/idsslapd-<instance>/etc/logmgmt/idslogmgmt.log file.

Attention: Tivoli Directory Server may crash if the size of any log file exceeds the size of the system file size limit. Such a situation may typically occur when tracing is enabled on the server.

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/V6.3/idslogmgmt.log
```

and to the following file on Windows:

```
<TDS 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 global log settings

If you have the Log Management Tool and IBM Tivoli Directory Integrator installed, then 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 to keep only three archived logs for each log, you can set the maximum log archives value to three for all the logs using the global log settings. The global log settings apply to all logs. The global log settings apply to all log management entries unless they are overridden by specifying the settings explicitly for individual log entries.

To edit Global log settings, use one of the following methods:

Using the Web Administration Tool

1. Select **Global log settings** and click the **Edit settings** button or select **Edit settings** from the **Select Action** drop-down list and click **Go**.
2. Specify the threshold size for the log in MB under **Log size threshold (MB)**. If you want to specify a size limit in MB, select the option and specify a numeric value in the field. Otherwise, select **Unlimited**.
3. Specify the maximum number of logs to be archived. If you want to specify the maximum number of logs to be archived, select the option and specify a numeric value in the field. If you don't want to archive logs, select **No archives**. To set it to unlimited, select **Unlimited**.
4. Specify the path name for logs to be archived. If you want to specify a path name, select the option and enter the absolute path name for logs to be archived. To specify the archive path same as that of log file, select **Same directory as of log file**.
5. Specify the frequency between two cycles of Event by selecting an item from the **Select frequency** check box.
6. Specify the start date and start time for Event in the Starting on fields. You can also click the calendar icon to specify the start date. The start time should be in the following format: 12:30:00 PM.
7. After you have finished, do one of the following:
   - Click **Next** to continue with the configuring of log settings.
   - Click **Finish** to save the changes and return to the Modify log settings panel.
   - Click **Cancel** to discard changes made on this panel and to navigate to the Modify log settings panel.

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 server log settings

An administration server is a limited LDAP server that accepts extended operations to stop, start, and restart the LDAP server. The administration server log (idsdiradm.log is the default file name) enables you to view status and errors encountered by the administration server.

To modify the administration server 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 **Administration server log**.
3. Enter the path and file name for the administration server error log. Ensure that the file exists on the LDAP server and that the path is valid. See "Default log paths" on page 359 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. Under **Log Schedule**, do the following:
   - Specify the frequency between two cycles of Event by selecting an item from the **Select frequency** check box.
   - Specify the start date and start time for Event in the Starting on fields. You can also click the calendar icon to specify the start date. The start time should be in the following format: 12:30:00 PM.
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 82. After stopping the server you must also stop and start the administration server locally to resynchronize the ports.
   - Issue the commands:

     ```
     ibmdirctl -D <AdminDN> -w <AdminPW> -p <admin server portnumber> stop

     ibmdirctl -D <AdminDN> -w <AdminPW> -p <admin server portnumber> admstop

     idsdiradm

     ibmdirctl -D <AdminDN> -w <AdminPW>  -p <admin server portnumber> start
     ```
   - For Windows systems, you can also:
     a. Go to **Control Panel**->**Administrative Tools**->**Services**.
     b. Select **IBM Tivoli Directory Admin Server V6.3 – <***InstanceName***>** .

c. Do one of the following:
   - Click **Action -> Stop**.
   - Click **Stop the service**.

d. Select **IBM Tivoli Directory Admin Server V6.3 – <***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 server 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 server 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 who are assigned AuditAdmin or ServerConfigGroupMember role 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 server 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:

- The Primary directory administrator and administrative group members with Audit administrator and Server configuration group member roles are the only users who can access the administration server audit log settings.

- 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.

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 server 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 **Administration server audit log**.
3. Select **Enable admin server audit logging** to use the audit log utility with the administration server.

   Note: The default setting is enabled. You only need to select the check box, if you have previously disabled the administration server audit log.
4. Enter the path and file name for the administration server audit log. Ensure that the file exists on the ldap server and that the path is valid. See "Default log paths" on page 359 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. Under **Log Schedule**, do the following:
   - Specify the frequency between two cycles of Event by selecting an item from the **Select frequency** check box.
   - Specify the start date and start time for Event in the Starting on fields. You can also click the calendar icon to specify the start date. The start time should be in the following format: 12:30:00 PM.
9. Under **Operations to log**, do the following:
   - Select the **Bind** check box to enable logging for bind operation. Otherwise, to disable logging for bind operation, clear the check box.

- Select the **Unbind** check box to enable logging for unbind operation. Otherwise, to disable logging for unbind operation, clear the check box.
- Select the **Search** check box to record LDAP search operations performed by any client . Otherwise, to disable search, clear the check box.
- Select the **Add** check box to records additions to LDAP. Otherwise, to disable this feature, clear the check box.
- Select the **Modify** check box to record modifications to LDAP. Otherwise, to disable this feature, clear the check box.
- Select the **Delete** check box to records deletions from LDAP. Otherwise, to disable this feature, clear the check box.
- Select the **Modify RDN** check box to record modifications made to RDNs. Otherwise, to disable this feature, clear the check box.
- Select the **Event notification** check box to record event notifications. Otherwise, to disable this feature, clear the check box.
- Select the **Extended operations** check box to enable logging for extended operations. Otherwise, to disable logging for extended operations, clear the check box.

10. 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: 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-auditbind
ibm-auditbind: {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-auditsearch
ibm-auditsearch: {TRUE|FALSE}
#select TRUE to enable, FALSE to disable
-
replace: ibm-auditadd
```

```
ibm-auditadd: {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-auditdelete
ibm-auditdelete: {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-auditextopevent
ibm-auditextopevent: {TRUE|FALSE}
#select TRUE to enable, FALSE to disable
-
replace: ibm-auditExtOp
ibm-auditExtOp: {TRUE|FALSE}
#select TRUE to enable, FALSE to disable
```

To update the settings dynamically, issue the following commands:

```
idsldapexop -p <instance port> -D <adminDN> -w <adminPW> -op readconfig \
 -scope entire

idsldapexop -p <administration server port> -D <adminDN> -w <adminPW> \
 -op readconfig -scope entire
```

Disabling the administration server audit log

To disable audit logging:

Using Web Administration

1. Expand **Logs** in the navigation area, click **Modify log settings**.
2. Click **Administration server audit log**.
3. Deselect **Enable admin server audit logging**.
4. 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
```

To update the settings dynamically, issue the following commands:

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

idsldapexop -p <administration server port> -D <adminDN> -w <adminPW>
-op readconfig -scope entire
```

Configuring preaudit records

Auditing can be configured to audit operations before they complete. This is known as pre-auditing. When pre-audit records are enabled, the audit plugin is invoked to update an audit record before the operation completes. To enable pre-auditing, you must set the value of the IBMSLAPD_PREOP_AUDIT environment variable to "YES". This can be done by accessing the environment variable or by using the ldapmodify command with the following format:

```
ldapmodify -D <adminDN> -w <adminPW>
dn: cn=Front End, cn=configuration
changetype: modify
add: ibm-slapdSetEnv
ibm-slapdSetEnv: IBMSLAPD_PREOP_AUDIT=YES
```

Note:

- The server must be restarted for the changes to take effect.
- Pre-auditing must be used only for debugging purposes. It changes the format and breaks tools that parse the logs.

An example of a pair of diagnostic audit records when pre-audit is enabled, where the sequence identifier is 3: <"PREOP: 3" and "POSTOP: 3">, is as follows:

```
AuditV3--2007-08-29-11:44:32.912-06:00DST--V3 PREOP: 3 threadId:  1161116592
Search--bindDN: cn=root--client: 127.0.0.1:1044--connectionID:
3--received: 2007-08-29-11:44:32.912-06:00DST--Success
controlType: 1.3.6.1.4.1.42.2.27.8.5.1
criticality: false
base: o=sample
scope: baseObject
derefAliases: neverDerefAliases
typesOnly: false
filter: (objectclass=*)

AuditV3--2007-08-29-11:44:33.092-06:00DST--V3 POSTOP: 3 threadId:  1161116592
Search--bindDN: cn=root--client: 127.0.0.1:1044--connectionID:
3--received: 2007-08-29-11:44:32.912-06:00DST--Success
controlType: 1.3.6.1.4.1.42.2.27.8.5.1
criticality: false
base: o=sample
scope: baseObject
derefAliases: neverDerefAliases
typesOnly: false
filter: (objectclass=*)
```

Enabling the server audit log and modifying server 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 Version 6.3 Server Plug-ins Reference*.

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 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:

```
2009-08-05-14:06:20.957-06:00--GLPSRV009I IBM Tivoli Directory (SSL),
Version 6.3   Server started.
```

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:
```

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

.

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:
  - 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

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 **Server Audit log**.

   Note:

   - The directory administrator and administrative group members are the only users who can access this panel.
   - On some platforms, logging is provided through standard operating system logging mechanisms. On these platforms, this panel cannot be used to configure directory server logs. For example, on an OS/400® platform, the directory server job log contains all server messages. However, in the case of i5/OS directory server version 6.1 and above, the Audit log panel is displayed and directory server logs for audit can be configured.
   - If you have the Log Management Tool installed, you can set the Log size threshold, Maximum log archives, and Log archive path values. Values entered into these fields will not take effect if the Log Management Tool is not installed. See the *IBM Tivoli Directory Server Version 6.3 Problem Determination Guide* for more information about the Log Management Tool.
3. Select **Enable server 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 359 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. Under **Log Schedule**, do the following:
    - Specify the frequency between two cycles of Event by selecting an item from the **Select frequency** check box.
    - Specify the start date and start time for Event in the Starting on fields. You can also click the calendar icon to specify the start date. The start time should be in the following format: 12:30:00 PM.
12. 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-auditfailedoponly
ibm-auditfailedoponly: {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

Click **Server administration** in the Web Administration navigation area and then click **Logs** in the expanded list.

1. Click **Modify log settings**.
2. Click **Server audit log**.
3. Deselect **Enable audit logging**.
4. 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
```

Performance profiling

IBM Tivoli directory server provides information about the run-time performance of the server using a performance trace based on the independent trace facility (ldtrc). Additionally, Tivoli Directory Server provides information indicative of performance hotspots during operation execution in the audit record for each operation. Hence, the server can publish performance information in:

- A performance trace, based on the independent trace facility.
- Audit logs.

Performance profiling through the independent trace facility

The performance profile information in trace is intended to help users diagnose performance problems. By using the independent trace facility, performance profiling is accomplished with minimum impact on server performance. The independent trace facility profiles operation performance that consists of timestamps at key points traversed during an operation execution for a running server instance. The timestamps are profiled during different stages such as the following:

- RDBM search processing
- RDBM bind processing
- RDBM compare processing
- RDBM write processing

Note: Timestamp collection points for individual operations are provided only for the RDBM backend.

The instance configuration option ibm-slapdStartupTraceEnabled governs the tracing of performance records at server startup. With dynamic tracing (ldaptrace client utility), the independent trace utility can be made to start or stop collecting performance records after server startup. To activate tracing of performance records dynamically, do the following:

1. Activate tracing for performance records. To do this, issue the ldaptrace command of the following format:

   ```
   ldaptrace –h <hostname> -p <port number> -D <adminDN> -w <adminpwd> -l on \
    –t start -- -perf
   ```

2. Dump the trace to a binary trace file. To do this, issue the following command:

   ```
   ldtrc dmp trace.bin
   ```

3. Format the trace. To do this, issue the following command:

   ```
   ldtrc fmt trace.bin trace.txt
   ```

After formatting the trace you can analyze the trace and diagnose performance problems. To turn off tracing, issue the following command:

```
ldtrc off
```

Given below is an example of formatted performance trace:

```
prf_entry LD PERF FrontEnd::operation_in_workQ (3.100.98.1.1.0)
pid 10255; tid 1167334320; sec 1159183071; nsec 84815000
En-queue bind op; Worker thread ID 1133718448;
Work Q size now = 1; client conn (9.124.231.39: conn ID 1)
```

Auditing for performance profiling

Tracing timestamps using the independent trace facility gives a detailed performance profile. However, to identify performance bottlenecks during operation execution, you can also check the audit log for the summary figures indicating performance hotspots. These hotspots are best provided as a summary. For instance, the operation response time, time spent in worker queue, the accumulated RDBM lock wait times, and time spent in client I/O per operation. Following are the hotspots identified for auditing:

- When an operation has to wait in the worker thread queue for a long time before the worker thread actually starts executing the operation.
- The time spent for cache contention inside the backend needs to be tracked.
- The time spent in handling client I/O, that is, the time spent in receiving the request and returning the result. This value can also be used for detecting bottlenecks because of slow clients or network issues.

For each operation, performance data field in the audit records is controlled using the configuration option “ibm-auditPerformance”. The value of the “ibm-auditPerformance” field is ‘false’ by default and therefore no performance data will be collected and published by default.

When the value of the “ibm-auditPerformance” field is set to ‘true’, performance data will be collected and published in the audit logs for each operation that is enabled to be audited.

If the “ibm-auditPerformance” field is enabled, that is, set to ‘true’, in audit record section four performance data fields are audited: operationResponseTime, timeOnWorkQ, rdbmLockWaitTime, and clientIOTime. The value of these performance data fields is in milliseconds. A brief description of the performance data fields is given below:

- **operationResponseTime** – This field represents the time difference in milliseconds between the time the operation was received and the time its response was sent. The operation received time and the response sent time of an operation are published in audit v3 header.
- **timeOnWorkQ** – This field represents time in milliseconds spent in the worker queue before execution is initiated on the operation. The value of this field is the difference between the time execution was initiated and the time the operation was received.
- **rdbmLockWaitTime** - This field represents time in milliseconds spent in acquiring locks over RDBM caches during operation execution. The value in this field helps administrators to determine the time spent for cache contention against real work.

  The lock wait time over the following resources are also considered.

  – Resource cache
  – DN cache
  – Entry cache
  – Filter cache
  – Attribute cache

    Note: Starting with the IBM Tivoli Directory Server 6.3 release, attribute cache is deprecated. Henceforth, users should avoid using attribute cache.

  – Deadlock detector
  – RDBM locks

- **clientIOTime** – This field represents time in milliseconds that was spent in receiving the complete operation request and returning the complete operation response. This field is implemented in the operation structure and is updated on receiving the complete BER for operation request and on successfully returning the response BER message for the operation.

An example of the audit version 3 format for a search operation issued when ibm-auditPerformance is enabled will look like:

```
AuditV3--2006-09-09-10:49:01.863-06:00DST--V3 Search--bindDN:
cn=root--client: 127.0.0.1:40722--connectionID: 2--received:
2006-09-09-10:49:01.803-06:00DST–Success
controlType: 1.3.6.1.4.1.42.2.27.8.5.1
criticality: false
base: o=sample
scope: wholeSubtree
derefAliases: neverDerefAliases
typesOnly: false
filter: (&(cn=C*)(sn=A*))
operationResponseTime: 591
timeOnWorkQ: 1
rdbmLockWaitTime: 0
clientIOTime: 180
```

To enable audit for performance data, use one of the following methods:

Using Web Administration

1. Expand **Logs** under Server administration in the navigation area and click **Modify log settings**.
2. Click **Server audit log**.
3. Under audit performance data, select the **Enable audit for performance data** check box to log performance data related to the server in the audit log.

Using command line

Issue the following command to enable audit for performance data:

```
ldapmodify -h <hostname> -p <port number> -D <adminDN>  -w <adminpwd>
dn: cn=Audit,cn=Log Management,Configuration
changetype: modify
replace: ibm-auditPerformance
ibm-auditPerformance: true
```

Modifying bulkload log settings

Bulkload is used for loading entries. The bulkload log allows you to view status and errors related to bulkload. See the **idsbulkload** command information in the *IBM Tivoli Directory Server Version 6.3 Command Reference* for more information.

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 359 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. Under **Log Schedule**, do the following:
   - Specify the frequency between two cycles of Event by selecting an item from the **Select frequency** check box.
   - Specify the start date and start time for Event in the Starting on fields. You can also click the calendar icon to specify the start date. The start time should be in the following format: 12:30:00 PM.
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:

```
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 Appendix L, "Dynamically-changed attributes," on page 587 for a list of the attributes that can be updated dynamically.

Modifying tools log settings

The configuration tools log enables you to view status and error messages related to the configuration tools, such as **idscfgdb**, **idsucfgdb**, **idscfgchglog**, **idsucfgchglog**, **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 **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 359 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. Under **Log Schedule**, do the following:
   - Specify the frequency between two cycles of Event by selecting an item from the **Select frequency** check box.
   - Specify the start date and start time for Event in the Starting on fields. You can also click the calendar icon to specify the start date. The start time should be in the following format: 12:30:00 PM.
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=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 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 359 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. Under **Log Schedule**, do the following:
   - Specify the frequency between two cycles of Event by selecting an item from the **Select frequency** check box.
   - Specify the start date and start time for Event in the Starting on fields. You can also click the calendar icon to specify the start date. The start time should be in the following format: 12:30:00 PM.
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=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 Appendix L, “Dynamically-changed attributes,” on page 587 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**.

   Note:

- The directory administrator and administrative group members are the only users who can access this panel.
- On some platforms, logging is provided through standard operating system logging mechanisms. On these platforms, this panel cannot be used to configure or view directory server logs. For example, on an OS/400 platform, the directory server job log contains all server messages. However, in the case of i5/OS directory server version 6.1 and above, the Lost and found log panel is displayed and logs related to errors that occur as a result of a replication conflict can be recorded in the Lost and found log.
- If you have the Log Management Tool installed, you can set the Log size threshold, Maximum log archives, and Log archive path values. Values entered into these fields will not take effect if the Log Management Tool is not installed. See the *IBM Tivoli Directory Server Version 6.3 Problem Determination Guide* for more information about the Log Management Tool.

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 359 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. Select the **Log members for group entries involved in a conflict** check box to log the members of group entries into lost and found log during replication conflict resolution. The "ibm-slapdLogMembers" attribute in the entry "cn=Replication, cn=Log Management, cn=Configuration" is associated with this control. The group members' cache should be enabled for performance reasons when group member entries are required to be logged in the lost and found log. If groups have very large number of member entries, it is advisable to disable logging of all members.

   Note: The attribute ibm-slapdLogMembers is significant only in the case of "cn=Replication, cn=Log Management, cn=Configuration" entry. For all other log settings, the ibm-slapdLogMembers attribute remains insignificant.

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. Under **Log Schedule**, do the following:

- Specify the frequency between two cycles of Event by selecting an item from the **Select frequency** check box.
- Specify the start date and start time for Event in the Starting on fields. You can also click the calendar icon to specify the start date. The start time should be in the following format: 12:30:00 PM.

9. 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 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 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 359 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. Under **Log Schedule**, do the following:
   - Specify the frequency between two cycles of Event by selecting an item from the **Select frequency** check box.
   - Specify the start date and start time for Event in the Starting on fields. You can also click the calendar icon to specify the start date. The start time should be in the following format: 12:30:00 PM.
8. 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 2009 GLPSRV009I IBM Tivoli Directory (SSL), Version 6.3
         Server started.
     ```

   - Medium logs a medium amount of error information, for example,

     ```
     Oct 06 10:35:41 2009 GLPCOM024I The extended Operation plugin is successfully
         loaded from libloga.dll.
     Oct 06 10:35:41 2009 GLPCOM003I Non-SSL port initialized to 389.
     Oct 06 10:35:44 2009 GLPSRV009I IBM Tivoli Directory (SSL), Version 6.3
         Server started.
     ```

   - High logs the most amount of error information, for example

     ```
     Oct 06 10:37:48 2009 GLPSRV047W Anonymous binds will be allowed.
     Oct 06 10:37:48 2009 GLPCOM024I The extended Operation plugin is successfully
         loaded from libloga.dll.
     Oct 06 10:37:48 2009 GLPSRV003I Configuration file successfully read.
     Oct 06 10:37:48 2009 GLPCOM003I Non-SSL port initialized to 389.
     Oct 06 10:37:51 2009 GLPSRV009I IBM Tivoli Directory (SSL), Version 6.3
         Server started.
     ```

9. 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 <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 Appendix L, "Dynamically-changed attributes," on page 587 for a list of the attributes that can be updated dynamically.

Start/stop server tracing

To start or stop server tracing use the following method

Using Web Administration

If you have not done so already, click **Server administration** in the Web Administration navigation area and then click **Logs** in the expanded list. Next, click **Start/stop server tracing**.

On this panel, you can:

- Enable server tracing
- Set the level of trace debug data to be collected
- Specify the debug output file to which the trace information to be send

To enable trace facility:

1. Select the **Enable server tracing** check box to enable tracing for this server instance.
2. Specify a trace debug level in the **Trace debug levels** field.
3. Specify the file to which the trace information should be send in the **Trace debug file** field.
4. After you have finished, do one of the following:
   - Click **OK** to save the changes and return to the Introduction panel.
   - Click **Cancel** to discard changes made and return to the Introduction panel.

Viewing logs

The following sections show you how to view the IBM Tivoli Directory Server logs. For a selected log file, the View logs panel displays the most recent logs in the View log table in ascending order. The View log table displays 20 rows, where a logged item could span over one or more rows. You can navigate over the pages in the View log table by clicking the navigation arrow provided on the status bar of the table or by entering the page number in the field on the status bar and clicking **Go**.

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**.

   Note:

   - The directory administrator and administrative group members are the only users who can access this panel.
   - On some platforms, logging is provided through standard operating system logging mechanisms. On these platforms, this panel cannot be used to view directory server logs. For example, on an OS/400 platform, the directory server job log contains all server messages. However, in the case of i5/OS directory server version 6.0 and above, the Select log combo box will only display Audit log and Lost and found log, provided the ibm-supportedCapability OIDs 1.3.18.0.2.32.80 and 1.3.18.0.2.32.52 for Audit log and Lost and found log respectively are displayed on root DSE search.
   - When the Web admin tool is used to access the admin server:
     - The status bar on the View logs panel displays a message indicating that the tool is connected to the admin server. If you access panels that are not supported by admin server, a message is displayed indicating that the functions on the panels are not supported.
     - The View logs panel is enabled based on the capabilities present in rootDSE for ibm-supportedcapabilities attribute.
     - The Clear button on the View logs panel is disabled as the admin server does not support clear log request.
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 server error log

To view the administration server 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 Administration server 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 Administration server error log:

```
ldapexop -D <adminDN> -w <adminPW> -op clearlog -log  idsdiradm
```

Viewing the admin server audit log settings

To view the administration server 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 server 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 server 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 IBM Tivoli 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 IBM Tivoli 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 IBM Tivoli 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 IBM Tivoli 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 IBM Tivoli 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
```

Log integration into CBE and CARS format

In an effort to create self-managing environment, IBM has taken initiative in introducing "Autonomic Computing". Autonomic computing is an open standard based architecture that allows systems to configure, heal, optimize, and protect itself. In order to determine the conditions of the different components of the system, it is necessary to standardize the format of the event data so that the system can resolve its current conditions.

To standardize the format of data for the problem determination architecture IBM introduced a common format for log and trace information called the Common Base Event (CBE) format. This format creates consistency across similar fields and improves the availability to correlate across multiple logs. CBE is based on a 3-tupple structured format, which includes:

- Component impacted by a situation, or the source
- Component observing a situation
- Situation data, the properties describing the situation including correlation information

The 3-tupple format makes it possible to write and deploy resource-independent management functions that can isolate a failing component.

In an effort to align IBM Tivoli Directory Server to autonomic computing space, it is a must to have the logs such as error log, audit log, and so on, produced by the Tivoli Directory Server product to provide these logs in CBE format.

IBM Common Auditing and Reporting Service (CARS) component leverages CBE, which is a common format for events proposed by IBM, and IBM Common Event Infrastructure (CEI) technologies to provide an audit infrastructure. The purpose of CBE is to facilitate effective intercommunication among disparate components within an enterprise. In order to effectively process audit data, the CARS component requires the audit data to be in the CBE format. CEI is an IBM strategic event infrastructure for submission, persistent storage, query, and subscription of the CBE events. The CARS component uses the CEI interfaces for submission of events. These events can be denoted as auditable by using configuration options at the CEI Server that stores them in a CEI XML Event store that meets the auditing requirements.

The CARS component allows staging of data from the CEI XML Event store into report tables. IBM products and customers can provide audit reports based on auditable events staged into report tables. The CARS component also supports managing the lifecycle of auditable events, which includes archive, restore, and audit reports on restored archives.

Log management tool for CBE, CEI, and CARS features

To start the execution of the CBE, CEI, and CARS features for a Tivoli Directory Server instance, you need to run the Tivoli Directory Server log management tool, idslogmgmt, as an owner of that instance.

Note: Only one instance of idslogmgmt can be run on a Tivoli Directory Server instance, and only one instance of idslogmgmt that manages the admin tools log can be run.

To implement the CBE, CEI, and CARS features, you need to launch IBM Tivoli Directory Integrator server and the assembly lines using the idslogmgmt wrapper. The log management assembly lines will initially read and process the parameters passed by the wrapper script. Next, the log management assembly lines read the Tivoli Directory Server instance repository file and determine the version of log management tool associated with the servers installed. For the list of servers, the ibmslapd.conf file is read and the log management settings are retrieved. The tool checks for the setting updates in the Tivoli Directory Server instances' configuration files in regular intervals. The default interval is 5 minutes. If IDSLMG_CHECK_INTERVAL variable is set, then the value set in this variable takes precedence. See *IBM Tivoli Directory Server Version 6.3 Installation and Configuration Guide* to know more about CARS installation and configuration.

After the log management configuration settings are read from the ibmslapd.conf file, the tool finds the location of logs and performs the appropriate log management activities. The activities can include managing of log disk space usage or converting proprietary format log data into CBE format and sending that data to a file or a CEI server.

When the idslogmgmt tool is run, a pid file, idslogmgmt.pid, containing the process ID will be created and updated in the *<instance home>\tmp* directory. This pid file help in determining which idslogmgmt is running or stopped for a Tivoli Directory Server instance when the status action is specified by the log management extend operation. This only applies to instance specific idslogmgmt

execution and not in the execution in which admin tools parameters are specified. See the Common Base Event (CBE) features section in *IBM Tivoli Directory Server Version 6.3 Problem Determination Guide* to know more about special case scenarios related to CBE.

Entries for log management

The log management attributes associated with the CBE, CEI, and CARS feature are placed under the following entries depending on the attributes.

cn=default, cn=Log Management, cn=configuration
: This applies to all log management entries unless they are overwritten by specifying the settings explicitly in the individual log entries.

cn=<specific_log_name>, cn=Log Management, cn=configuration
: This applies only to the log specified by the entry. The default settings for this log can be overwritten by specifying the settings in this entry. The values for <specific_log_name> are: ibmslapd, audit, tools, bulkload, admin, admin audit, db2cli, replication, and ddsetup.

CARS Reports

The CARS report generates the required CBE properties. Tivoli Directory Serve log entries that are mapped to CBE properties are basically categorized into the Base Properties and Security Extension Properties. The Base Properties tables list the properties that are part of the CBE v1.0.1 specification. The Security Extension Properties tables list the properties that are part of the CBE extension for Security Events v0.21 specification. The properties are expressed in XPath statement, which describes where the property is found within a CBE.

A sample CBE formatted output

A sample CBE formatted output for an audit record would be as given below:

```
AuditV3--2005-11-14-18:27:37.444-06:00--V3 Bind--bindDN:
         cn=root--client: 127.0.0.1:1193--connectionID:
         1--received: 2005-11-14-18:27:37.444-06:00--Success
controlType: 1.3.6.1.4.1.42.2.27.8.5.1
criticality: false
name: cn=root
authenticationChoice: simple
Admin Acct Status: Not Locked

<?xml version="1.0" encoding="UTF-8"?>
<CommonBaseEvent creationTime="2005-11-14T12:27:37"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:noNamespaceSchemaLocation="commonbaseevent1_0.xsd"
         globalInstanceId="i00000000000000000000000000000000"
         sequenceNumber="00000000000000000000000000000001"
         extensionName= "SECURITY_AUDIT_AUTHN">
   <sourceComponentId component="Official Product Name"
         subcomponent="audit"
         componentIdType="ProductName"
         componentType="http://www.ibm.com/namespace/autonomic/Tivoli_componentTypes"
         location="127.0.0.1:389"
         locationType="IPV6"
         instanceId="ldapdev"/>
   <situation categoryName="ConnectSituation">
         <situationType reasoningScope="EXTERNAL"
                        successDisposition="SUCCESSFUL"
                        situationDisposition="AVAILABLE"/>
   </situation>
   <extendedDataElements name="action">
         <values>authentication</values>
   </extendedDataElements>
```

```
   <extendedDataElements name="authnType">
         <values>ldap_3.0</values>
   </extendedDataElements>
   <extendedDataElements name="outcome">
         <result>SUCCESSFUL</result>
   </extendedDataElements>
   <extendedDataElements name="outcome">
         <failureReason>authenticationFailure</failureReason>
   </extendedDataElements>
   <extendedDataElements name="resourceInfo">
         <children name="type">
           <values>application</values>
         </children>
         <children name="nameInPolicy">
           <values>ldap</values>
         </children>
         <children name="nameInApp">
           <values>ldap</values>
         </children>
   </extendedDataElements>
   <extendedDataElements name="userInfo">
         <children name="appUserName">
           <values>cn=root</values>
         </children>
         <children name="proxyUserName">
           <values>NOT AVAILABLE</values>
         </children>
         <children name="registryUserName">
           <values>NOT AVAILABLE</values>
         </children>
         <children name="sessionID">
           <values>1</values>
         </children>
         <children name="location">
           <values>127.0.0.1:1193</values>
         </children>
         <children name="locationType">
           <values>IPV6</values>
         </children>
   </extendedDataElements>
</CommonBaseEvent>
```

Configuring log management attributes for CBE and CARS

Using Web Administration

Here, an example of the admin server log is considered. Depending on the log that you select, there might be some change in the controls displayed on the panel.

If you have not done so already, do the following:

1. Click **Server administration** in the Web Administration navigation area and then click Logs in the expanded list.
2. Click **Modify log settings**.
3. From the list of logs, select **Admin server log**.

To modify the admin server log:

1. Enter the path and file name for the administration server log.
2. Under Log size threshold (MB), select one of the following:
   - Select the radio button with the edit field next to it and enter the maximum log size in Megabytes.
   - If you want to use the default limit, select the radio button next to the drop-down menu. Select **Default** from the drop-down menu.

- If you do not want to set a limit to the log size, select the radio button next to the drop-down menu. Select **Unlimited** from the drop-down menu.

3. Under Maximum log archives, select one of the following:
   - Select the radio button with the edit field next to it and enter the maximum number of archives you want to save. One archived log is an earlier log that reached its size threshold.
   - If you want to use the default maximum log archives value, select the radio button next to the drop-down menu. Select **Default** from the drop-down menu.
   - If you do not want to archive logs, select the radio button next to the drop-down menu. Select **No archives** from the drop-down menu.
   - If you do not want to limit the number of archived logs, select the radio button next to the drop-down menu. Select **Unlimited** from the drop-down menu.
4. Under Log archive path, select 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 **Default path** radio button.
5. Specify the frequency of between two cycles of the CBE feature by selecting an item from the **Select frequency** check box.
6. Specify the start date and start time for the CBE feature in the Starting on fields. You can also click the calendar icon to specify the start date. The start time should be in the following format: *12:30:00 PM*.
7. After you have finished, do one of the following:
   - Click **Next** to continue with the configuring of log settings.
   - Click **Finish** to save the changes and return to the Modify log settings panel.
   - Click **Cancel** to discard changes made on this panel and to navigate to the Modify log settings panel.

To configure log settings for event-formatted log file, do the following:

- Select the **Send log records to event-formatted log file** check box to enable CBE formatted log file for the user.
- Specify the path name to store the CBE formatted log file in the **File path** field.
- Specify the file name in the **File name prefix** field for the CBE formatted log.
- Specify the threshold size for the CBE formatted log file in MB under Log size threshold (MB). If you want to specify a size limit in MB, select the option and specify a numeric value in the field. Otherwise, select **Unlimited**.
- Specify the maximum number of logs to be archived for the CBE formatted log. If you want to specify the maximum number of logs to be archived, select the option and specify a numeric value in the field. To set it to unlimited, select **Unlimited**.
- Specify the path name where CBE formatted log should be archived. If you want to specify a path name, select the option and enter the absolute path name for logs to be archived. To specify the archive path same as that of log file, select **Same directory as of log file**.
- Specify the log level for the CBE formatted log. The available log levels are High, Medium, and Low.
- After you have finished, do one of the following:
  – Click **Back** to go to the previous panel.

- Click **Next** to continue with the configuring of log settings.
- Click **Finish** to save the changes and return to Modify log settings panel.
- Click **Cancel** to discard changes made on this panel and to navigate to the Modify log settings panel.

Note:

- If no value is entered in Log archive path, the default value will be assigned.
- If 0 is set in fields that require numerical value, it is considered as "Unlimited" except for "Maximum log archives", where 0 is considered as "No archives".

To configure Common Audit and Reporting Service:

- Select the **Send log records to common audit and reporting service** check box to enable the CBE feature to read Tivoli Directory Server proprietary formatted logs and to convert them to CBE format and to write them to CEI server.
- In the **Host** field, enter the host name of the CEI server.
- In the **Port** field, enter the port number on which the CEI server listens on.
- Specify the log level for the CBE formatted log. The available log levels are High, Medium, and Low.
- After you have finished, do one of the following:
  - Click **Back** to go to the previous panel.
  - Click **Finish** to save the changes and return to Modify log settings panel.
  - Click **Cancel** to discard changes made on this panel and to navigate to the Modify log settings panel.

To start or stop log management using the Web administration tool: If you have not done so already, click **Logs** under **Server administration** in the Web Administration navigation area and click **Start/Stop log management** in the expanded list.

Using this panel, the primary administrator and local administrative group members with AuditAdmin or ServerConfigGroupMember role can start and stop the log management service.

To start or stop the log management service:

- Do one of the following:
  - If the log management service is running, click **Stop** to stop the service.
  - If the log management service is stopped, click **Start** to start the service.
- Click **Close** to return to the "Introduction" panel.

Using the command line

To set the attribute values for the CBE and CARS feature:

```
#idsldapmodify -h <host_name> -p <portnumber>  -D <cn=RDN_value> -w <password> \
 -f <file_name>
```

where contents of <file_name> is as follows:

```
dn: cn=<specific_log_name>,cn=Log Management, cn=configuration
ibm-slapdLogEventFileEnabled: true
-
add:ibm-slapdLogCARSEnabled
ibm-slapdLogCARSEnabled: false
-
```

```
add: ibm-slapdLogEventFormat
ibm-slapdLogEventFormat: CBE
-
add: ibm-slapdLogMgmtStartTime
ibm-slapdLogMgmtStartTime: 200609010000
-
add: ibm-slapdLogMgmtFrequency
ibm-slapdLogMgmtFrequency: 20
-
add:ibm-slapdLogEventFileSizeThreshold
ibm-slapdLogEventFileSizeThreshold: 2
-
add:ibm-slapdLogEventFileMaxArchives
ibm-slapdLogEventFileMaxArchives: 2
-
add:ibm-slapdLogEventFileArchivePath
ibm-slapdLogEventFileArchivePath: <path_name>/TempDir
-
add: ibm-slapdLogEventFileOptions
ibm-slapdLogEventFileOptions: <h|m|l>
-
add: ibm-slapdLogEventFilePath
ibm-slapdLogEventFilePath: /home/inst1/idsslapd-<instance_name>/logs
-
add:ibm-slapdLogEventFilePrefix
ibm-slapdLogEventFilePrefix: <log_name>
-
add: ibm-slapdLogSizeThreshold
ibm-slapdLogSizeThreshold: 1
-
add: ibm-slapdLogMaxArchives
ibm-slapdLogMaxArchives: 1
-
add: ibm-slapdLogArchivePath
ibm-slapdLogArchivePath: <path_name>/TempDir1
```

To start an instance:

```
ibmslapd -I <instance_name> -n
```

You can start log management locally or remotely. To start log management locally, issue the following command:

```
idslogmgmt -I <instance_name>
```

To start, get status, and stop log management remotely, issue the following commands:

```
ibmdirctl -D <adminDN> -w <password> -h <host_name> \
 -p <administration server port number> startlogmgmt

ibmdirctl -D <adminDN> -w <password> -h <host_name> \
 -p <administration server port number> statuslogmgmt

ibmdirctl -D <adminDN> -w <password> -h <host_name> \
 -p <administration server port number> stoplogmgmt
```

Part 3. Directory Management

© Copyright IBM Corp. 2002, 2010

Chapter 18. 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 414). After you have located the level for the entry that

© Copyright IBM Corp. 2002, 2010

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=sample. 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. 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 403.

    b. If an attribute requires binary data, click **Binary data**. See "Binary data for attributes" on page 403

    c. If your server has language tags enabled, click **Language tag value** to add or remove language tag descriptors. See "Language tags" on page 405 and "Language tag values for attributes" on page 407 for more information.

    d. If an attribute contains referrals, click **Manage referral**. See Chapter 13, "Referrals," on page 211 and "Creating default referrals" on page 215 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.
14. After successfully adding an entry, you will be prompted to add a similar entry. To add a similar entry, click **Yes**.To exit and return to the Manage entries panel, click **No**.

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=sample
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 on 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 407 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 tp 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=sample
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=sample" (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 specifing 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 403.
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" on page 408.

Searching for entries containing attributes with language tags

Issuing the command,

```
idsldapsearch -b "o=sample" "cn=Mark Anthony" sn
```

return the following results:

```
cn=Mark Anthony,o=sample
sn=Anthony
sn;lang-spanish=Antonio
```

Note: All versions of "sn" are displayed in the output.

Issuing the command,

```
idsldapsearch -b "o=sample" "cn=Mark Anthony" sn;lang-spanish
```

returns the following results.

```
cn=Mark Anthony,o=sample
sn;lang-spanish=Antonio
```

Note: Only "sn;lang-spanish" is displayed in the output.

Issuing the command,

```
idsldapsearch -b "o=sample" "sn;lang-spanish=Antonio"
```

returns the entire entry:

```
cn=Mark Anthony,o=sample
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=sample
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=sample"
```

The above delete command will fail if the entry to be deleted, "cn=John Doe, ou=Austin, o=sample", is not a leaf entry. In order to delete a non-leaf entry, use the -s option of the **idsldapdelete** utility as shown below :

```
idsldapdelete -D <adminDN> -w <adminPW> -s "cn=John Doe, ou=Austin, o=sample"
```

This command will delete the entry "cn=John Doe, ou=Austin, o=sample" and all the entries below it.

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** or click the RDN name of an entry in the RDN column to open the Edit attributes panel.

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 403.

   b. In case of multi-valued attributes, if an attribute has values more than the maximum number of values to return for each attribute limit, then the values of the attribute is displayed using a combo box, where the number of the values displayed will be equal to the value of the limit. Also, for this attribute the Multiple values button will not be displayed and a message indicating this attribute value truncation is displayed.

   c. If an attribute requires binary data, click **Binary data**. See “Binary data for attributes” on page 403

   d. If your server has language tags enabled, click **Language tag value** to add or remove language tag descriptors. See “Language tags” on page 405 and “Language tag values for attributes” on page 407 for more information.

   e. If an attribute contains referrals, click **Manage referral**. See Chapter 13, “Referrals,” on page 211 and “Creating default referrals” on page 215 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.

Note: In case an entry has more attributes than the maximum number of attributes to return for each entry limit, then the entry is returned with all the values of attributes until the maximum number of attributes to return for each entry limit is reached. The attributes for which no values are returned are displayed at the lower-end of the panel. These attributes are displayed along with a message indicating that the entry is not complete.

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=sample" "cn=Robert Garcia"
```

Note: The -r option causes the old name to be removed.

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=sample" "cn=Bob Garcia, ou=deptABC, ou=Austin, o=sample" "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=sample
changetype: modrdn
newrdn: cn=Bob Garcia
deleteoldrdn: 0
newsuperior: ou=deptXYZ, ou=Austin, o=sample
```

Note: Renaming or moving an entry in a proxy environment is supported only if it does not move entries across partitions.

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=sample
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**.

1. Change the RDN entry in the DN field. For example change cn=John Doe to cn=Jim Smith.
2. 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.
3. On the required attributes tab, change the cn entry to the new RDN. In this example Jim Smith.
4. Change the other required attributes as appropriate. In this example change the sn attribute from Doe to Smith.
5. Click **Next** to display the Optional attributes tab.
6. Change the optional attributes as appropriate and click **Next** to display the Static memberships tab.
7. At the Static memberships tab, choose the group memberships that you want the copied entry to be a member of. This tab also allows the user to edit the memberships of the copied entry.

   Note: The Static memberships tab is only displayed when you copy an entry, where the entry being copied is a member of a static group. In case an entry is not a member of a static group, the Static memberships tab will not be displayed for the Copy entry operation.
8. When you have finished making the necessary changes click **Finish** to create the new entry.
9. 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 454 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=sample" (objectclass=*)
   ```

   which returns something like the following (save this information to a file):

   ```
   dn: cn=Bob Garcia, ou=deptABC, ou=Austin, o=sample
   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=sample
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=sample, 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 430.

See Chapter 19, "Access control lists," on page 419 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 403.

   b. If an attribute requires binary data, click **Binary data**. See "Binary data for attributes" on page 403

c. If your server has language tags enabled, click **Language tag value** to add or remove language tag descriptors. See "Language tags" on page 405 and "Language tag values for attributes" on page 407 for more information.
d. If an attribute contains referrals, click **Manage referral**. See Chapter 13, "Referrals," on page 211 and "Creating default referrals" on page 215 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=sample
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=sample
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 405 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 405 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 416.
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 and action.
    - **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 18, “Working with directory entries,” on page 401) 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**.

Using command line

Given below are examples of a simple and an advanced search. To execute a simple search issue the following command:

```
idsldapsearch -D <userDN> -w <userPW> -b <Subtree DN> -s SUB  cn=John
```

To execute an advanced search, issue the following command:

```
idsldapsearch -D <userDN> -w <userPW> -b <Subtree DN> -s SUB (&(cn=John)(sn=Smith))
```

The above example searches for entries with cn=John and sn=Smith. Here, two search criteria have been combined into a single filter using the AND (&) logical operator.

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=sample.
  - **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 13, “Referrals,” on page 211 and Chapter 19, “Access control lists,” on page 419for more information.
- Select the **Include deleted entries** check box to enable deleted entries to be returned for a search operation.

See “Search Settings” on page 115 for additional information about searches.

Chapter 19. 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 local administration group members who are assigned the DirDataAdmin role 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.

© Copyright IBM Corp. 2002, 2010

- 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:

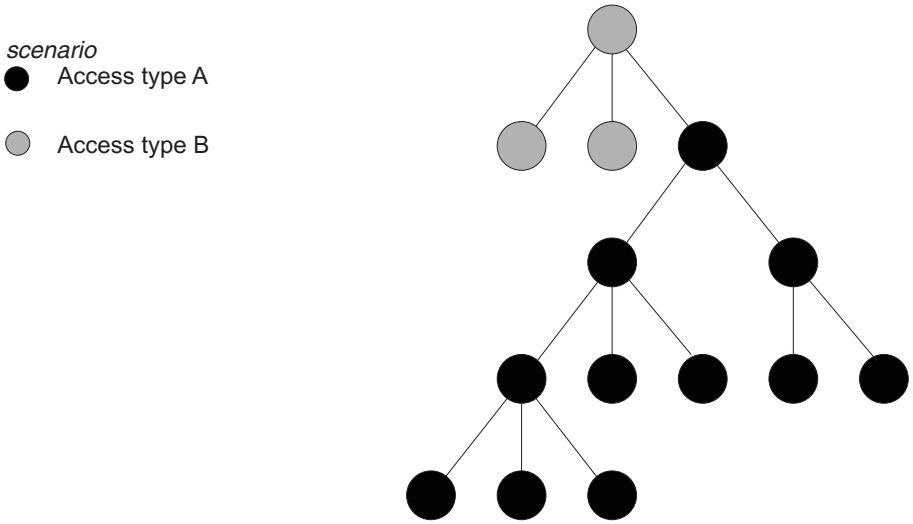
- ibm-filterAclEntry
- ibm-filterAclInherit

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.

ACL type usage scenarios

Non-filter ACLs are intended to be useful in situations where the access topology of the directory calls for a homogeneous sub-tree distribution of permissions, as described in the example given below. In this scenario, access needs to be applied to the directory objects in an even distribution in the tree.

Figure 19. Non-filter ACL scenario



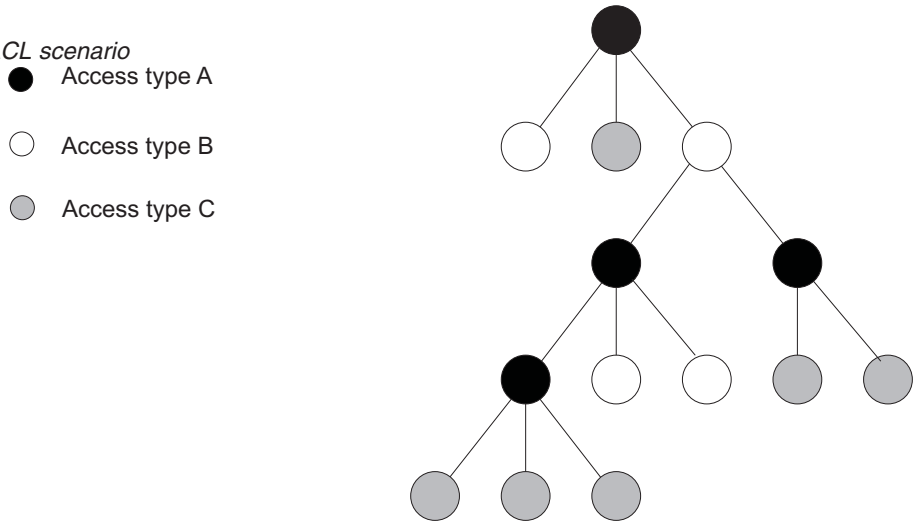
Example 1: Non-filter ACL Scenario

To accomplish this, a single set of non-filter ACL specifications can be defined at, or near, the top entry of the directory. The ACLs can propagate homogeneously throughout the directory sub-tree, so that they apply to all sub-tree objects. Also, since there is no comparison matching associated with this type of ACL, less processing is involved.

Filter-based ACLs are intended to be useful in situations where the access topology of the directory calls for a heterogeneous sub-tree distribution of permissions, as described in the example given below. In this scenario several access types are required, and they need to be applied to the directory objects in a more scattered

distribution.

Figure 20. Filter-based ACL scenario



Example 2: Filter-based ACL Scenario

To accomplish this, a single set of filter-based ACL specifications can be defined at the suffix entry of the directory with filters that are associated with each of the desired access types. The filters correspond to attributes contained in the various objects that are distributed throughout the directory tree.

The correct permissions will be applied to a particular directory object based on a successful comparison match with the attributes contained in that object. ACL administration is simplified due to a single location at the suffix. In contrast, to achieve the same set of ACLs using non-filter ACLs would require ACL specifications in every directory object in the tree.

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 ACI 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"

<objectPermisssions> ::=  <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=sample or group: cn=deptXYZ, o=sample".

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, o=sample" and the bindDn is "cn=personA, o=sample", the permissions granted are a combination of those given to "cn=this" and those given to "cn=personA, o=sample".

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, o=sample 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, o=sample	normal:rsc:sensitive:rcs:critical:rwsc
cn=personB, o=sample	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, o=sample AclEntry:

```
access-id:cn=personA, o=sample: object:ad
AclEntry: access-id: cn = this:critical:rwsc
AclEntry: group: cn=Anybody: normal:rsc
AclEntry: group: cn=Authenticated: sensitive:rcs
```

For an operation performed on cn=personA, o=sample:

User Binding as	Would receive
cn=personA, o=sample	object:ad:critical:rwsc
cn=personB, o=sample	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, o=sample". Note that the group permissions are not given because there is a more specific aclentry ("access-id:cn=personA, o=sample") for the bind DN ("cn=personA, o=sample").

Example 3

Consider the following ACL for object: cn=personA, o=sample 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, o=sample	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	• search, read (on attributes in RDN) • search (on attributes specified in the search filter) • search (on attributes returned with just names) • search, read (on attributes returned with values)
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=sample 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=sample
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=sample
postalcode: 1515
aclentry: group: cn=Bowling Team, ou=Groups, o=sample: 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 420 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, local administrative group members who are assigned the DirDataAdmin role, 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 aclEntry 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 **aclEntry** is found in access evaluation, the group level aclEntries are not included in access calculation. The exception is that if the matching access-id level **aclEntries** are all defined under cn=this, then all matching group level **aclEntries** 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 aclEntry. 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 aclEntries 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=sample belongs to both group1 and group2 with the following aclEntries defined:

1. aclEntry: access-id: cn=Person A, o=sample: at.attributel:grant:rsc:sensitive:deny:rsc
2. aclEntry: group: cn=group1,o=sample:critical:deny:rwsc
3. aclEntry: group: cn=group2,o=sample: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=sample: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=sample.
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 51 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-flitered 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=sample. 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 431 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=sample, 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=sample 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-flitered ACLs: To remove non-flitered 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 DNs 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 flitered 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=sample. 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=Ricarco 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=sample 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 431 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=sample, 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=sample 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 flitered 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=sample.
   - 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 ACIs 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=sample
ownerPropagate: true

entryOwner: group:cn=System Owners, o=sample
ownerPropagate: true
```

The next example shows how an access ID "cn=Person 1, o=sample" 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 ACI, 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=sample:(objectclass=groupOfNames):
                    at.attribute1:grant:rsc

ibm-filterAclInherit: false
```

The next example shows how a group "cn=Dept XYZ, o=sample" is being given permissions to read, search and compare attribute1. The permission applies to the entire subtree below the node containing this ACI.

```
aclEntry: group:cn=Dept XYZ,o=sample:at.attribute1:grant:rsc
aclPropagate: true
```

The next example shows how a role "cn=System Admins,o=sample" 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 ACI.

```
aclEntry: role:cn=System Admins,o=sample:object:grant:a:at.
         attribute2:grant:rsc:critical:grant:rsc
aclPropagate: false
```

Modifying the ACI 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 ACIs for an entry:

```
aclEntry: group:cn=Dept ABC,o=sample:normal:grant:rsc
aclPropagate: true
```

perform the following change:

```
dn: cn=some entry
changetype: modify
replace: aclEntry
aclEntry: group:cn=Dept XYZ,o=sample:normal:grant:rsc
```

The resulting ACI is:

```
aclEntry: group:cn=Dept XYZ,o=sample: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=sample:(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=sample:(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=sample:(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=sample:normal:grant:rsc
```

with a modification:

```
dn: cn=some entry
changetype: modify
add: aclEntry
aclEntry: group:cn=Dept ABC,o=sample:at.attribute1:grant:rsc
```

would yield an multi-valued aclEntry of:

```
aclEntry: group:cn=Dept XYZ,o=sample:normal:grant:rsc
aclEntry: group:cn=Dept ABC,o=sample:at.attribute1:grant:rsc
```

For example, given the ACI:

```
Ibm-filterAclEntry: group:cn=Dept XYZ,o=sample:(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=sample:(cn=Manager ABC)
                  :at.attribute1:grant:rsc
```

would yield an multi-valued aclEntry of:

```
Ibm-filterAclEntry: group:cn=Dept XYZ,o=sample:(cn=Manager XYZ):normal
                  :grant:rsc
ibm-filterAclEntry: group:cn=Dept ABC,o=sample:(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=sample: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=sample:normal:grant:sc:normal:deny:r:critical
        :grant::sensitive:grant:r
```

For example, given the following ACI:

```
Ibm-filterAclEntry: group:cn=Dept XYZ,o=sample:(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=sample:(cn=Manager XYZ):normal
                  :deny:r:critical:deny::sensitive:grant:r
```

yields an aclEntry of:

```
ibm-filterAclEntry: group:cn=Dept XYZ,o=sample:(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=sample:object:grant:ad
aclEntry: group:cn=Dept XYZ,o=sample:normal:grant:rwsc

dn: cn = some entry
```

```
changetype: modify
delete: aclEntry
aclEntry: group:cn=Dept XYZ,o=sample:object:grant:ad
```

yields a remaining ACI on the server of :

```
aclEntry: group:cn=Dept XYZ,o=sample:normal:grant:rwsc
```

Given an ACI of:

```
ibm-filterAclEntry: group:cn=Dept XYZ,o=sample:(cn=Manager XYZ):object
                  :grant:ad
ibm-filterAclEntry: group:cn=Dept XYZ,o=sample:(cn=Manager XYZ):normal
                  :grant:rwsc

dn: cn = some entry
changetype: modify
delete: ibm-filterAclEntry
ibm-filterAclEntry: group:cn=Dept XYZ,o=sample:(cn=Manager XYZ):object
                  :grant:ad
```

yields a remaining ACI on the server of:

```
ibm-filterAclEntry: group:cn=Dept XYZ,o=sample:(cn=Manager XYZ):normal
                  :grant:rwsc
```

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=sample" -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 20. 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 19, "Access control lists," on page 419. 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.

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,o=sample
objectclass: accessGroup
cn: Dev.Staff
member: cn=John Doe,ou=Austin,o=sample
member: cn=Jane Smith,ou=Austin,o=sample
member: cn=James Smith,ou=Austin,o=sample
```

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.

Note: Referential integrity results in updates to the modifyTimeStamp of the group entry to which a member belongs. In a replication environment, ldap operations of type deletion, modrdn, or movement of a member entry from one tree to another invokes referential integrity on both Master (Supplier) and Replica (Consumer). To avoid any replication conflict that may arise because of group entries on master and replica bearing different timestamp values, the modifyTimeStamp of the affected group is set to the value of the modifyTimeStamp of the member entry that was affected in the last operation, subject to the modifyTimeStamp of the last operation being later than the existing modifyTimeStamp of the group.

© Copyright IBM Corp. 2002, 2010

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 **idsldapsearch** command information in the *IBM Tivoli Directory Server Version 6.3 Command Reference* for more 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 **memberURL**s 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=sample
 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=sample
 member: cn=Person 2.2, cn=Dept 2, cn=Employees, o=sample
 ibm-memberGroup: cn=Group 8, cn=Nested Static, cn=Groups, o=sample
```

The introduction of cycles into the nested static group hierarchy is not allowed. If it is determined that a nested static 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=sample
 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=sample??one?objectClass=person
```

```
ibm-memberGroup: cn=Group 9, cn=Nested Dynamic, cn=Groups, o=sample
member: cn=Person 10.1, cn=Dept 2, cn=Employees, o=sample
member: cn=Person 10.2, cn=Dept 2, cn=Employees, o=sample
```

Determining group membership

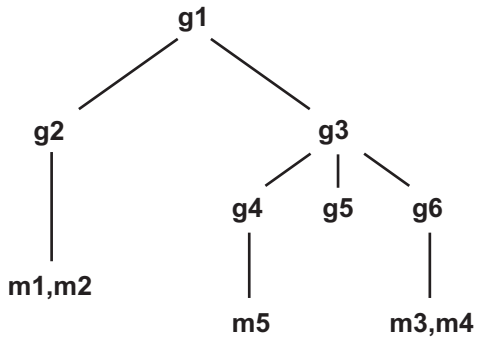
Two operational attributes can be used to query aggregate group membership. For a given group entry, the **ibm-allMembers** operational attribute enumerates the aggregate set of group membership, including static, dynamic, and nested members, as described by the nested group hierarchy. For a given user entry, the **ibm-allGroups** operational attribute enumerates the aggregate set of groups, including ancestor groups, to which that user has membership.

Note:

- The ibm-allMembers operational attribute is processed in a distributed environment also.
- The proxy server obtains the dynamic members of a nested group only if they reside on the same backend server. Also, in case of proxy server, only global admin group members can perform the ibm-allMembers search.
- The ibm-allMembers search is supported only for base searches.
- 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, assume that the directory contains the following entries:

```
dn: cn=g1,cn=groups,o=sample
objectclass: groupOfNames
objectclass: ibm-nestedGroup
cn: g1
ibm-memberGroup: cn=g2,cn=groups,o=sample
ibm-memberGroup: cn=g4,cn=groups,o=sample
ibm-memberGroup: cn=g5,cn=groups,o=sample

dn: cn=m1, cn=users,o=sample
objectclass:person
```

```
cn: m1
sn: one
aclentry: access-id:cn=user1,cn=users,o=sample:normal:rsc
aclentry: access-id:cn=user2,cn=users,o=sample:normal:rsc

dn: cn=m2, cn=users,o=sample objectclass:person
cn: m2
sn: two
aclentry: access-id:cn=user1,cn=users,o=sample:normal:rsc
aclentry: access-id:cn=user2,cn=users,o=sample
```

Assume that **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=sample
objectclass: accessGroup
cn: g2
member: cn=m1,cn=users,o=sample
member: cn=m2,cn=users,o=sample
aclentry: access-id:cn=user1,cn=users,o=sample:normal:rsc
aclentry: access-id:cn=user2,cn=users,o=sample: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=sample
objectclass: accessGroup
cn: g4
member: cn=m5, cn=users,o=sample
```

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=sample
objectclass: container
objectclass: ibm-dynamicGroup
cn: g5
memberURL: ldap:///cn=users,o=sample??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=sample
objectclass:person
cn: m3
sn: three
aclentry: access-id:cn=user1,cn=users,o=sample:normal:rsc
aclentry: access-id:cn=user2,cn=users,o=sample:normal:rsc

dn: cn=m4, cn=users,o=sample
objectclass:person
cn: m4
sn: four
aclentry: access-id:cn=user1,cn=users,o=sample:normal:rsc
aclentry: access-id:cn=user2,cn=users,o=sample
```

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=sample -w user1pwd -s base -b cn=g1,
           cn=groups,o=sample objectclass=* ibm-allmembers
```

```
cn=g1,cn=groups,o=sample
ibm-allmembers: CN=M1,CN=USERS,o=sample
ibm-allmembers: CN=M2,CN=USERS,o=sample
ibm-allmembers: CN=M3,CN=USERS,o=sample
ibm-allmembers: CN=M4,CN=USERS,o=sample
ibm-allmembers: CN=M5,CN=USERS,o=sample
```

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=sample -w user2pwd -s base -b cn=g1,
          cn=groups,o=sample objectclass=* ibm-allmembers


cn=g1,cn=groups,o=sample
ibm-allmembers: CN=M3,CN=USERS,o=sample
ibm-allmembers: CN=M5,CN=USERS,o=sample
```

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=sample -w user2pwd -s base -b cn=m3,
          cn=users,o=sample objectclass=* ibm-allgroups


cn=m3,cn=users,o=sample
ibm-allgroups: CN=G1,CN=GROUPS,o=sample
```

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=sample -w user2pwd -s base -b
          cn=m1,cn=users,o=sample objectclass=* ibm-allgroups


cn=m1,cn=users,o=sample
```

Example 5:

Depending on the ACLs associated with an user, the evaluation of the search consisting of the **ibm-allMembers** operational attribute for dynamic groups might give varied results. This example illustrates how access control can affect evaluation of the **ibm-allMembers** operational attributes for dynamic groups.

Consider the entries for two groups in LDIF defined as follows:

```
dn: cn=claims,cn=groups,o=sample
objectclass: top
objectclass: groupOfURLs
memberURL: ldap:///cn=users,o=sample??sub?(ibm-group=claims)
cn: claims

dn: cn=departmentNum, cn=groups, o=sample
objectclass: top
objectclass: groupOfURLs
memberURL: ldap:///cn=users,o=sample??one?(|(departmentnumber=2001)
     (departmentnumber=2002))
```

Consider the entries for users in LDIF defined as follows:

```
dn: uid=adavid, cn=users, o=sample
objectclass: top
objectclass: organizationalPerson
objectclass: inetOrgPerson
objectclass: ibm-dynamicMember
cn: Al
sn: David
departmentnumber: 2001
ibm-group: claims

dn: uid=jchevy, cn=users, o=sample
objectclass: top
objectclass: organizationalPerson
objectclass: inetOrgPerson
objectclass: ibm-dynamicMember
cn: Jerry
sn: Chevy
departmentnumber: 2002
ibm-group: claims
```

Here, the default access control, **cn=anybody**, is used, which has read, search, and compare access. This DN has access-class defined as "normal".

An user with required administrative privileges runs a search returning **ibm-allMembers** for these groups, the search returns:

```
idsldapsearch -D cn=root -w ? -b "cn=groups, o=sample" -s one objectclass=*
               ibm-allMembers

cn=departmentNum,cn=groups,o=sample
ibm-allMembers=uid=adavid,cn=users,o=sample
ibm-allMembers=uid=jchevy,cn=users,o=sample

cn=claims,cn=groups,o=sample
ibm-allMembers=uid=adavid,cn=users,o=sample
ibm-allMembers=uid=jchevy,cn=users,o=sample
```

The result displays the entries that satisfy the search criteria departmentnumber=2001 or departmentnumber=2002 and ibm-group=claims.

If the same search is performed anonymously, the search returns:

```
idsldapsearch -b "cn=groups, o=sample" -s one objectclass=* ibm-allMembers

cn=departmentNum,cn=groups,o=sample
ibm-allMembers=uid=adavid,cn=users,o=sample
ibm-allMembers=uid=jchevy,cn=users,o=sample

cn=claims,cn=groups,o=sample
```

In the displayed result, entries that are members of the departmentNum group are returned that satisfy the search criteria departmentnumber=2001 or departmentnumber=2002, and no entries are returned as a member of the claims group. This is because the ibm-group attribute has access-class defined as "critical", while the departmentnumber attribute has access-class defined as "normal". Moreover, anonymous users do not have search access to attributes of access-class "critical".

In a dynamic group, the members are defined using an LDAP search. Therefore, the search for dynamic members and determination of group membership is internal to the directory server and therefore no access control applies.

However, if a client application retrieves ibm-allGroups to manage authority within some other application, then you need to be sure that the application does these searches using an identity that has the necessary authority.

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=sample. 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 405 for more information.

11. In the **member** field, add the DN for at least one member. For example cn=Bob Garcia,ou=austin,o=sample.

    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=sample.

    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 403 for information on adding binary values.

14. Click **Finish** to create the entry.

See "Managing members of group entries" on page 453 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=sample.

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=sample. 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 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 405 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=sample??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=sample. 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 405 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=sample**.
    a. Click **Multiple values**.
    b. In the **member** field, type **cn=Tuesday,ou=Groups,o=sample**.
    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=sample and click **Expand**.

   Note: An expandable entry indicates that the entry has child entries. Expandable entries have a plus '+' sign next to them in the Expand column. You can click the '+' sign next to the entry to view the child entries of the selected entry.
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=sample and cn=tuesday,ou=group,o=sample are listed.

6. Click the **Effective group members** tab.
7. Specify the maximum number of members to return for a group. If you click Maximum number of members to return, you must enter a number. Otherwise, click **Unlimited**.
8. To populate the table with the members of a group, click **Load** or select Load from Select Action and click **Go**.

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=sample 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. Specify the maximum number of members to return for a group. If you click Maximum number of members to return, you must enter a number. Otherwise, click **Unlimited**.
6. 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=sample** and **cn=Ricardo Garcia,ou=austin,o=sample** 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:

7. Type the name of entry that you want to add as a member of the group for example **cn=Kyle Nguyen,ou=austin,o=sample** in the member field or select it using the **Browse** function (Expand o=sample → Expand ou=Austin → Select cn=Kyle Nguyen,ou=austin,o=sample).
8. Click **Add**.
9. **cn=Kyle Nguyen,ou=austin,o=sample** 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=sample**, **cn=Ricardo Garcia,ou=austin,o=sample** and **cn=Kyle Nguyen,ou=austin,o=sample** are now members of the Monday group.
10. If you click on the **Effective group members** tab and click **Refresh**, **cn=Bob Garcia,ou=austin,o=sample**, **cn=Ricardo Garcia,ou=austin,o=sample** and **cn=Kyle Nguyen,ou=austin,o=sample** are now displayed as members.

Editing a member entry in a group

To edit a member entry in a group:

1. From the navigation area, expand **Directory management** .
2. Click **Manage entries**.
3. Expand the various subtrees and select the group entry that you want to work on.
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 edit. For this, click **Static group members**.
6. To populate the table with the members of a group, click **Load** or select Load from Select Action and click **Go**.
7. To edit an existing member's entry details, select the member entry you want to edit from the **member** or **uniqueMember** table and do one of the following:
   - Click **Edit**.
   - Select **Edit** from the **Select Action** drop-down menu and click **Go**.

   Note: This displays the Edit attributes panel for the selected member entry. On this panel, you may modify the appropriate fields.

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=sample 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. Specify the maximum number of members to return for a group. If you click Maximum number of members to return, you must enter a number. Otherwise, click **Unlimited**.
7. To populate the table with the members of a group, click **Load** or select Load from Select Action and click **Go**.
8. 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**.
9. You are prompted to confirm the removal. Click **OK** to remove the member.
10. 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.

Note: You can also delete a static member entry by entering a member DN in the member field and by clicking Delete. The Delete button is displayed only when no members are loaded in the member table.

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=sample.
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=sample** 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 membesrhip for the entry **cn=Bob Garcia,ou=austin,o=sample**.

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=sample.
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=sample 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 414.

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 21. Managing search limit groups

In 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 “Search Settings” on page 115 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 theses 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.

© Copyright IBM Corp. 2002, 2010

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=sample`.

   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 403.

   b. If an attribute requires binary data, click **Binary data**. See “Binary data for attributes” on page 403

c. If your server has language tags enabled, click **Language tag value** to add or remove language tag descriptors. See "Language tags" on page 405 and "Language tag values for attributes" on page 407 for more information.

d. If an attribute contains referrals, click **Manage referral**. See Chapter 13, "Referrals," on page 211 and "Creating default referrals" on page 215 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=sample
member: cn=user2,o=sample
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 409.

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=sample
```

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 410.

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*>. The 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=sample
member: cn=user2,o=sample
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 408.

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 22. 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 IBM Tivoli Directory Server on behalf of multiple users.

Note: Proxy authorization is different from the proxy server.

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 theses 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 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 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.

© Copyright IBM Corp. 2002, 2010

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=sample`.

   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 403.

   b. If an attribute requires binary data, click **Binary data**. See “Binary data for attributes” on page 403

   c. If your server has language tags enabled, click **Language tag value** to add or remove language tag descriptors. See “Language tags” on page 405 and “Language tag values for attributes” on page 407 for more information.

   d. If an attribute contains referrals, click **Manage referral**. See Chapter 13, “Referrals,” on page 211 and “Creating default referrals” on page 215 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=sample
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=sample
```

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=sample" -w <client1password>
           -y "cn=userA,o=sample" -b "o=sample" -s sub ou=austin
```

Based on the above idsldapsearch specification, client1 can read from the target directories whatever userA has permission to read.

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 409.

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=sample
-
add: member
member: cn=client2, ou=austin, o=sample
-
add: member
member: cn=client3, ou=austin, o=sample
```

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 410.

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=sample
member: cn=client2, ou=austin, o=sample
member: cn=client3, ou=austin, o=sample
```

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=sample
member: cn=client2, ou=austin, o=sample
member: cn=client3, ou=austin, o=sample
```

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 408.

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=sample" -w <client1password>
           -y "cn=userA,o=sample" -b "o=sample" -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

© Copyright IBM Corp. 2002, 2010

Chapter 23. 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=sample. 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=sample**.
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=sample**.
7. Change the Subject type to **group**.
8. Click **Add**.

© Copyright IBM Corp. 2002, 2010

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 accidently delete him or herself. In this example the location might be **o=sample**.
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=sample (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 403.

   b. If an attribute requires binary data, click **Binary data**. See "Binary data for attributes" on page 403

   c. If your server has language tags enabled, click **Language tag value** to add or remove language tag descriptors. See "Language tags" on page 405 and "Language tag values for attributes" on page 407 for more information.

   d. If an attribute contains referrals, click **Manage referral**. See Chapter 13, "Referrals," on page 211 and "Creating default referrals" on page 215 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=sample) and select the realm you just created, **cn=realm1,o=sample**.
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=sample** in the member field or select it using the **Browse** function (expand o=sample and select cn=John Doe,o=sample).
6. Click **Add**.
7. **cn=John Doe,o=sample** 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 453 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=sample**, locate the template in the subtree **o=sample**. 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 479.
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=sample** 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=sample**.
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 441 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 454 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 (acls) 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=sample**. 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 430.

See Chapter 19, "Access control lists," on page 419 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=sample**. 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 479.
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 477 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 430.

See Chapter 19, “Access control lists,” on page 419 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 30 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 481.
   - **Delete** - See “Removing a user” on page 481.
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 441 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 30 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 481.
- **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. Appendixes

© Copyright IBM Corp. 2002, 2010

Appendix A. Error codes

The possible values for an LDAP error code are shown in the following tables:

Table 23. 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.

© Copyright IBM Corp. 2002, 2010

Table 23. 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 23. 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 23. 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. **Note:** 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 24. 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 25. 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
176	LDAP_REG_FILE_NOT_FOUND	B0	File not found in NT registry
177	LDAP_REG_CANNOT_OPEN	B1	Can not open NT registry
178	LDAP_REG_ENTRY_NOT_FOUND	B2	Entry not found in NT registry
192	LDAP_CONF_FILE_NOT_OPENED	C0	Plugin configuration file not opened
193	LDAP_PLUGIN_NOT_LOADED	C1	Plugin library not loaded
194	LDAP_PLUGIN_FUNCTION_ NOT_RESOLVED	C2	Plugin function not resolved
195	LDAP_PLUGIN_NOT_INITIALIZED	C3	Plugin library not initialized
196	LDAP_PLUGIN_COULD_NOT_BIND	C4	Could not bind to plugin function
208	LDAP_SASL_GSS_NO_SEC_CONTEXT	D0	gss_init_sec_context failed

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 6.3. 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 6.3 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 and above 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.

supportedsaslmechanisms
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.

© Copyright IBM Corp. 2002, 2010

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 6.3.

ibm-enabledcapabilities
Lists the server capabilities currenty enabled on the server. See "OIDs for supported and enabled capabilities" on page 493 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 6.3, the vendor version is set to 6.3.

ibm-slapdServerBackend
Specifies whether the server loads a database or proxy backend.

ibm-slapdSizeLimit
Limits the number of entries returned by a search initiated by nonadministrative users.

ibm-slapdTimeLimit
Specifies in seconds the maximum amount of time the server spends processing a search request initiated by nonadministrative users.

ibm-slapdDerefAliases
Describes how the server is configured to handle dereferrencing.

ibm-supportedAuditVersion
The supported version of auditing. For example, in version 6.0 and above 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 501 for the values.

ibm-supportedcapabilities
Lists the server capabilities currently supported by the server. See "OIDs for supported and enabled capabilities" on page 493 for the values.

ibm-sasldigestrealmname
Displays the SASL digest realm name associated with the server.

ibm-slapdServerInstanceName
Name of the directory server instance running on the server.

ibm-slapdisconfigurationmode
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 26. OIDs for supported and enabled capabilities

Short name with OID	Description	Supported by IBM Tivoli Directory Base Server v6.3	Supported by IBM Tivoli Directory Proxy Server v6.3	
			Without partitioned data	With partitioned data
Enhanced Replication Model 1.3.18.0.2.32.1	Identifies the replication model including subtree and cascading replication.	Yes	N/A	N/A
EntryChecksum 1.3.18.0.2.32.2	Indicates that this server supports the ibm-entrychecksum and ibm-entrychecksumop features.	Yes	Yes	Yes
Entry UUID 1.3.18.0.2.32.3	This value is listed in the ibm-capabilities Subentry for those suffixes that support the ibm-entryuuid attribute.	Yes	Yes	Yes
Filter ACLs 1.3.18.0.2.32.4	Identifies that this server supports the IBM Filter ACL model	Yes	Yes	Yes
Password Policy 1.3.18.0.2.32.5	Identifies that this server supports password policies	Yes	Yes	Yes
Sort by DN 1.3.18.0.2.32.6	Enables searches sorted by DNs in addition to regular attributes.	Yes	No	No
Administration Group Delegation 1.3.18.0.2.32.8	Server supports the delegation of server administration to a group of administrators that are specified in the configuration backend.	Yes	Yes	Yes
Denial of Service Prevention 1.3.18.0.2.32.9	Server supports the denial of service prevention feature including read/write time-outs.	Yes	Yes	Yes
Dereference Alias Option 1.3.18.0.2.32.10	Server supports an option to not dereference aliases by default * Proxy rootDSE does not list ** Aliases across partitions are not dereferenced	Yes	Yes(*)	Yes(**)

Table 26. OIDs for supported and enabled capabilities (continued)

Short name with OID	Description	Supported by IBM Tivoli Directory Base Server v6.3	Supported by IBM Tivoli Directory Proxy Server v6.3	
			Without partitioned data	With partitioned data
Admin Server Audit Logging 1.3.18.0.2.32.11	Server supports the auditing of the admin server.	Yes	Yes	Yes
128 Character Table Names 1.3.18.0.2.32.12	The server feature to allow name of unique attributes to be higher than 18 characters (with the maximum of 128 characters). * Proxy rootDSE does not list ** Uniqueness is not guaranteed across partitions	Yes	Yes(*)	Yes(**)
Attribute Caching Search Filter Resolution 1.3.18.0.2.32.13	The server supports attribute caching for search filter resolution.	Yes	N/A	N/A
Dynamic Tracing 1.3.18.0.2.32.14	Server supports active tracing for the server with an LDAP extended operation.	Yes	Yes	Yes
Entry And Subtree Dynamic Updates 1.3.18.0.2.32.15	The server supports dynamic configuration updates on entries and subtrees.	Yes	Yes	Yes
Globally Unique Attributes 1.3.18.0.2.32.16	The server feature to enforce globally unique attribute values.	Yes	No	No
Group-Specific Search Limits 1.3.18.0.2.32.17	Supports extended search limits for a group of people. * Proxy rootDSE does not list ** Group Based Search limits don't work consistently when data is partitioned	Yes	Yes(*)	Yes(**)
IBMpolicies Replication Subtree 1.3.18.0.2.32.18	Server supports the replication of the cn=IBMpolicies subtree.	Yes	Yes	Yes
Max Age ChangeLog Entries 1.3.18.0.2.32.19	Specifies that the server is capable of retaining changelog entries based on age.	Yes	N/A	N/A
Monitor Logging Counts 1.3.18.0.2.32.20	The server provides monitor logging counts for messages added to server, command-line interface, and audit log files.	Yes	Yes	Yes
Monitor Active Workers Information 1.3.18.0.2.32.21	The server provides monitor information for active workers (cn=workers,cn=monitor).	Yes	Yes	Yes

Table 26. OIDs for supported and enabled capabilities (continued)

Short name with OID	Description	Supported by IBM Tivoli Directory Base Server v6.3	Supported by IBM Tivoli Directory Proxy Server v6.3	
			Without partitioned data	With partitioned data
Monitor Connection Type Counts 1.3.18.0.2.32.22	The server provides monitor connection type counts for SSL and TLS connections.	Yes	Yes	Yes
Monitor Connections Information 1.3.18.0.2.32.23	The server provides monitor information for connections by IP address instead of connection ID (cn=connections, cn=monitor)	Yes	Yes	Yes
Monitor Operation Counts 1.3.18.0.2.32.24	The server provides new monitor operation counts for initiated and completed operation types. * The operations completed counts do not reflect actual operations completed in the proxy. Instead it represents operations that are either completed or have been sent to a backend server for processing. Proxy Specific Monitors should be used in Tivoli Directory Server 6.1 and above versions.	Yes	Yes(*)	Yes(*)
Monitor Tracing Info 1.3.18.0.2.32.25	The server provides monitor information for tracing options currently being used.	Yes	Yes	Yes
Null Base Subtree Search 1.3.18.0.2.32.26	Server allows null based subtree search, which searches the entire DIT defined in the server.	Yes	No	No
Proxy Authorization 1.3.18.0.2.32.27	Server supports Proxy Authorization for a group of users.	Yes	No	No
TLS Capabilities 1.3.18.0.2.32.28	Specifies that the server is actually capable of doing TLS.	Yes	Yes	Yes
Non-Blocking Replication 1.3.18.0.2.32.29	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.	Yes	N/A	N/A
Kerberos Capability 1.3.18.0.2.32.30	Specifies that the server is capable of using Kerberos.	Yes	No	No
ibm-allMembers and ibm-allGroups operational attributes 1.3.18.0.2.32.31	Indicates whether or not a backend supports searching on the ibm-allGroups and ibm-allMembers operational attributes.	Yes	Yes	Yes

Table 26. OIDs for supported and enabled capabilities (continued)

Short name with OID	Description	Supported by IBM Tivoli Directory Base Server v6.3	Supported by IBM Tivoli Directory Proxy Server v6.3	
			Without partitioned data	With partitioned data
All operational Attributes 1.3.6.1.4.1.4203.1.5.1	All operational Attributes * Proxy rootDSE does not list ** Some operational attributes are dependent on the data not being distributed.	Yes	Yes(*)	Yes(**)
Language Tags 1.3.6.1.4.1.4203.1.5.4	Server supports language tags.	Yes	No	No
FIPS mode for GSKit 1.3.18.0.2.32.32	Enables the server to use the encryption algorithms from the ICC FIPS-certified library	Yes	Yes	Yes
Modify DN (leaf move) 1.3.18.0.2.32.35	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. * modify DN allowed only if the change does not cross partitions	Yes	Yes	Yes(*)
Simplify resizing of attributes 1.3.18.0.2.32.37	Allows customers to increase the maximum length of attributes through the schema modification facilities.	Yes	N/A	N/A
Global Administration Group 1.3.18.0.2.32.38	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.	Yes	Yes	Yes
AES Encryption Option 1.3.18.0.2.32.39	Server supports AES Password Encryption.	Yes	Yes	Yes
Auditing of Compare 1.3.18.0.2.32.40	Server supports auditing of compare operations.	Yes	Yes	Yes
Log Management 1.3.18.0.2.32.41	Identifies that this server supports log management.	Yes	Yes	Yes
Multi-threaded Replication 1.3.18.0.2.32.42	Replication agreements can specify using multiple threads and connections to a consumer.	Yes	N/A	N/A

Table 26. OIDs for supported and enabled capabilities (continued)

Short name with OID	Description	Supported by IBM Tivoli Directory Base Server v6.3	Supported by IBM Tivoli Directory Proxy Server v6.3	
			Without partitioned data	With partitioned data
Supplier Replication Configuration 1.3.18.0.2.32.43	Server configuration of suppliers for replication.	Yes	N/A	N/A
Using CN=IBMPOLICIES for Global Updates 1.3.18.0.2.32.44	Server supports the replication of global updates using the replication topology in cn=IBMpolicies subtree.	Yes	N/A	N/A
Multihomed configuration support 1.3.18.0.2.32.45	Server supports configuration on multiple IP addresses (multihomed).	Yes	Yes	Yes
Multiple Directory Server Instances Architecture 1.3.18.0.2.32.46	Server is designed to run with multiple directory server instances on the same machine.	Yes	Yes	Yes
Configuration Tool Auditing 1.3.18.0.2.32.47	Server supports the auditing of the configuration tools.	Yes	Yes	Yes
Audit consolidation configuration settings 1.3.18.0.2.32.48	Indicates that audit log settings are available in the configuration file.	Yes	Yes	Yes
Proxy Server 1.3.18.0.2.32.49	Describes whether this server is capable of acting as a proxy server or regular RDBM server. Optional Information.	Yes	Yes	Yes
LDAP Attribute Cache Auto Adjust 1.3.18.0.2.32.50	Indicates if autonomic attribute cache is supported and enabled. **Note:** Starting with the IBM Tivoli Directory Server 6.3 release, attribute cache is deprecated. Henceforth, users should avoid using attribute cache.	Yes	N/A	N/A
Replication conflict resolution max entry size 1.3.18.0.2.32.51	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.	Yes	N/A	N/A
LostAndFound log file 1.3.18.0.2.32.52	Supports LostAndFound file for archiving replaced entries as a result of replication conflict resolution.	Yes	N/A	N/A
Password Policy Account Lockout 1.3.18.0.2.32.53	Identifies that this server supports password policy Account Locked feature.	Yes	Yes	Yes

Table 26. OIDs for supported and enabled capabilities (continued)

Short name with OID	Description	Supported by IBM Tivoli Directory Base Server v6.3	Supported by IBM Tivoli Directory Proxy Server v6.3	
			Without partitioned data	With partitioned data
Password Policy Admin 1.3.18.0.2.32.54	Identifies that this server supports Admin Password Policy.	Yes	Yes	Yes
SSL Fips processing mode 1.3.18.0.2.32.55	Server supports SSL FIPS mode processing.	Yes	Yes	Yes
IDS 6.0 ibm-entrychecksumop 1.3.18.0.2.32.56	Identifies that the 6.0 version of the ibm-entrychecksumop calculation was used on the server.	Yes	No	No
LDAP Password Global Start Time 1.3.18.0.2.32.57	Indicates that the server can support ibm-pwdPolicyStartTime attribute in the cn=pwdPolicy entry.	Yes	No	No
Audit Configuration Settings Consolidation 1.3.18.0.2.32.58	Identifies that the audit configuration settings are now residing in the ibmslapd configuration file only. * Transactions are supported only when all updates target a single partition.	Yes	Yes(*)	Yes(*)
CBE Log Format 1.3.18.0.2.32.59	Indicates that Tivoli Directory Server log management and conversion to event format is supported.	Yes	Yes	Yes
Encrypted Attribute Support 1.3.18.0.2.32.60	Server supports encrypted attributes.	Yes	Yes	Yes
Proxy Monitor search 1.3.18.0.2.32.61	Server supports special monitor searches intended for proxy server.	No	Yes	Yes
SSHA Password Encrypt 1.3.18.0.2.32.63	Server supports SSHA Password Encryption.	Yes	Yes	Yes
MD5 Password Encrypt 1.3.18.0.2.32.64	Server supports MD5 Password Encryption.	Yes	Yes	Yes
Filter Replication 1.3.18.0.2.32.65	The server feature designed to have only required entries and a subset of its attributes to be replicated.	Yes	N/A	N/A
Group Members Cache 1.3.18.0.2.32.66	Server supports caching group members.	Yes	N/A	N/A
PKCS11 Support 1.3.18.0.2.32.67	Server supports PKCS11 Encryption standard.	Yes	Yes	Yes

Table 26. OIDs for supported and enabled capabilities (continued)

Short name with OID	Description	Supported by IBM Tivoli Directory Base Server v6.3	Supported by IBM Tivoli Directory Proxy Server v6.3	
			Without partitioned data	With partitioned data
Server Admin Roles 1.3.18.0.2.32.68	Server supports Server Administration roles.	Yes	Yes	Yes
Digest MD5 Support 1.3.18.0.2.32.69	Server supports Digest MD5 Bind.	Yes	Yes	Yes
External Bind Support 1.3.18.0.2.32.70	Server supports External Bind.	Yes	Yes	Yes
Persistent Search 1.3.18.0.2.32.71	Server supports persistent search.	Yes	No	No
Admin Server Denial of Service Prevention 1.3.18.0.2.32.72	Admin Server supports Denial of Service Prevention.	Yes	Yes	Yes
Admin server Enhanced Monitor Support 1.3.18.0.2.32.73	Admin server supports "cn=monitor", "cn=connections,cn=monitor", and "cn=workers,cn=monitor" searches.	Yes	Yes	Yes
Admin Server Support for Schema Searches 1.3.18.0.2.32.74	Admin server supports searches on schema.	Yes	Yes	Yes
System Monitor Search 1.3.18.0.2.32.76	Server supports cn=system,cn=monitor search.	Yes	Yes	Yes
Multiple Password Policies 1.3.18.0.2.32.77	Server allows multiple password policy to be defined and used.	Yes	Yes	No
Passthrough Authentication 1.3.18.0.2.32.78	Server supports pass through authentication feature.	Yes	No	No
Dynamic Updates of Replication Supplier Request 1.3.18.0.2.32.79	Server supports dynamic updates of replication supplier information.	Yes	N/A	N/A
Audit Performance 1.3.18.0.2.32.81	Server supports auditing of performance for operations.	Yes	Yes	Yes
No Emergency Thread Support 1.3.18.0.2.32.82	Emergency Thread is not supported by server.	Yes	Yes	Yes

Table 26. OIDs for supported and enabled capabilities (continued)

Short name with OID	Description	Supported by IBM Tivoli Directory Base Server v6.3	Supported by IBM Tivoli Directory Proxy Server v6.3	
			Without partitioned data	With partitioned data
Enhanced Replication Group RI handling 1.3.18.0.2.32.83	Enhanced Replication Group RI handling	Yes	N/A	N/A
Reread the DB2 Password 1.3.18.0.2.32.84	Server re-reads the DB2 password to identify any change in DB2 password specified in configuration.	Yes	N/A	N/A
Proxy Failback Based on Replication Queue 1.3.18.0.2.32.85	Proxy Server will failback only when replication queue is below the threshold specified in configuration file.	No	Yes	Yes
Proxy Flow control 1.3.18.0.2.32.86	Proxy server supports flow control algorithm.	No	Yes	Yes
Backup restore configuration capability 1.3.18.0.2.32.87	Server supports configuring automatic backup and restore.	Yes	N/A	N/A
Password Policy Max Consecutive repeated characters 1.3.18.0.2.32.88	Server supports restricting maximum consecutive repeated characters in password policy.	Yes	Yes	Yes
Virtual List View Support 1.3.18.0.2.32.89	Server supports virtual list view control in searches.	Yes	No	No
Proxy Paged Search 1.3.18.0.2.32.90	Proxy Server supports paged control in searches.	No	Yes	Yes
Tombstone Support 1.3.18.0.2.32.92	Server supports tombstone for deleted entries.	Yes	No	No
Proxy Health Check outstanding limit 1.3.18.0.2.32.93	Proxy supports identifying a hung server based on the configured outstanding health check requests.	No	Yes	Yes
Replication Finegrained timestamps 1.3.18.0.2.32.94	Replication uses fine grained timestamp for resolving conflicts.	Yes	N/A	N/A
Distributed Dynamic group enabled 1.3.18.0.2.32.96	Proxy Server Supports enabling/Disabling Distributed dynamic group configuration option.	No	Yes	Yes
Distributed group enabled 1.3.18.0.2.32.97	Proxy Server Supports enabling/Disabling Distributed group configuration option.	No	Yes	Yes

Table 26. OIDs for supported and enabled capabilities (continued)

Short name with OID	Description	Supported by IBM Tivoli Directory Base Server v6.3	Supported by IBM Tivoli Directory Proxy Server v6.3	
			Without partitioned data	With partitioned data
SHA-2 1.3.18.0.2.32.99	Indicates that this server supports SHA-2 family of algorithms, which include: SHA-224, SHA-256, SHA-384, and SHA-512. The server also supports the Salted version of the SHA-2 family of algorithms, which include: SSHA-224, SSHA-256, SSHA-384, and SSHA-512. * SHA-2 is only applicable for servers with database backend.	Yes	N/A(*)	N/A(*)

OIDs for ACI mechanisms

The following table shows the OIDs for ACI mechanisms.

Table 27. 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 Tivoli Directory Server filter based ACLs.	1.3.18.0.2.26.3
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 28. OIDs for extended operations

Short name with OID	Description	Supported by the Administration Server	Supported by IBM Tivoli Directory Base Server v6.3	Supported by IBM Tivoli Directory Proxy Server v6.3	
				Without partitioned data	With partitioned data
Account status extended operation 1.3.18.0.2.12.58	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: open locked expired	No	Yes	No	No

Table 28. OIDs for extended operations (continued)

Short name with OID	Description	Supported by the Administration Server	Supported by IBM Tivoli Directory Base Server v6.3	Supported by IBM Tivoli Directory Proxy Server v6.3	
				Without partitioned data	With partitioned data
Attribute type extended operations 1.3.18.0.2.12.46	Retrieve attributes by supported capability: operational, language tag, attribute cache, unique or configuration.	Yes	Yes	Yes	Yes
Begin transaction extended operation 1.3.18.0.2.12.5	Begin a Transactional context.	No	Yes	Yes	Yes
Cascading replication operation extended operation 1.3.18.0.2.12.15	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.	No	Yes	No	No
Clear log extended operation 1.3.18.0.2.12.20	Request to Clear log file.	No	Yes	Yes	Yes
Control replication extended operation 1.3.18.0.2.12.16	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	No	Yes	No	No
Control queue extended operation 1.3.18.0.2.12.17	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.	No	Yes	No	No
DN normalization extended operation 1.3.18.0.2.12.30	Request to normalize a DN or a sequence of DNs.	Yes	Yes	No	No
Dynamic server trace extended operation 1.3.18.0.2.12.40	Activate or deactivate tracing in IBM Tivoli Directory Server.	No	Yes	Yes	Yes
Dynamic update requests extended operation 1.3.18.0.2.12.28	Request to update server configuration for IBM Tivoli Directory Server.	No	Yes	Yes	Yes
Effective password policy extended operation 1.3.18.0.2.12.75	Used for querying effective password policy for a user or a group.	No	Yes	No	No
End transaction extended operation 1.3.18.0.2.12.6	End Transactional context (commit/rollback).	No	Yes	Yes	Yes
Event notification register request extended operation 1.3.18.0.2.12.1	Request registration for events notification.	No	Yes	No	No

Table 28. OIDs for extended operations (continued)

Short name with OID	Description	Supported by the Administration Server	Supported by IBM Tivoli Directory Base Server v6.3	Supported by IBM Tivoli Directory Proxy Server v6.3	
				Without partitioned data	With partitioned data
Event notification unregister request extended operation 1.3.18.0.2.12.3	Unregister for events that were registered for using an Event Registration Request.	No	Yes	No	No
Get file extended operation 1.3.18.0.2.12.73	Returns the contents of a given file on the server.	No	Yes	Yes	Yes
Get lines extended operation 1.3.18.0.2.12.22	Request to get lines from a log file.	Yes	Yes	Yes	Yes
Get number of lines extended operation 1.3.18.0.2.12.24	Request number of lines in a log file.	Yes	Yes	Yes	Yes
Group evaluation extended operation 1.3.18.0.2.12.50	Requests all the groups that a given user belongs to.	No	Yes	No	No
Kill connection extended operation 1.3.18.0.2.12.35	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.	No	Yes	Yes	Yes
LDAP trace facility extended operation 1.3.18.0.2.12.41	Use this extended operation to control LDAP Trace Facility remotely using the Administration Server.	Yes	Yes	Yes	Yes
Locate entry extended operation 1.3.18.0.2.12.71	This extended operation is used to extract the back-end server details of a given set of entry DNs and provide the details to the client.	No	No	Yes	Yes
LogMgmtControl extended operation 1.3.18.0.2.12.70	The LogMgmtControl extended operation is used to start, stop, and query the status of the log management for an IBM Tivoli Directory Server instance running on a server.	Yes	Yes	Yes	Yes
Online Backup extended operation 1.3.18.0.2.12.74	Performs online backup of the directory server instance's DB2 database.	No	Yes	No	No
Password policy bind initialize and verify extended operation 1.3.18.0.2.12.79	The Password policy bind initialize and verify extended operation performs password policy bind initialization and verification for a specified user.	No	Yes	No	No
Password policy finalize and verify bind extended operation 1.3.18.0.2.12.80	The Password policy finalize and verify bind extended operation performs password policy post-bind processing for a specified user.	No	Yes	No	No

Table 28. OIDs for extended operations (continued)

Short name with OID	Description	Supported by the Administration Server	Supported by IBM Tivoli Directory Base Server v6.3	Supported by IBM Tivoli Directory Proxy Server v6.3	
				Without partitioned data	With partitioned data
Prepare Transaction extended operation 1.3.18.0.2.12.64	Using the prepare transaction extended operation the client requests the server to start processing the operations sent in a transaction.	No	Yes	Yes	Yes
Proxy Backend Server Resume Role Extended Operation 1.3.18.0.2.12.65	This extended operation enables a proxy server to resume the configured role of a back-end server in a distributed directory environment.	No	No	Yes	Yes
Quiesce or unquiesce replication context extended operation 1.3.18.0.2.12.19	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.	No	Yes	No	No
Replication error log extended operation 1.3.18.0.2.12.56	Maintenance of a replication error log.	No	Yes	No	No
Replication topology extended operation 1.3.18.0.2.12.54	Trigger a replication of replication topology-related entries under a given replication context.	No	Yes	No	No
ServerBackupRestore extended operation 1.3.18.0.2.12.81	Issues request to the administration server to backup a directory server's data and configuration files or restore directory server's data and configuration from an existing backup.	Yes	Yes	No	No
Start, stop server extended operations 1.3.18.0.2.12.26	Request to start, stop or restart an LDAP server.	Yes	Yes	Yes	Yes
Start TLS extended operation 1.3.6.1.4.1.1466.20037	Request to start Transport Layer Security.	Yes	Yes	Yes	Yes
Unique attributes extended operation 1.3.18.0.2.12.44	The unique attributes extended operation provides a list of all non-unique (duplicate) values for a particular attribute.	No	Yes	No	No
Update configuration extended operation 1.3.18.0.2.12.28	Request to update server configuration for IBM Tivoli Directory Server and Tivoli Directory Proxy Server.	Yes	Yes	Yes	Yes
User type extended operation 1.3.18.0.2.12.37	Request to get the User Type of the bound user.	Yes	Yes	Yes	Yes

OIDs for controls

The following table shows OIDs for controls.

Table 29. OIDs for controls

Short name with OID	Description	Supported by the Administration Server	Supported by IBM Tivoli Directory Base Server v6.3	Supported by IBM Tivoli Directory Proxy Server v6.3	
				Without partitioned data	With partitioned data
AES bind control 1.3.18.0.2.10.28	This control enables IBM Tivoli Directory Server to send updates to the consumer server with passwords already encrypted using AES.	No	Yes	No	No
Audit control 1.3.18.0.2.10.22	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.	Yes	Yes	Yes	Yes
Do not replicate control 1.3.18.0.2.10.23	This control can be specified on an update operation (add, delete, modify,modDn, modRdn).	No	Yes	No	No
Group authorization control 1.3.18.0.2.10.21	The control sends a list of groups that a user belongs to.	No	Yes	No	No
Ldap delete operation timestamp control 1.3.18.0.2.10.32	This control is used to send the modified timestamp values to a replica during a delete operation.	No	Yes	No	No
Limit Number of Attribute Values Control 1.3.18.0.2.10.30	This control limits the number of attribute values returned for an entry in a search operation.	No	Yes	Yes	Yes
Manage DSAIT control 2.16.840.1.113730.3.4.2	Causes entries with the "ref" attribute to be treated as normal entries, allowing clients to read and modify these entries. * In IBM Tivoli Directory Proxy Server (without partitioned data), even if this control is not included in the request the proxy server always sends the Manage DSAIT control to the back-end server.	No	Yes	Yes (*)	No
Modify groups only control 1.3.18.0.2.10.25	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 modfiy DN request does not need to exist on the server.	No	Yes	No	No

Table 29. OIDs for controls (continued)

Short name with OID	Description	Supported by the Administration Server	Supported by IBM Tivoli Directory Base Server v6.3	Supported by IBM Tivoli Directory Proxy Server v6.3	
				Without partitioned data	With partitioned data
No replication conflict resolution control 1.3.18.0.2.10.27	When present, a replica server accepts a replicated entry without trying to resolve any replication conflict for this entry.	No	Yes	No	No
Omit group referential integrity control 1.3.18.0.2.10.26	Omits the group referential integrity processing on a delete or modrdn request. When present on a delete or rename operation, the entry is deleted from or renamed in the directory, but the entry's membership is not removed or renamed in the groups in which the entry is a member.	No	Yes	No	No
Paged search results control 1.2.840.113556.1.4.319	Allows management of the amount of data returned from a search request.	No	Yes	Yes	Yes
Password policy request control 1.3.6.1.4.1.42.2.27.8.5.1	Password policy request or response	Yes	Yes	Yes	Yes
Persistent search control 2.16.840.1.113730.3.4.3	This control provide clients a means to receive notification of changes in the LDAP server.	No	Yes	No	No
Proxy authorization control 2.16.840.1.113730.3.4.18	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.	No	Yes	No	No
Refresh entry control 1.3.18.0.2.10.24	This control is returned when a target server detects a conflict during a replicated modify operation.	No	Yes	No	No
Replication supplier bind control 1.3.18.0.2.10.18	This control is added by the supplier, if the supplier is a gateway server.	No	Yes	No	No
Return deleted objects control 1.3.18.0.2.10.33	This control when included in a null base search requests, all entries in the database including those entries with attribute isDeleted set to TRUE are returned.	No	Yes	No	No
Server administration control 1.3.18.0.2.10.15	Allows an update operation by the administrator under conditions when the operation would normally be refused (server is quiesced, a read-only replica, etc.) * In IBM Tivoli Directory Proxy Server, this control is supported only for bind operations.	Yes	Yes	Yes (*)	Yes (*)

Table 29. OIDs for controls (continued)

Short name with OID	Description	Supported by the Administration Server	Supported by IBM Tivoli Directory Base Server v6.3	Supported by IBM Tivoli Directory Proxy Server v6.3	
				Without partitioned data	With partitioned data
Sorted search results control 1.2.840.113556.1.4.473	Allows a client to receive search results sorted by a list of criteria, where each criterion represents a sort key.	No	Yes	No	No
Subtree delete control 1.2.840.113556.1.4.805	This control is attached to a Delete request to indicate that the specified entry and all descendent entries are to be deleted.	No	Yes	No	No
Transaction control 1.3.18.0.2.10.5	Marks the operation as part of a transactional context. * In IBM Tivoli Directory Proxy Server, transactions are supported only when all updates target a single partition.	No	Yes	Yes (*)	Yes (*)
Virtual list view control 2.16.840.1.113730.3.4.9	This control extends the regular LDAP search operation and includes a server side sorting control.	No	Yes	No	No

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 IBM Tivoli Directory Server.

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:: IGJlZ2lucyB3aXRoIGEgc3BhY2U=
```

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 Higher Learning, c=US
cn: John E Doe
cn: John Doe
objectclass: person
sn: Doe

dn: cn=Bjorn L Doe, o=University of Higher Learning, c=US
cn: Bjorn L Doe
cn: Bjorn Doe
objectclass: person
sn: Doe

dn: cn=Jennifer K. Doe, o=University of Higher Learning, c=US
cn: Jennifer K. Doe
cn: Jennifer Doe
objectclass: person
sn: Doe
jpegPhoto:: /9j/4AAQSkZJRgABAAAAAQABAAD/2wBDABALD
 A4MChAODQ4SERATGCgaGBYWGDEjJR0oOjM9PDkzODdASFxOQ
 ERXRTc4UG1RV19iZ2hnPk1xeXBkeFxlZ2P/2wBDARESEhgVG
...
```

© Copyright IBM Corp. 2002, 2010

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 511 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: IBM Tivoli Directory Server 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:: V2hhdCBhIGNhcmVmdWwgcmVhZGVyIHlvd
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=sample" should be defined before attempting to load
 # this data.

 version: 1

 dn: o=sample
 objectclass: top
 objectclass: organization
 o: sample

 dn: ou=Austin, o=sample
 ou: Austin
 objectclass: organizationalUnit
 seealso: cn=Linda Carlesberg, ou=Austin, o=sample
```

This same file could be used without the version: 1 header information, as in previous releases of IBM Tivoli Directory Server:

```
 # IBM Directorysample LDIF file
 #
 # The suffix "o=sample" should be defined before attempting to load
 # this data.

 dn: o=sample
 objectclass: top
 objectclass: organization
 o: sample

 dn: ou=Austin, o=sample
 ou: Austin
 objectclass: organizationalUnit
 seealso: cn=Linda Carlesberg, ou=Austin, o=sample
```

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 30. IANA-defined character sets

Character	Locale					DB2 Code Page	
Set Name	HP-UX	Linux, Linux_390,	NT	AIX	Solaris	UNIX	NT
ISO-8859-1	X	X	X	X	X	819	1252
ISO-8859-2	X	X	X	X	X	912	1250
ISO-8859-5	X	X	X	X	X	915	1251
ISO-8859-6	X	X	X	X	X	1089	1256
ISO-8859-7	X	X	X	X	X	813	1253
ISO-8859-8	X	X	X	X	X	916	1255
ISO-8859-9	X	X	X	X	X	920	1254
ISO-8859–15	X	n/a	X	X	X		
IBM437	n/a	n/a	X	n/a	n/a	437	437
IBM850	n/a	n/a	X	X	n/a	850	850
IBM852	n/a	n/a	X	n/a	n/a	852	852
IBM857	n/a	n/a	X	n/a	n/a	857	857
IBM862	n/a	n/a	X	n/a	n/a	862	862
IBM864	n/a	n/a	X	n/a	n/a	864	864
IBM866	n/a	n/a	X	n/a	n/a	866	866
IBM869	n/a	n/a	X	n/a	n/a	869	869
IBM1250	n/a	n/a	X	n/a	n/a		
IBM1251	n/a	n/a	X	n/a	n/a		
IBM1253	n/a	n/a	X	n/a	n/a		
IBM1254	n/a	n/a	X	n/a	n/a		
IBM1255	n/a	n/a	X	n/a	n/a		
IBM1256	n/a	n/a	X	n/a	n/a		
TIS-620	n/a	n/a	X	X	n/a	874	874
EUC-JP	X	X	n/a	X	X	954	n/a
EUC-KR	n/a	n/a	n/a	X	X*	970	n/a
EUC-CN	n/a	n/a	n/a	X	X	1383	n/a
EUC-TW	X	n/a	n/a	X	X	964	n/a
Shift-JIS	n/a	X	X	X	X	932	943
KSC	n/a	n/a	X	n/a	n/a	n/a	949
GBK	n/a	n/a	X	X	n/a	1386	1386
Big5	X	n/a	X	X	X	950	950
GB18030	n/a	X	X	X	X		
HP15CN	X (with non-GB18030)						

* Supported at Solaris 7.

Notes:

1. The new Chinese character set standard (GB18030) is supported with appropriate patches available from www.sun.com and 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				

© Copyright IBM Corp. 2002, 2010

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 6.0 and above versions 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, the IPv6 URL format is not recognized by the non-IPv6 enabled clients and servers. For example:
   - Referals 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 version 6.0 and above supports scope:global only in IPv6 addresses on Linux systems.

© Copyright IBM Corp. 2002, 2010

Appendix F. Simple Network Management Protocol agent

The Simple Network Management Protocol (SNMP) agent services request for monitoring the state of the directory server and issues traps to the Network Management Station. Using the IBM Tivoli Directory Integrator assembly line with the SNMP agent, the performance and wellness information of the directory server can be reported and monitored. The Tivoli Directory Integrator assembly line will collect and report performance and wellness information like monitor search, root DSE search, and system information of the directory server it is monitoring. Directory server performance information will be logged periodically and will be made available in Extensible Markup Language (XML) format defined for Common Base Event (CBE).

Note:

- You must have IBM Tivoli Directory Integrator 7.1 installed to use the SNMP agent.

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.

To monitor IBM Tivoli Directory Server, you need to modify the properties and configuration files for the Simple Network Management Protocol (SNMP) agent.

Each directory server instance has a separate entry in the idssnmp.properties file. Configuration details will be unique for each directory server instance monitored by the idssnmp tool. This enables the idssnmp tool to monitor multiple directory server instances. A single instance of idssnmp tool that is launched will be able to monitor all the directory server instances mentioned in the idssnmp.properties file.

The idssnmp.properties file is encrypted by default once the idssnmp agent is started. This file is located in the *<TDSinstall_directory>*\idstools\snmp directory. The idssnmp.properties file contains the following:

```
server: <IP_address>
port: <port_number>
isSSL: True/False
ldapbindDN: <bind_DN>
bindDNpwd: <bind_pwd>
systemuser: <user_ID>
systemuserpwd: <user_pwd>
filterCacheActive: True/False
filterCacheThreshold: <Threshold Value in percentage>
pendingRequestsActive: True/False
pendingRequestsThreshold: <Threshold Value>
pendingRequestsSinceLastIntervalActive: True/False
pendingRequestsSinceLastIntervalThreshold: <Threshold Value>
activeConnectionActive: True/False
activeConnectionThreshold: <Threshold Value>
memoryUtilizationActive: True/False
memoryUtilizationThreshold: <Threshold Value in kilobytes>
cpuUtilizationActive: True/False
cpuUtilizationThreshold: <Threshold Value in percentage>
diskSpaceUtilizationActive: True/False
diskSpaceUtilizationThreshold: <Threshold Value in kilobytes>
```

© Copyright IBM Corp. 2002, 2010

```
replicationPendingChangeCountActive: True/False
replicationPendingChangeCountThreshold: <Threshold Value>
replicationStatusActive: True/False
trapForMessageId-<log_type>: <GLP...>
```

where:

server Represents the IP address of the monitored LDAP server.

port Represents the port on which the monitored LDAP server is running.

isSSL Indicates if the communication between the LDAP instance and the SNMP Agent is SSL encrypted.

ldapbindDN
Represents the bind DN.

bindDNpwd
Represents the bind password.

systemuser
Represents the system user id.

systemuserpwd
Represents the system user password.

filterCacheActive
If set to true, then a trap alert is generated when the percentage of search filter cache used exceeds the threshold limit.

filterCacheThreshold
Specifies the threshold value in percentage.

pendingRequestsActive
If set to true, then a trap alert is generated when the difference between number of operations requested and the number of operations completed (pending requests) exceeds the threshold limit.

pendingRequestsThreshold
Specifies the threshold value.

pendingRequestsSinceLastIntervalActive
If set to true, then a trap alert is generated when the number of pending requests since the last interval exceeds the threshold limit.

pendingRequestsSinceLastIntervalThreshold
Specifies the threshold value.

activeConnectionActive
If set to true, then a trap alert is generated when the number of active connections exceed the threshold limit.

activeConnectionThreshold
Specifies the threshold value.

memoryUtilizationActive
If set to true, then a trap alert is generated when the maximum system memory utilization exceeds the threshold limit.

memoryUtilizationThreshold
Specifies the threshold value in kilobytes.

cpuUtilizationActive
If set to true, then a trap alert is generated when the Maximum CPU utilization exceeds the threshold limit. This is applicable only for non-windows operating systems.

cpuUtilizationThreshold
Specifies the threshold value in percentage.

diskSpaceUtilizationActive
If set to true, then a trap alert is generated when the disk space utilization by the directory where DB2 database is stored exceeds the threshold limit.

diskSpaceUtilizationThreshold
Specifies the threshold value in kilobytes.

replicationPendingChangeCountActive
If set to true, then a trap alert is generated when the replication queue reaches a predefined threshold, for instance if the queue grows larger than 10000 entries.

replicationPendingChangeCountThreshold
Specifies the threshold value.

replicationStatusActive
If set to true, then a trap alert is generated if the current state of replication is incompatible, server is down, authentication has failed, or down level server is not supported.

trapForMessageId
Represents a list of message identifiers. The list will be a "," separated list of message identifiers. An SNMP trap will be generated in the event of a matching message identifier in the server log requested through an ldap extended operation. The log type describes the type of log required by the ldap extended operation. Each log type must be mentioned separately. For instance:

- trapForMessageId-slapd:
- trapForMessageId-audit:
- trapForMessageId-ibmdiradm:

If you want to send traps for all the messages generated in the log file, you can specify one of the following:

- TRAP_MAX – This will send traps for all (Information, Warning and Error) messages seen in the log files.
- TRAP_MID – This will send traps only for all Warning and Error messages seen in the log files.
- TRAP_MIN – This will send traps only for all Error messages seen in the log files.

Given below is an example of traps that can be set for log files slapd, audit, and ibmdiradm:

```
trapForMessageId-slapd: TRAP_MID
trapForMessageId-audit: TRAP_MAX
trapForMessageId-ibmdiradm: TRAP_MID
```

Note:

- TRAP_MIN and TRAP_MID are not valid values for trapForMessageId-audit. This is because the audit log contains only information messages.
- The traps sent by the idssnmp tool contain the OID 1.3.6.1.4.1.2.6.199.1.1.7. This OID holds the name of the instance to which the event corresponds to.

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 *<TDSinstall_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 IBM Tivoli Directory Server. If you want to monitor something other than the directory server, the SNMP agent for 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. 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".
3. 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.
4. 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.

5. 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.

Note:

- Load the following MIBS to your NMS:

  <*TDSinstall_directory*>\idstools\snmp\IBM-DIRECTORYSERVER-MIB
  <*TDSinstall_directory*>\idstools\snmp\INET-ADDRESS-MIB

The SNMP agent can be started by running the idssnmp script located in the <*TDSinstall_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*).

SNMP Logging

By default, the idssnmp application logs its data to the file /var/idsldap/V6.3/idssnmp.log on the UNIX platform and <*TDSinstall_directory*>\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 Tivoli Directory Integrator produces:

- ibmdi.log
- idssnmpinit.log

These files are produced because the Tivoli Directory Integrator 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:

- <*TDSinstall_directory*>/idstools/snmp/logs (UNIX)
- <*TDSinstall_directory*>\idstools\snmp\logs (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.

Note: The Tivoli Directory Integrator assembly line will periodically log directory server performance information in XML format defined for Common Base Event (CBE).

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.

If IBM Tivoli Directory Integrator fails, 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 AssemblyLine specified by the **-r** parameter and then exits.

9 License expired or invalid.

Appendix G. Active Directory synchronization

Active Directory synchronization is a tool for synchronizing users and groups between Microsoft Active Directory and an IBM Tivoli Directory Server instance.Synchronization is one-way, from Active Directory to IBM Tivoli Directory Server only.

Note: Synchronization of users and groups between Active Directory and IBM Tivoli Directory Server instances through IBM Tivoli Directory Proxy server is not supported.

Active Directory synchronization uses IBM Tivoli Directory Integrator for synchronizing the directories. You must have IBM Tivoli Directory Integrator installed before Active Directory synchronization can be run. IBM Tivoli Directory Integrator is used to run the configuration, and the IBM Tivoli Directory Integrator Administration and Monitoring Console is used to start, stop, restart, and monitor execution. You must have IBM Tivoli Directory Integrator installed before Active Directory synchronization can run.

Notes:

1. The Active Directory synchronization feature and IBM Tivoli Directory Integrator must be on the same computer as the associated directory server instance.
2. Active Directory synchronization synchronizes only users and groups. It does not synchronize other objects in the directory.
3. Active Directory synchronization does not synchronize nested organizational units (OUs).
4. Multiple attributes from Active Directory cannot be mapped to a single attribute in IBM Tivoli Directory Server.
5. Mapping of the userPassword attribute is not allowed. (User password data is not synchronized by this solution.)
6. Active Directory synchronization can synchronize users and groups from one or more user containers of Active Directory to a single OU of Tivoli Directory Server. However, it does not synchronize multiple user and group containers of Active Directory to multiple OUs of Tivoli Directory Server.
7. You can specify multiple user containers to synchronize with a single organizational unit (OU) in Tivoli Directory Server.

   Note: You can specify multiple user containers to synchronize with a single organizational unit (OU) in Tivoli Directory Server by using the semicolon (;) as a separator. (Other characters used as separators are not supported.) If you use the semicolon (;) separator, enclose the argument in quotation marks ("), as shown in the following example: "ou=SWUGroups,dc=adsync,dc=com;ou=STGGroups,dc=adsync,dc=com"

   The sAMAccountName attribute from Active Directory will be used to compose the $dn attribute in Tivoli Directory Server. Because the sAMAccountName attribute is unique in a domain, there will not be conflicts when synchronizing multiple Active Directory user containers to a single Tivoli Directory Server OU.

© Copyright IBM Corp. 2002, 2010

8. The solution currently supports an SSL connection to Active Directory, but does not support an SSL connection to IBM Tivoli Directory Server.
9. If you configure or change the administrator DN or password (or both) for the directory server instance after configuring Active Directory synchronization, you must reconfigure Active Directory synchronization.
10. If the user or group container names from Active Directory are changed dynamically (while Active Directory synchronization is running), you must reconfigure Active Directory synchronization with the new names or Active Directory synchronization will no longer run.
11. If you modify IBM Tivoli Directory Server users and groups by a means other than Active Directory synchronization, Active Directory synchronization might not work correctly.
12. Synchronization is one-way, from Active Directory to IBM Tivoli Directory Server only.
13. Only user entry attributes are synchronized.
14. The synchronization from Active Directory to Tivoli Directory Server cannot be guaranteed if a user modifies users or groups in a Tivoli Directory Server instance externally, that is, outside the synchronization solution.

Steps for using Active Directory synchronization

After you install IBM Tivoli Directory Server and IBM Tivoli Directory Integrator, create and configure a directory server instance, use the following steps to configure and use Active Directory synchronization:

1. If you use a copy of IBM Tivoli Directory Integrator that you did not install in the default path (on UNIX based systems: /opt/IBM/TDI/V7.1 and on Windows systems: C:\Program Files\IBM\TDI\V7.1), you must set the IDS_LDAP_TDI_HOME environment variable to the directory where you installed IBM Tivoli Directory Integrator V7.1.

   Note: On Windows systems, if there are spaces in this path, Active Directory synchronization will not work properly. Set the environment variable to a path with no spaces and no quotation marks, or use the short name when you specify the path.
2. Optionally, load the sample users.ldif and groups.ldif files into the Active Directory Server. Use the documentation for Active Directory Server.
3. Configure Active Directory synchronization using the IBM Tivoli Directory Server Configuration Tool or the **idsadscfg** command. This generates the adsync_private.prop and adsync_public.prop files. See the *IBM Tivoli Directory Server Version 6.3 Installation and Configuration Guide* for information.
4. Modify the adsync_public.prop file to customize optional attributes and SSL parameters, if needed. See "Files used by Active Directory synchronization" on page 527 for information. If you are using SSL, see "Configuring Active Directory synchronization to use an SSL connection to Active Directory" on page 531 for information.
5. Start Active Directory synchronization, using the **idsadsrun** command. You are asked if you want to fully synchronize, followed by real time synchronization, or only start real time synchronization. See "Running Active Directory synchronization" on page 530 for information.

   Changes made to the Active Directory entries will be read by the Active Directory synchronization tool, which identifies the changes.

Active Directory synchronization will synchronize any changes to IBM Tivoli Directory Server. IBM Tivoli Directory Integrator Administration and Monitoring Console can be used for further administration and monitoring.

Note:

- You can use either the Configuration Tool or the **idsadscfg** command to configure Active Directory synchronization.
- For information about configuring Active Directory synchronization with the Configuration Tool, see the *IBM Tivoli Directory Server Version 6.3 Installation and Configuration Guide*.
- For information about configuring Active Directory synchronization with the **idsadscfg** command, see the *IBM Tivoli Directory Server Version 6.3 Command Reference*.

Files used by Active Directory synchronization

The following files are used by Active Directory synchronization:

adsync.xml

The adsync.xml file is a configuration file. It provides the pre-configured assembly lines and connectors needed to perform the synchronization of Active Directory with IBM Tivoli Directory Server. It was originally created using the IBM Tivoli Directory Integrator Configuration Editor utility. There is only one adsync.xml file on a computer. This file cannot be changed. For customization, each IBM Tivoli Directory Server instance is configured to create properties files that define how the Active Directory synchronization works for the specific directory server instance.

Properties files

Two property files (adsync_private.prop and adsync_public.prop) are updated when a directory server instance is configured for Active Directory synchronization. The adsync.xml file contains a reference to these external properties files.

- The adsync_private.prop properties file is encrypted and the property values in this file can be modified only through the Configuration Tool or the **idsadscfg** command.

  Active Directory synchronization uses the following attributes as mandatory attributes to synchronize the IBM Tivoli Directory Server user entries:

  – $dn attribute (created from sAMAccountName
  – cn
  – sn
  – uid
  – objectClass

  The following is an example of an adsync_private.prop file:

```
Adhost1.AdGroupContainer:ou=SWUGroups,dc=adsynctest,dc=com
Adhost1.AdLdapLoginName:cn=administrator,cn=users,dc=adsynctest,dc=com
Adhost1.AdLdapPwd:dd06proxy
Adhost1.AdLdapSrchBase:dc=adsynctest,dc=com
Adhost1.AdLdapUrl:ldap://localhost:389
Adhost1.AdLdapUserContainer:ou=sales,dc=adsynctest,dc=com
Adhost1.IbmLdapGroupContainer:ou=groups,o=sample
Adhost1.IbmLdapUserContainer:ou=Austin,o=sample
IbmLdapLoginName:cn=root
```

```
IbmLdapPwd:sec001ret
IbmLdapSrchBase:ou=austin,o=sample
IbmLdapSuffix:o=sample
IbmLdapUrl:ldap://localhost:2389
```

The remaining attributes are optional and you can change these attributes (and the mapping) in the <*TDSinstance_home*>/idsslapd-<instance>/etc/tdisoldir/adsync_public.prop file.

- The adsync_public.prop file is an ASCII text file. It contains the following properties, which you can modify:

Table 31. Properties in the adsync_public.prop file

Property	Description	Example
AdDc1.AdSSL	A value of **true** indicates that SSL has been configured and will be used for the connection to Active Directory. A value of **false** indicates that the connection will not be over an SSL session. If the value is set to **true**, you must follow the instructions in "Configuring Active Directory synchronization to use an SSL connection to Active Directory" on page 531 to configure a key file on the IBM Tivoli Directory Integrator server before running the configuration.	true
TdsOptionalAttributes	List of attributes, separated by semicolon (;) characters, with optional syntax [attribute:(colon)attribute]. These attributes signify that each attribute name from Active Directory can be stored in IBM Tivoli Directory Integrator (Initial Work Entry) with a different name, which can further be used to map the attribute to a different attribute in IBM Tivoli Directory Server. The attributes are case sensitive. Multiple attributes cannot be mapped to a single IBM Tivoli Directory Server Attribute. The userPassword attribute cannot be mapped.	otherTelephone:telephoneNumber signifies that attribute otherTelephone from Active Directory is mapped to the corresponding attribute telephoneNumber in the IBM Tivoli Directory Server entry.

Table 31. Properties in the adsync_public.prop file (continued)

Property	Description	Example
LogLevel	Active Directory solution uses the LogLevel parameter to log the adsync configuration and execution details. The following log levels can be specified: • DEBUG • INFO • WARN • ERROR • FATAL	

The following is an example of an adsync_public.prop file:

```
Adhost1.OptionalAttributes:mail;displayName:cn;l:city;postalCode;init
ials:initials;givenName:givenName;streetAddress:street;st:st;departme
nt:departmentNumber;telephoneNumber:telephoneNumber;title:title;physi
calDeliveryOfficeName:roomNumber;otherTelephone:telephoneNumber;descr
iption:descriptionibmLdaphost1.OptionalAttributes:mail; cn; city:l;po
stalCode;initials;givenName;street;st;departmentNumber;telephoneNumbe
r;title;roomNumber;descriptionAdhost1.AdLdapSSL: false
```

adsync_cfg.xml

This file is used during configuration, either through the Configuration Tool or the **idsadscfg** command. This file contains an AssemblyLine that takes the configured parameters and creates the adsync_private.prop and adsync_public.prop properties files.

groups.ldif (sample file)

This file contains the sample groups to be added to the Active Directory setup. This is the sample data used to synchronize between Active Directory and IBM Tivoli Directory Server. Do not use this file as it is. It is modified during configuration with the user and group container and domain information you specify.

An example groups.ldif file is:

```
CN=SWUGroup1,OU=SWUGroups,dc=adsynctest,dc=com
objectClass=top
objectClass=group
cn=SWUGroup1
member=CN=lwood,ou=sales,dc=adsynctest,dc=com
member=CN=jdixon,ou=sales,dc=adsynctest,dc=com
member=CN=jcarroll,ou=sales,dc=adsynctest,dc=com
member=CN=twatson,ou=sales,dc=adsynctest,dc=com
member=CN=jsanchez,ou=sales,dc=adsynctest,dc=com
sAMAccountName=SWUGroup1
groupType=-2147483646

CN=SWUGroup2,OU=SWUGroups,dc=adsynctest,dc=com
objectClass=top
objectClass=group
cn=SWUGroup2
member=CN=amason,ou=sales,dc=adsynctest,dc=com
member=CN=swilson,ou=sales,dc=adsynctest,dc=com
member=CN=Elizabeth Brown,ou=sales,dc=adsynctest,dc=com
sAMAccountName=SWUGroup2
groupType=-2147483646
```

users.ldif (sample file)

This file contains the sample users to be added to the Active Directory

setup. This is the sample data used to synchronize between Active Directory and IBM Tivoli Directory Server. Do not use this file as it is. It is modified during configuration with the user and group container and domain information you specify.

An example users.ldif file is:

```
CN=lwood,ou=sales,dc=adsynctest,dc=com
cn=lwood
displayName=LORI H. WOOD
givenName=LORI
initials=H
objectClass=top
objectClass=person
objectClass=organizationalPerson
objectClass=user
physicalDeliveryOfficeName=8B001
name=lwood
sAMAccountName=lwood
sn=WOOD
userAccountControl=544
userPrincipalName=lwood@xyz.com

CN=pburns,ou=sales,dc=adsynctest,dc=com
cn=pburns
displayName=PATRICK BURNS
givenName=PATRICK
objectClass=top
objectClass=person
objectClass=organizationalPerson
objectClass=user
name=pburns
sAMAccountName=pburns
sn=BURNS
userAccountControl=544
userPrincipalName=pburns@xyz.com
```

adsync.log

Synchronization details (idsadsrun execution) are logged in adsync.log file which resides at the following location: <*TDS instance_home*>/idsslapd-<instance>/etc/tdisoldir/logs folder.

Details about logging can be configured using the LogLevel parameter in the <*TDS instance_home*>/idsslapd-<instance>/etc/tdisoldir/adsync_public.prop file. Default value of the LogLevel parameter is INFO which can be changed to DEBUG to get the debug logs. For further details refer to Tivoli Directory Integrator Logging.

Running Active Directory synchronization

Use the **idsadsrun** command to run Active Directory synchronization after it is configured. For information about running Active Directory synchronization with the **idsadsrun** command, see the *IBM Tivoli Directory Server Version 6.3 Command Reference*.

Note: If there are errors in the parameters specified for Active Directory, these errors will found at runtime and not during configuration. If errors are reported during runtime for the Active Directory parameters, you must reconfigure the Active Directory parameters correctly using the Configuration Tool (in the Active Directory synchronization: Active Directory details window) or the **idsadscfg** command.

Configuring Active Directory synchronization to use an SSL connection to Active Directory

You can use an SSL connection to Active Directory Server. (However, you cannot use an SSL connection to IBM Tivoli Directory Server.) To set up Active Directory synchronization to work with an SSL connection to Active Directory:

1. Configure Active Directory synchronization to use the appropriate port number. Additionally:
   - If you are using the Configuration Tool to configure, be sure to select the **Use SSL connection to Active directory** check box in the Active Directory synchronization: Instance Details window and to provide the correct port number in the **Host port** field in the Active Directory synchronization: Active Directory details window.
   - If you are using the **idsadscfg** command to configure, be sure to use the **-Z** flag.
2. Configure Active Directory to SSL using the self certificate as follows:
   a. Install Certificate Services on the Windows 2003 Server and an Enterprise Certificate Authority in the Active Directory Domain. Be sure that you install an Enterprise Certificate Authority.
   b. Start the Certificate Server Service. This creates a virtual directory in Internet Information Service (IIS) that enables you to distribute certificates.
   c. Create a Security (Group) Policy to direct Domain Controllers to get an SSL certificate from the Certificate Authority (CA).
   d. Open the Active Directory Users and Computers Administrative tool.
   e. Under the domain, right-click **Domain Controllers**. Select **Properties**.
   f. On the **Group Policy** tab, click to edit the Default Domain Controllers Policy.
   g. Go to **Computer Configuration –> Windows Settings –> Security Settings –> Public Key Policies**.
   h. Right-click **Automatic Certificate Request Settings**.
   i. Select **New**.
   j. Select **Automatic Certificate Request**.
   k. Run the wizard. Select the Certificate Template for a Domain Controller.
   l. Select your Enterprise Certificate Authority as the CA. Selecting a third-party CA works as well.
   m. Complete the wizard.

      All Domain Controllers now automatically request a certificate from the CA and support LDAP using SSL on port 636.
   n. Retrieve the Certificate Authority Certificate to the computer on which you installed Active Directory synchronization.

      Note: You must install IIS before installing the certificate server.
   o. Open a Web browser on the computer on which you installed Active Directory synchronization.
   p. Go to http://<*server_name*>/certsrv/ (where <*server_name*> is the name of the Windows 2003 server). You are asked to log in.
   q. Select the task **Retrieve the CA certificate or certificate revocation list** and click **Next**.
   r. The next page automatically highlights the CA certificate. Click **Download CA certificate**.

s. A new download window opens. Save the file to the hard drive.

3. Generate a jks file and configure Active Directory synchronization as follows:

   Create a certificate store using **keytool**. Use the keytool.exe file to create the certificate store and import the CA certificate into this store.

   Note: The keytool.exe file is located in the IBM Tivoli Directory Integrator directory in the _jvm\bin directory.

   Use the following command:

   ```
   _jvm\bin\keytool -import –file certnew.cer -keystore <keystore_name>.jks
       -storepass <password> -alias <keyalias_name>
   ```

   For example, assume the following values:

   - Keystorename = idi.jks
   - Password = secret Keyalias
   - name = AD_CA

   The command with these values is as follows. (Assume that you are in the C:\Program Files\IBM\TivoliDirectoryIntegrator directory.)

   ```
   _jvm\bin\keytool –import -file certnew.cer -keystore idi.jks
       -storepass secret -alias AD_CA
   ```

   To verify the contents of your keystore, type the following:

   ```
   _jvm\bin\keytool -list -keystore idi.jks -storepass secret
   ```

   This results in the following:

   ```
   Keystore type: jks
   Keystore provider: SUN
   Your keystore contains 1 entry:
   ad_ca, Mon Nov 04 22:11:46 MST 2002, trustedCertEntry,
   Certificate fingerprint (MD5): A0:2D:0E:4A:68:34:7F:A0:21:36:78:65:A7:1B:25:55
   ```

4. Configure Active Directory synchronization to use the keystore created in step 3, as follows:

   Edit the <*TDS instance_home*>\idsslapd-<instance>\etc\tdisoldir\ solution.properties file to set the keystore file location, keystore file password, and keystore file type. (In the current release, only the jks type is supported.)

   ```
   #server authentication
   #example
   javax.net.ssl.trustStore=c::\test\idi.jks
   javax.net.ssl.trustStorePassword=secret
   javax.net.ssl.trustStoreType=jks
   #client authentication
   #example
   javax.net.ssl.keyStore=c:\test\idi.jks
   javax.net.ssl.keyStorePassword=secret
   javax.net.ssl.keyStoreType=jks
   ```

5. Start Active Directory synchronization using the **idsadsrun** command. The solution will connect to Active Directory over SSL.

Appendix H. Additional information on password policy

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 or the password policy start time whichever is recent.
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.
pwdHistory	Directory String	Stores the history of previously used passwords. The password portion of this attribute is stored using the same encryption method as the userPassword is stored in. The passwords stored in this attribute are compared to the new userPassword that the user has entered.
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.
ibm-pwdIndividualPolicyDn	GeneralizedTime	DN of a password policy entry which can be associated with a user entry.
ibm-pwdGroupPolicyDn	GeneralizedTime	DN of a password policy entry which can be associated with a group entry.

© Copyright IBM Corp. 2002, 2010

Interoperability support for password policy response control

In order to return RFC compliant password policy response control for interoperability, user must set the environment variable, USE_OPENLDAP_PWDPOLICY_CONTROL, to YES. To do this issue the idsldapmodify command of the following format:

```
idsldapmodify -p port -D <adminDN> -w <adminPW>
dn: cn=Front End, cn=configuration
changetype: modify
add: ibm-slapdSetEnv
ibm-slapdSetEnv: USE_OPENLDAP_PWDPOLICY_CONTROL=YES
```

After setting the environment variable, restart the server to effect the changes made.

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:

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

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, 2009, with a password expiration policy of 186 days, query for entries for which the password was changed at least 186 days ago (February 22, 2009):

```
idsldapsearch –b "cn=users,o=sample" –s sub "(!(pwdChangedTime>=20090222000000Z))" <dn>
```

where the filter is equivalent to pwdChangedTime is less than or equal to midnight, February 22, 2009.

To query for locked accounts, use the pwdAccountLockedTime:

```
idsldapsearch –b "cn=users,o=sample" –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=sample" –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=sample
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=sample
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=sample
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=sample
changetype: modify
delete: pwdReset


idsldapmodify –D cn=root –w ? -k
dn: uid=user2,cn=users,o=sample
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=sample –w ?
dn: uid=user1,cn=users,o=sample
changetype: modify
replace: ibm-pwdAccountLocked
ibm-pwdAccountLocked: TRUE
```

To unlock the account:

```
idsldapmodify –D uid=useradmin,cn=users,o=sample –w ?
dn: uid=user1,cn=users,o=sample
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 multiple password policy attributes

For replication of multiple password policy attributes, the server's participating in replication must have the OID, LDAP_MULTIPLE_PASSWORD_POLICIES_OID, for the ibm-supportedcapabilities and ibm-enabledcapabilities attributes. The OID number for this capability is 1.3.18.0.2.32.77. If this OID is present in a server's root DSE, it indicates that the server can support multiple password policies as well as more granular password policy error messages.

Replicating password policy operational attributes

In a replication environment, certain password policy attributes must be replicated to the server's within the replication topology to have consistency in implementing password policy. For this, the global password policy entry "cn=pwdpolicy,cn=ibmpolicies" must be replicated to all the consumers of the cn=ibmpolicies subtree. To ensure that all servers have same password policy entries, the password policy entries must be defined under the cn=ibmpolicies entry and should be replicated to consumers.

The user-related elements of the password policy are stored in the operational attributes of entries. 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 x 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 inconstancies 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. However, before replicating the ibm-pwdAccountLocked attribute on to the consumer server, the LDAP_PASSWORD_POLICY_ACCOUNT_LOCKED_OID must be present for supported capabilities on the server. If LDAP_PASSWORD_POLICY_ACCOUNT_LOCKED_OID is not present on the consumer server replication must remove the attribute ibm-pwdAccountLocked before it sends any updates to servers.

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 I. Required attribute definitions for IBM Tivoli Directory Server

```
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
```

© Copyright IBM Corp. 2002, 2010

```
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-capabilitiessubentry'
DESC 'Names the ibm-capabilitiessubentry 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-filterAclEntry'
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( 'filterAclEntry'  'filterAclEntry' )
ACCESS-CLASS restricted
LENGTH 32700 )

attributetypes=( 1.3.18.0.2.4.2445
NAME 'ibm-filterAclInherit'
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( 'filterAclInherit'  'filterAclInherit' )
ACCESS-CLASS restricted
LENGTH 5 )

attributetypes=( 1.3.18.0.2.4.3238
NAME 'ibm-pwdPolicyStartTime'
DESC 'Specifies the time Password Policy was last turned on.'
ORDERING 2.5.13.28
SYNTAX 1.3.6.1.4.1.1466.115.121.1.24
SINGLE-VALUE
USAGE directoryOperation )
IBMAttributetypes=( 1.3.18.0.2.4.3238
DBNAME( 'pwdPolicyStartTim'  ' pwdPolicyStartTim '  )
ACCESS-CLASS normal
LENGTH 30 )

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( 'replicationPendin'  'replicationPendin' )
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
SAGE 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=( 1.3.18.0.2.4.3147
NAME 'ibm-slapdCachedAttributeAutoAdjust'
DESC 'Specifies if autonomic attribute caching is to be enabled.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
SINGLE-VALUE
USAGE directoryOperation )
IBMAttributetypes=(1.3.18.0.2.4.3147
DBNAME('slapdCachAttrAA'   'slapdCachAttrAA' )
ACCESS-CLASS normal
LENGTH 5)

attributetypes=( 1.3.18.0.2.4.3149
NAME 'ibm-slapdCachedAttributeAutoAdjustTime'
DESC 'Time to start autonomic attribute cache processing.
Values are in the form of Thhmmss where hh is hours, mm is minutes
and ss is seconds, using a 24 hour clock.'
EQUALITY 1.3.6.1.4.1.1466.109.114.2
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
SINGLE-VALUE
USAGE directoryOperation )
IBMAttributetypes=(1.3.18.0.2.4.3149
DBNAME('slapdCachAttrAAT'     'slapdCachAttrAAT' )
ACCESS-CLASS normal
LENGTH 7)

attributetypes=( 1.3.18.0.2.4.3148
NAME 'ibm-slapdCachedAttributeAutoAdjustTimeInterval'
DESC 'Specifies the time interval, in hours,
for autonomic attribute cache processing.'
EQUALITY 2.5.13.14
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE directoryOperation )
IBMAttributetypes=(1.3.18.0.2.4.3148
DBNAME('slapdCachAttrAAI'    'slapdCachAttrAAI' )
ACCESS-CLASS normal
LENGTH 11)

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 '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 )

ttributetypes=( 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=( 1.3.18.0.2.4.3287   NAME 'ibm-slapdGroupMembersCacheBypassLimit'
DESC 'Maximum number of members
that can be in a group in order for the group and its members to be cached
in the group members cache.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE directoryOperation )
IBMAttributetypes=( 1.3.18.0.2.4.3287
DBNAME( 'slapdGMCacheByp'  'slapdGMCacheByp'  )
ACCESS-CLASS normal
LENGTH 11)

attributetypes=( 1.3.18.0.2.4.3297
NAME NAME 'ibm-slapdGroupMembersCacheSize' DESC 'Maximum number of group
entries whose members should be cached.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE directoryOperation )
IBMAttributetypes=( 1.3.18.0.2.4.3297
DBNAME('slapdGMCacheSiz'  'slapdGMCacheSiz')
ACCESS-CLASS normal
LENGTH 11)

attributetypes=( 1.3.18.0.2.4.3399
NAME NAME 'ibm-slapdProxyMaxPendingOpsPerClient' DESC 'The maximum number of
operations that could be pending for a single backend server from a single
client connection. If not specified, defaults to 5'
EQUALITY 2.5.13.14
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE directoryOperation )
IBMAttributetypes=( 1.3.18.0.2.4.3399
DBNAME( 'ProxyMaxPendOps'  'ProxyMaxPendOps' )
ACCESS-CLASS critical
LENGTH 11)

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
DBNAME( 'enableEvntNotify'  'enableEvntNotify')
ACCESS-CLASS critical
LENGTH 5 )

attributetypes=( 1.3.18.0.2.4.xxxx
NAME 'ibm-slapdEnablePersistentSearch'
```

```
DESC 'If set to FALSE, the server will ignore non-critical
persistent search control sent with a search request and
will return LDAP_UNWILLING_TO_PERFORM for critical persistent
search control sent with a search request'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
SINGLE-VALUE
USAGE directoryOperation )
IBMAttributetypes=( 1.3.18.0.2.4.xxxx
DBNAME( 'enablePersistentSearch' )
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( 'slapdIncldeSchema'  'slapdIncldeSchema' )
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 ldapservicename 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-slapdLdapCrlHost'
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( 'LdapCrlHost' 'LdapCrlHost' )
ACCESS-CLASS critical
```

```
LENGTH 256 )

attributetypes=( 1.3.18.0.2.4.2407
NAME 'ibm-slapdLdapCrlPassword'
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-slapdLdapCrlPassword 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( 'CrlPassword' 'CrlPassword' )
ACCESS-CLASS critical )

attributetypes=( 1.3.18.0.2.4.2404
NAME 'ibm-slapdLdapCrlPort'
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 )
BMAttributetypes=( 1.3.18.0.2.4.2404
DBNAME( 'LdapCrlPort'  'LdapCrlPort' )
ACCESS-CLASS critical
LENGTH 11 )

attributetypes=( 1.3.18.0.2.4.2403
NAME 'ibm-slapdLdapCrlUser'
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-slapdLdapCrlUser 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( 'LdapCrlUser'  'LdapCrlUser' )
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 )
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 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)
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 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)
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. Minimum = 1 Maximum = 500'
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 | ssha | md5
| sha224 | sha256 | sha384 | sha512 | ssha224 | ssha256 | ssha384 | ssha512 }.
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 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
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 \24LANG)  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 preceeded 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 ommitted, 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 preceeded 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 ommitted, 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=( 1.3.18.0.2.4.3412
NAME'ibm-slapdTombstoneEnabled'
DESC 'Enable or Disable tombstones to record deleted entries.
The default value is FALSE'
```

```
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
SINGLE-VALUE
USAGE directoryOperation)
IBMAttributetypes=( 1.3.18.0.2.4.3412
DBNAME( 'slapdTSEnabled'  'slapdTSEnabled' )
ACCESS-CLASS normal
LENGTH 5 )

attributetypes=( 1.3.18.0.2.4.3413
NAME 'ibm-slapdTombstoneLifetime'
DESC 'Specifies the time in hours that tombstones may live.
When the time limit is reached the tombstones will be deleted
from the database.'
EQUALITY 2.5.13.14
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE directoryOperation )
IBMAttributetypes=( 1.3.18.0.2.4.3413
DBNAME( 'slapdTSLifetime'  'slapdTSLifetime' )
ACCESS-CLASS normal
LENGTH 11 )

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 accepable 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 )
IBMAttributetypes=( 1.3.18.0.2.4.2327
DBNAME( 'supportedReplicat'  'supportedReplicat' )
ACCESS-CLASS system
LENGTH 240 )

attributetypes=( 1.3.18.0.2.4.470
NAME 'IBMAttributeTypes'
DESC '  '
EQUALITY 2.5.13.2
```

```
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE
USAGE directoryOperation )
IBMAttributetypes=( 1.3.18.0.2.4.470
DBNAME( 'IBMAttributeTypes'  'IBMAttributeTypes' )
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 )
IBMAttributetypes=( 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 )
IBMAttributetypes=( 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 )
IBMAttributetypes=( 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 )
IBMAttributetypes=( 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 anotherdirectory 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 J. 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 on the second server instance 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. See the **idsgendirksf** command information in the *IBM Tivoli Directory Server Version 6.3 Command Reference* for more information about the idsgendirksf utility. The file is in the idsslapd-*instance_name*\etc directory on Windows systems, or in the idsslapd-*instance_name*/etc directory on AIX, Linux, and Solaris systems. (*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.

If you are creating a new directory server instance and you want it to be cryptographically synchronized with other directory server instances, use the following procedure:

1. On the original server, obtain the encryption salt value by performing the following search:

   ```
   ldapsearch -D adminDN -w adminPw -b "cn=crypto,cn=localhost"
     objectclass=* ibm-slapdCryptoSalt
   ```

2. A value similar to the following is returned:

   ```
   ibm-slapdCryptoSalt=d?TRm$'ucc5m
   ```

© Copyright IBM Corp. 2002, 2010

The part of the string after the equals to sign (=) is the encryption salt value. In this example, the encryption salt value is `d?TRm$'ucc5m`

3. Find the encryption seed value that was supplied when creating the original server.
4. Create the new server using one of the following:
   - Use the **Instance Administration Tool** and provide the encryption seed value from the original server in the **Encryption seed string** field and the encryption salt value from the original server in the **Encryption salt string** field.
   - Use the **idsicrt** command, and specify the **-e** *encryptionseed* and **-g** *encryptsalt* options.

Appendix K. 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.

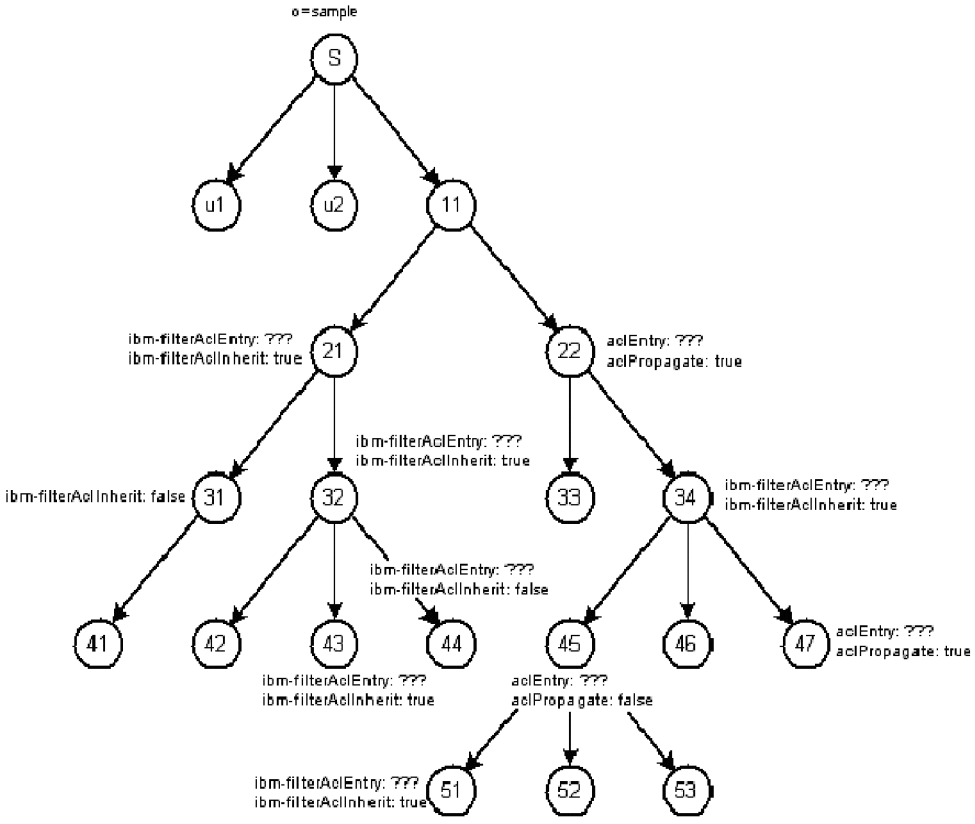


Figure 21. Filtered ACLs and non-filtered ACLs

```
LDIF File:

version: 1

dn: o=sample
objectclass: organization
objectclass: top
```

© Copyright IBM Corp. 2002, 2010

```
o: sample

dn: cn=User1, o=sample
cn: User1
sn: User
objectclass: person
objectclass: top
userPassword: User1

dn: o=Level11, o=sample
o: Level11
objectclass: organization
objectclass: top

dn: o=Level21, o=Level11, o=sample
o: Level21
objectclass: organization
objectclass: top
ibm-filterAclEntry: access-id:CN=USER1,o=sample:(o=Level32):normal:rwsc:
 sensitive:rsc:critical:rsc

dn: o=Level31, o=Level21, o=Level11, o=sample
o: Level31
objectclass: organization
objectclass: top
ibm-filterAclInherit: FALSE

dn:o=Level41, o=Level31, o=Level21, o=Level11, o=sample
o: Level41
objectclass: organization
objectclass: top

dn: o=Level32, o=Level21, o=Level11, o=sample
o: Level32
objectclass: organization
objectclass: top
ibm-filterAclEntry: access-id:CN=USER1,o=sample:(o=Level42):normal:rwsc:
 sensitive:rsc:critical:rsc
ibm-filterAclEntry: access-id:CN=USER1,o=sample:(o=Level43):normal:rwsc:
 sensitive:rwsc:critical:rsc
ibm-filterAclEntry: access-id:CN=USER2,o=sample:(o=Level44):normal:rwsc:
 sensitive:rsc:critical:rsc

dn: o=Level42, o=Level32, o=Level21, o=Level11, o=sample
o: Level42
objectclass: organization
objectclass: top

dn: o=Level43, o=Level32, o=Level21, o=Level11, o=sample
o: Level43
objectclass: organization
objectclass: top
ibm-filterAclEntry: access-id:CN=USER1,o=sample:(o=Level43):normal:rwsc:
 sensitive:rsc:critical:rwsc

dn: o=Level44, o=Level32, o=Level21, o=Level11, o=sample
o: Level44
objectclass: organization
objectclass: top
ibm-filterAclEntry: access-id:CN=USER1,o=sample:(o=Level44):normal:rwsc:
 sensitive:rsc:critical:rsc
ibm-filterAclInherit: FALSE

dn: cn=User2, o=sample
cn: User2
sn: User
objectclass: person
```

```
objectclass: top
userPassword: User2

dn: o=Level22, o=Level11, o=sample
o: Level22
objectclass: organization
objectclass: top
aclentry: access-id:CN=USER2,o=sample:normal:rsc:at.sn:deny:c:sensitive:
 c:critical:c

dn: o=Level33, o=Level22, o=Level11, o=sample
o: Level33
objectclass: organization
objectclass: top

dn: o=Level34, o=Level22, o=Level11, o=sample
o: Level34
objectclass: organization
objectclass: top
ibm-filterAclEntry: access-id:CN=USER2,o=sample:(o=Level34):normal:rwsc:
 sensitive:rsc:critical:rsc
ibm-filterAclEntry: access-id:CN=USER2,o=sample:(o=Level51):normal:rwsc:
 sensitive:rwsc:critical:rsc
ibm-filterAclEntry: access-id:CN=USER1,o=sample:(o=Level53):normal:rwsc:
 sensitive:rsc:critical:rsc
ibm-filterAclEntry: access-id:CN=USER2,o=sample:(o=Level46):normal:rwsc:
 sensitive:rsc:critical:rsc

dn: o=Level45, o=Level34, o=Level22, o=Level11, o=sample
o: Level45
objectclass: organization
objectclass: top
aclentry: access-id:CN=USER2,o=sample:normal:rwsc:sensitive:rsc:critical
 :rsc
aclpropagate: FALSE

dn: o=Level51, o=Level45, o=Level34, o=Level22, o=Level11, o=sample
o: Level51
objectclass: organization
objectclass: top
ibm-filterAclEntry: access-id:CN=USER2,o=sample:(o=Level51):normal:rwsc:
 sensitive:rsc:critical:rsc

dn: o=Level52, o=Level45, o=Level34, o=Level22, o=Level11, o=sample
o: Level52
objectclass: organization
objectclass: top

dn: o=Level53, o=Level45, o=Level34, o=Level22, o=Level11, o=sample
o: Level53
objectclass: organization
objectclass: top

dn: o=Level46, o=Level34, o=Level22, o=Level11, o=sample
o: Level46
objectclass: organization
objectclass: top

dn: o=Level47, o=Level34, o=Level22, o=Level11, o=sample
o: Level47
objectclass: organization
objectclass: top
aclentry: access-id:CN=USER2,o=sample:normal:rwsc: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=sample objectclass=*
 ibm-effectiveACL ibm-filterAclEntry
 ibm-filterACLInherit aclEntry aclPropagate

o=sample
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.

```
cn=User1,o=sample
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=Level11,o=sample
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=Level21,o=Level11,o=sample
ibm-filterACLInherit=TRUE
ibm-filterAclEntry=access-id:CN=USER1,o=sample:(o=Level32):normal:rwsc:
 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=sample
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=Level41,o=Level31,o=Level21,o=Level11,o=sample
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=Level32,o=Level21,o=Level11,o=sample
ibm-filterACLInherit=TRUE
ibm-filterAclEntry=access-id:CN=USER2,o=sample:(o=Level44):normal:rwsc:
 sensitive:rsc:critical:rsc
ibm-filterAclEntry=access-id:CN=USER1,o=sample:(o=Level43):normal:rwsc:
 sensitive:rwsc:critical:rsc
ibm-filterAclEntry=access-id:CN=USER1,o=sample:(o=Level42):normal:rwsc:
 sensitive:rsc:critical:rsc
ibm-effectiveACL=access-id:CN=USER1,o=sample:normal:rwsc: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=Level21,o=Level11,o=sample` entry. No other filtered ACLs apply to this entry, so the effective ACL is taken directly from the filtered ACL defined on the `o=Level21,o=Level11,o=sample` entry.

```
o=Level42,o=Level32,o=Level21,o=Level11,o=sample
ibm-effectiveACL=access-id:CN=USER1,o=sample:normal:rwsc:sensitive:rsc:
 critical:rsc
```

The filtered ACL defined on the `o=Level32,o=Level21,o=Level11,o=sample` entry is used to calculate the effective ACL for this entry.

```
o=Level43,o=Level32,o=Level21,o=Level11,o=sample
ibm-filterACLInherit=TRUE
ibm-filterAclEntry=access-id:CN=USER1,o=sample:(o=Level43):normal:rwsc:
 sensitive:rsc:critical:rwsc
ibm-effectiveACL=access-id:CN=USER1,o=sample:normal:rwsc:sensitive:rwsc:
 critical:rwsc
```

This entry is a simple example of how filtered ACLs accumulate. The filtered ACL defined on the `o=Level32,o=Level21,o=Level11,o=sample` entry is combined with the filtered ACL defined on the `o=Level43,o=Level32,o=Level21,o=Level11,o=sample` entry to give read, write, search and compare access to all three classes of attributes for user 1.

```
o=Level44,o=Level32,o=Level21,o=Level11,o=sample
ibm-filterACLInherit=FALSE
ibm-filterAclEntry=access-id:CN=USER1,o=sample:(o=Level44):normal:rwsc:
 sensitive:rsc:critical:rsc
ibm-effectiveACL=access-id:CN=USER1,o=sample:normal:rwsc: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=Level32,o=Level21,o=Level11,o=sample` 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=sample 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,o=sample:normal:rwsc:sensitive:rsc:
 critical:rsc
ibm-effectiveACL=access-id:CN=USER1,o=sample:normal:rwsc:sensitive:rsc:
 critical:rsc
```

```
cn=User2,o=sample
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=sample
aclPropagate=TRUE
aclEntry=access-id:CN=USER2,o=sample:sensitive:c:at.sn:deny:c:normal:
 rsc:critical:c
ibm-effectiveACL=access-id:CN=USER2,o=sample: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=sample
aclPropagate=TRUE
aclEntry=access-id:CN=USER2,o=sample:sensitive:c:at.sn:deny:c:normal:
 rsc:critical:c
ibm-effectiveACL=access-id:CN=USER2,o=sample: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=sample entry propagating down to the o=Level33,o=Level22,o=Level11,o=sample entry. This propagation occurs because the aclPropagate attribute was set to TRUE in the o=Level22,o=Level11,o=sample entry.

```
o=Level34,o=Level22,o=Level11,o=sample
ibm-filterACLInherit=TRUE
ibm-filterAclEntry=access-id:CN=USER2,o=sample:(o=Level46):normal:rwsc:
 sensitive:rsc:critical:rsc
ibm-filterAclEntry=access-id:CN=USER1,o=sample:(o=Level53):normal:rwsc:
 sensitive:rsc:critical:rsc
ibm-filterAclEntry=access-id:CN=USER2,o=sample:(o=Level51):normal:rwsc:
 sensitive:rwsc:critical:rsc
ibm-filterAclEntry=access-id:CN=USER2,o=sample:(o=Level34):normal:rwsc:
 sensitive:rsc:critical:rsc
ibm-effectiveACL=access-id:CN=USER2,o=sample:normal:rwsc: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=sample 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=Level45,o=Level34,o=Level22,o=Level11,o=sample
aclPropagate=FALSE
aclEntry=access-id:CN=USER2,o=sample:sensitive:rsc:normal:rwsc:critical:
 rsc
ibm-effectiveACL=access-id:CN=USER2,o=sample:critical:rsc:normal:rwsc:
 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=Level51,o=Level45,o=Level34,o=Level22,o=Level11,o=sample
ibm-filterACLInherit=TRUE
ibm-filterAclEntry=access-id:CN=USER2,o=sample:(o=Level51):normal:rwsc:
 sensitive:rsc:critical:rsc
ibm-effectiveACL=access-id:CN=USER2,o=sample:normal:rwsc:sensitive:rwsc:
 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=Level34,o=Level22,o=Level11,o=sample` entry and the `o=Level51,o=Level45,o=Level34,o=Level22,o=Level11,o=sample` entry.

```
o=Level52,o=Level45,o=Level34,o=Level22,o=Level11,o=sample
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 `Level45` 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=Level53,o=Level45,o=Level34,o=Level22,o=Level11,o=sample
ibm-effectiveACL=access-id:CN=USER1,o=sample:normal:rwsc:sensitive:rsc:
 critical:rsc
```

The effective ACL for this entry is the filtered ACL defined in the `o=Level34,o=Level22,o=Level11,o=sample` entry.

```
o=Level46,o=Level34,o=Level22,o=Level11,o=sample
ibm-effectiveACL=access-id:CN=USER2,o=sample:normal:rwsc:sensitive:rsc:
 critical:rsc
```

The effective ACL for this entry is the propagated non-filtered ACL defined on the `o=Level34,o=Level22,o=Level11,o=sample` entry.

```
o=Level47,o=Level34,o=Level22,o=Level11,o=sample
aclPropagate=TRUE
aclEntry=access-id:CN=USER2,o=sample:sensitive:rsc:normal:rwsc:critical:
 rsc
ibm-effectiveACL=access-id:CN=USER2,o=sample:critical:rsc:normal:rwsc:
 sensitive:rsc
```

This entry has an explicit non-filtered ACL defined, so the effective ACL is taken from the explicitly defined ACL.

Appendix L. 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. If you use the command line to update values, you must request the *ldapexop -op readconfig* option. For more information, see the **idsldapexop** command information in the *IBM Tivoli Directory Server Version 6.3 Command Reference*.

cn=Configuration

- ibm-slapdadmindn
- ibm-slapdAdminGroupEnabled
- ibm-slapdadminpw
- ibm-slapdDerefAliases
- ibm-slapdpwencryption
- ibm-slapdsizelimit
- ibm-slapdtimelimit
- ibm-slapdAdminRole
- ibm-slapdPtaEnabled

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

© Copyright IBM Corp. 2002, 2010

- ibm-slapdaclcachesize
-
- 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
- ibm-slapdMaxTimeBetweenPrepareAndCommit

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
- ibm-slapdCachedAttributeAutoAdjust
- ibm-slapdCachedAttributeAutoAdjustTime
- ibm-slapdCachedAttributeAutoAdjustTimeInterval
- ibm-slapdNumRetry
- ibm-slapdGroupMembersCacheSize
- ibm-slapdGroupMembersCacheBypassLimit

- ibm-slapdDbUserPW
- ibm-slapdTombstoneEnabled
- ibm-slapdTombstoneLifetime

cn=change log, cn=RDBM Backends, cn=IBM Directory, cn=Schemas, cn=Configuration

- ibm-slapdCachedAttribute
- ibm-slapdCachedAttributeSize
- ibm-slapdCachedAttributeAutoAdjust
- ibm-slapdCachedAttributeAutoAdjustTime
- ibm-slapdCachedAttributeAutoAdjustTimeInterval

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
- passwordMaxConsecutiveRepeatedChars
- passwordMinDiffChars

cn=Replication, cn=configuration

- ibm-slapdReplConflictMaxEntrySize
- ibm-slapdMaxPendingChangesDisplayed
- ibm-slapdReplMaxErrors
- ibm-slapdReplContextCacheSize
- ibm-slapdReplRestrictedAccess
- ibm-slapdEnableConflictResolutionForGroups

cn=VirtualListView, cn=Configuration

- ibm-slapdVLVEnabled
- ibm-slapdMaxVLVBeforeCount

cn=Persistent Search, cn=Configuration

- ibm-slapdMaxPersistentSearches
- ibm-slapdEnablePersistentSearch

cn=RDBM Backup, cn=Configuration

- ibm-slapdBackupLocation
- ibm-slapdBackupAt
- ibm-slapdBackupEvery
- ibm-slapdBackupOnline

- ibm-slapdBackupEnabled
- ibm-slapdBackupChangelog

cn=Master Server, cn=Configuration

- ibm-slapdMasterDN
- ibm-slapdMasterPW

Appendix M. IBM Tivoli Directory Server backup and restore

Introduction

Tivoli Directory Server provides multiple methods for backing up and restoring directory server instance information. There are methods that back up the complete information for a directory server instance, and methods that back up only the data in the database. This appendix contains information about methods that back up only the data in the database, which includes DB2 backup and restore commands to back up and restore the DB2 data. For more information regarding methods for backing up and restoring directory server instance information, see Chapter 16, "Directory Server backup and restore," on page 349

IBM Tivoli Directory Server uses IBM DB2 relational database to store directory information. To ensure the availability of directory information and to recover critical data from loss or corruption, it is necessary for Tivoli Directory Server directory administrators to design a backup and restore strategy for their Tivoli Directory Server environments.

DB2 provides online backup feature, which allows taking backup of a database while the database is being accessed by other applications, such as Tivoli Directory Server. Before considering a backup and restore strategy that includes online backup, be aware that performing an online backup consumes a significant amount of DB2 resources.

This appendix starts with a description of the Tivoli Directory Server database and tablespace definitions. Individual sections describe alternatives to Tivoli Directory Server backup and restore procedure that include DB2 offline and online backup, DB2 offline restore, and redirected restore.

Tivoli Directory Server directory schema and database definitions

Tivoli Directory Server uses directory schema files to define the underlying DB2 directory database, which is used to store data. In order to recover a data stored on Tivoli Directory Server, you are required to back up the files containing the Tivoli Directory Server directory configuration and schema and the DB2 databases.

Tivoli Directory Server directory schema

By default, Tivoli Directory Server maintains its schema files in the etc directory under the directory server instance owner's home directory. For example, for the **ldapdb2** instance owner, the schema file location would be:

```
/home/ldapdb2/idsslapd-ldapdb2/etc
```

Note: You can also specify a different location for the schema files during instance creation provided the instance owner has write access on the directory.

Each time you start the server, it checks the schema files, validates them against the underlying DB2 database, and checks that the database is correctly configured to support the schema.

A new instance can be configured to have the same schema by copying the schema files to the new server instance owner's <inst_owner_home>/idsslapd-

© Copyright IBM Corp. 2002, 2010

<inst_name>/etc directory. For example, to back up the schema files on AIX, where ldapdb2 is the Tivoli Directory Server instance being used and the directory /safeplace/etc is the location where the schema files are to be saved, issue the following command:

```
cp /home/ldapdb2/idsslapd-ldapdb2/etc/* /safeplace/etc
```

To set up a new instance with the same schema, issue the following command:

```
cp /safeplace/etc/* /home/<newuser>/idsslapd-<new_user>/etc
```

Tivoli Directory Server directory database and tablespaces

A tablespace is a storage structure where the actual data underlying database objects can be stored. DB2 supports two types of tablespaces:

- **System Managed Space (SMS)** - In SMS, the operating system's file system manager allocates and manages the space where the table is stored. This is suitable for tablespaces that grow and shrink rapidly.
- **Database Managed Space (DMS)** - In DMS, the database manager controls the storage space.

During the database configuration, by default, a database with DMS tablespaces is created. The locations for tablespaces can be explicitly provided by the administrator. If the specified location is a file system, then DMS cooked tablespace is created. In case the specified location is a raw/block device, DMS raw tablespace is created. Raw devices are devices where no file system is installed. For information on how to create database with SMS or DMS tablespaces and the default values for various parameters, see the **idscfgdb** command information in the *IBM Tivoli Directory Server Version 6.3 Command Reference*.

Note:

- DB2 by default creates 3 tablespaces- USERSPACE1, SYSCATSPACE, and TEMPSPACE1. Tivoli Directory Server creates an additional tablespace called LDAPSPACE,
- USERSPACE1 and LDAPSPACE tablespaces store the Tivoli Directory Server data.
- Administrators can opt to create either DMS or SMS type tablespace for USERSPACE1 and LDAPSPACE.

Since DB2 backup and restore can be done at the database level, the tablespace level, or both levels, it is important to understand the underlying structure to determine which backup and restore method might be best for different Tivoli Directory Server environments. In general, it is advisable that users do not use DB2 backup and restore at the tablespace level for reasons listed.

In the examples, ldapdb2 is used as the database name. You can use the ***db2 list database directory*** and ***db2 list tablespace show detail*** commands to find the database and tablespace information for your environment.

You can view the tablespaces by using the following DB2 commands run under the context of the DB2 instance owner. In this example, ldapdb2 is used:

```
db2 connect to <databasename>
db2 list tablespaces
```

The following examples show tablespace output for the Tivoli Directory Server directory database on an AIX, Linux, or Solaris system:

```
Tablespaces for Current Database

Tablespace ID                        = 0
Name                                 = SYSCATSPACE
Type                                 = System managed space
Contents                             = All permanent data. Regular table space.
State                                = 0x0000
   Detailed explanation:
     Normal

Tablespace ID                        = 1
Name                                 = TEMPSPACE1
Type                                 = System managed space
Contents                             = System Temporary data
State                                = 0x0000
   Detailed explanation:
     Normal

Tablespace ID                        = 2
Name                                 = USERSPACE1
Type                                 = Database managed space
Contents                             = All permanent data. Large table space.
State                                = 0x0000
   Detailed explanation:
     Normal

 Tablespace ID                        = 3
 Name                                 = LDAPSPACE
 Type                                 = Database managed space
 Contents                             = All permanent data. Large table space.
 State                                = 0x0000
   Detailed explanation:
     Normal
```

Tivoli Directory Server data is stored in two separate tablespaces: USERSPACE1 and LDAPSPACE. By default, there is only one container or directory for each tablespace. To view the details about the USERSPACE1 tablespace, issue the following DB2 command:

```
db2 list tablespace containers for 2
```

Given below are examples of output for the Tivoli Directory Server instance ldapdb2:

Output for tablespace container 2 in case of DMS cooked tablespace:

For non-windows:

```
Container ID = 0
Name = /home/ldapdb2/ldapdb2/NODE0000/SQL00001/USPACE
Type = File
```

For windows:

```
Container ID = 0
Name = C:\ldapdb2\NODE0000\SQL00001\USPACE
Type = File
```

Output for tablespace container 2 in case of DMS raw tablespace:

For linux:

```
Container ID = 0
Name = /dev/raw/raw1
Type = disk
```

For windows:

```
Container ID = 0
Name = \H
Type = disk
```

Output for tablespace container 2 in case of SMS tablespace:

For non-windows:

```
Container ID = 0
Name = /home/ldapdb2/ldapdb2/NODE0000/SQL00001/SQLT0002.0
Type = path
```

For windows:

```
Container ID = 0
Name = C:\ldapdb2\NODE0000\SQL00001\SQLT0002.0
Type = path
```

The default container or directory that DB2 uses for tablespace 2 (USERSPACE1) is /home/ldapdb2/ldapdb2/NODE0000/SQL00001/USPACE. It contains all of the ldapdb2 database tables, which have rows that fit in a 4K page size. This includes the attribute tables that are used for fast DB2 lookups. Tablespace 3 (LDAPSPACE) contains the remainder of the database tables that require a 32K page size. This includes the ldap_entry table, which contains the majority of the Tivoli Directory Server directory data and the replication tables. To view the tablespace container information for the LDAPSPACE tablespace, issue the following DB2 command:

```
db2 list tablespace containers for 3
```

Given below are examples of output for the Tivoli Directory Server instance ldapdb2:

Output for tablespace container 3 in case of DMS cooked tablespace:

For non-windows:

```
Container ID = 0
Name = /home/ldapdb2/ldap32kcont_ldapdb2/ldapspace
Type = File
```

For windows:

```
Container ID = 0
Name = C:\ldapdb2\ldap32kcont_ldapdb2\ldapspace
Type = File
```

Output for tablespace container 3 in case of DMS raw tablespace:

For linux:

```
Container ID = 0
Name = /dev/raw/raw2
Type = disk
```

For windows:

```
Container ID = 0
Name = \K
Type = disk
```

Output for tablespace container 3 in case of SMS tablespace:

For non-windows:

```
Container ID = 0
Name = /home/ldapdb2/ldap32kcont_ldapdb2
Type = path
```

For windows:

```
Container ID = 0
Name = C:\ldapdb2\ldap32kcont_ldapdb2
Type = path
```

It is important to notice that data in Tivoli Directory Server is spread out between tablespace 2 and tablespace 3, and both tablespaces need to be accessed for most of the single Tivoli Directory Server operations. In case of a search operation, the attribute tables in tablespace 2 are used to find the entries that match the given

criteria but the entry information is returned from the ldap_entry table in tablespace 3. For an update operation, the attribute tables in tablespace 2 and the ldap_entry (and possibly the replication tables) in tablespace 3 must be updated. For this reason, users should perform backup and restore only at the database level, so that the related sets of data are kept together. If the related sets of data are not kept together, recovering to a point in time where all of the data is consistent would be unlikely.

Tivoli Directory Server change log database and tablespaces

In Tivoli Directory Server 6.0 and above versions, the change log feature records all updates to the directory in a separate change log DB2 database, which is different from the database that holds the Tivoli Directory Server directory information tree (DIT). The change log database can be used by other applications to query and track LDAP updates. By default, the change log function is disabled. The change log function should be configured only if needed because it reduces update performance due to the additional logging overhead. A way to check for existence of the change log function is to look for the suffix CN=CHANGELOG. If it exists, the change log function is enabled.

When Tivoli Directory Server creates a database for the change log, it uses the ***db2 create database*** command to create a database named ldapclog. IBM Tivoli Directory Server creates this database with four SMS tablespaces identical to the ldapdb2 database.

You can view the tablespaces by using the following DB2 commands run under the context of the DB2 instance owner. In this example, ldapdb2 is used.

```
db2 connect to ldapclog
db2 list tablespaces
```

It is important to notice that the Tivoli Directory Server directory information is stored in a database (ldapdb2) which is different to the change log database (ldapclog). In order to keep related sets of data together, care must be taken to ensure they are backed up and restored in a consistent manner.

Overview of backup and restore procedures for LDAP

In a Tivoli Directory Server environment, you can back up and restore a database using the DB2 commands, Tivoli Directory Server backup and restore commands, and Tivoli Directory Server tools. These options have their advantages and disadvantages.

DB2 backup and restore are built-in commands available in DB2 to back up and restore databases. The advantages of using db2 backup and db2 restore commands or the dbback and dbrestore commands is that the DB2 configuration parameters and database optimizations parameters are preserved for the backed-up database. In addition, the restored database has the same performance tuning specifications as that of the backed-up database. One of the disadvantages of using db2 backup and restore is that the database backed-up on one hardware platform cannot be restored on a different platform. For example, a database backed up on AIX system cannot be restored on a Solaris system. In addition, database backed up on a Tivoli Directory Server version cannot be restored on a different Tivoli Directory Server version. You are also required to use the same version of DB2 for both the db2 backup and db2 restore operations. See *DB2 Administration Guide* to know more about DB2 backup and restore procedures. See *DB2 Command Reference* to know

more about the DB2 commands. The DB2 Administration Guide and the Command Reference are part of the online library installed with DB2 and Tivoli Directory Server.

The Tivoli Directory Server commands, idsdbback and idsdbrestore, for back up and restore of databases use the DB2 backup and restore commands. In addition to the features provide by the DB2 backup and restore commands, idsdbback and idsdbrestore also backs up and restores Tivoli Directory Server configuration and schema files. However, it is important to note that idsdbback does not support DB2 online backup in Tivoli Directory Server V6.0. The idsbback command can be used only when Tivoli Directory Server is not running. For more information about the use of these commands, see the Server utilities section in *IBM Tivoli Directory Server Version 6.3 Command Reference*.

An alternative to the DB2 and Tivoli Directory Server backup and restore commands are Tivoli Directory Server tools, such as the LDAP Data Interchange Format (LDIF) export and import commands, db2ldif and ldif2db. These tools can be used across dissimilar hardware platforms but the process is slower. These tools do not preserve the DB2 configuration parameters and database optimizations parameters. For more information about the use of these commands, see the Server utilities section in *IBM Tivoli Directory Server Version 6.3 Command Reference.*

Note: If you restore over an existing database, any performance tuning tasks on that existing database is lost. You must check all DB2 configuration parameters after performing a restore. Also, if you do not know whether a db2 reorgchk was performed before the database was backed up, run db2 reorgchk after the restore.

Examples of offline backup and restore procedure for a directory database

The DB2 commands to perform offline backup and restore operations for a directory database, ldapdb2, are as follows:

```
su – ldapdb2
db2start
db2 force applications all
db2 backup db ldapdb2 to <directory_or_device>
db2 restore db ldapdb2 from <directory_or_device> replace existing
```

where, *directory_or_device* is the name of a directory or device where the backup is stored.

The DB2 commands to perform offline backup and restore operations for the change log database are as follows:

```
su – ldapdb2
db2start
db2 force applications all
db2 backup db ldapclog to <directory_or_device>
db2 restore db ldapclog from <directory_or_device> replace existing
```

The most common error that occurs while restoring is a file permission error. Following are some reasons why this error might occur:

- The DB2 instance owner does not have permission to access the specified directory and file. One way to solve this is to change directory and file ownership to the DB2 instance owner. For example, enter the following command:

  ```
  chown ldapdb2 <fil_or_dev>
  ```

- The backed-up database is distributed across multiple directories, and those directories do not exist on the target system of the restore. Distributing the database across multiple directories is accomplished with a redirected restore. To solve this problem, either create the same directories on the target system or perform a redirected restore to specify the proper directories on the new system. When creating the same directories, ensure that the owner of the directories is the DB2 instance owner.

Replication considerations

Backup and restore operations may be used to initially synchronize a consumer with a supplier or whenever the supplier and consumer get out of sync. A consumer can get out of sync if it is not defined to the supplier or is not reachable by the supplier. In this case, the supplier does not know about the consumer and does not save updates on a propagation queue for that consumer.

Overview of online backup and restore procedures for Tivoli Directory Server

When a Tivoli Directory Server database is created, only circular logging is enabled for it. This means that log files are reused in a circular fashion, and are not saved or archived. With circular logging, rollforward recovery is not possible but crash recovery is possible. The directory server must be stopped and should be offline when backups are taken. Before performing online backups, administrators must plan a strategy to manage the DB2 log files that will be needed to perform a restore from an online backup.

Log management

When log archiving is configured for the database, rollforward recovery is possible. This is because of the following reasons:

- The logs record changes to the database during and after the backups are taken.
- Log files are kept even after they contain committed and externalized data referred to as “inactive” logs.

To configure log archiving, change the **logarchmeth1** database parameter from OFF to an appropriate value by selecting the archiving mode desired. The possible values for mode are:

LOGRETAIN
: In this mode, inactive log files are never overwritten. This means that inactive logs must be moved to an archive location to avoid running out of disk space for primary logs. The database configuration specifies the number of active primary log files and active secondary log files that can be created. When LOGRETAIN is set, DB2 will first fill up the primary logs, and then if the first primary log is still active, DB2 will create secondary logs. If the number of primary and secondary logs have been created and filled has reached the maximum limit before the first primary log becomes inactive, a “log full” condition will occur. As primary logs become inactive, DB2 will create additional primary logs as needed. In the LOGRETAIN mode, it is important to monitor the disk space available for the log files because if the disk fills up, directory updates will not be possible until the condition is rectified.

USEREXIT
: In this mode, archival and retrieval of logs is performed by a user-supplied user exit program called **db2uext2**. The user exit program is called to copy

a log file to an archive location as soon as the log file is full. This allows DB2 to rename and reuse the file once it becomes inactive. During recovery operations, after restoring a database from a backup, when inactive log files are required, DB2 will call the user exit program to retrieve the necessary logs from the archive location.

DISK:*directory*
With this setting, log management is performed using an algorithm similar to the USEREXIT mode. The difference between the two modes, USEREXIT and DISK:*directory*, is that instead of calling the user exit program, DB2 will automatically archive the logs from the active log directory to the specified directory. During recovery, DB2 retrieves these logs from that location.

TSM:[management class name]
This mode is similar to the USEREXIT mode except that logs will be automatically archived on the local Tivoli Storage Manager server. The management class name parameter is optional. If not specified, the default management class is used.

VENDOR:*library*
In this mode, logging operates in a mode similar to USEREXIT except that the specified vendor library is invoked to archive or retrieve the logs.

When this parameter is configured, the database is enabled for rollforward recovery. After logarchmeth1 is set to for log archiving, a full offline backup of the database must be made for the "backup pending state" to be satisfied so that the database can be used. To check if the database is in "backup pending state", look at the "Backup pending" value returned from the following DB2 command, which could either be YES or NO.

```
db2 get db config for ldapdb2
```

When the database is recoverable, the backups of the database can be completed online. Rollforward recovery reapplies the completed units of work recorded in the logs to the restored database, tablespace, or tablespaces. You can specify rollforward recovery either to the end of the logs or to a particular point in time.

A recovery history file is created with each database and this file is updated automatically with summary information whenever you carry out a backup or restore of a full database or tablespace. The recovery history file is a useful tracking mechanism for restore activity within a database. This file is created in the same directory as the database configuration file. It is automatically updated whenever one of the following activities is performed:

- Backup of a database or tablespace
- Restore of a database or tablespace
- Rollforward of a database or tablespace
- Alter of a tablespace
- Quiesce of a tablespace
- Rename of a tablespace
- Load of a table
- Drop of a table
- Reorganization of a table
- Update of table statistics

For information about existing backed-up databases, enter the following DB2 command:

```
db2 list history backup all for db ldapdb2
```

The database configuration file contains the logarchmeth1 and other parameters related to rollforward recovery. In some cases, since the default parameter settings will not work well, you may need to change some of these default settings for your setup. See the DB2 Administration Guide for detailed information about configuring these parameters in DB2.

Primary logs (logprimary)
This parameter specifies the number of primary logs that might be active at a given time.

Secondary logs (logsecond)
This parameter specifies the number of secondary log files that might be created if all active primary logs are full.

Log size (logfilsiz)
This parameter determines the number of pages for each of the configured logs. A page is 4 KB in size.

Log buffer (logbufsz)
This parameter enables you to specify the amount of database shared memory to use as a buffer for log records before writing these records to disk.

Number of commits to group (mincommit)
This parameter enables you to delay the writing of log records to disk until a minimum number of commits have been performed.

New log path (newlogpath)
You can change the location where active logs and future archive logs are placed by changing the value for this configuration parameter to point to either a different directory or a device.

Primary log archive method (logarchmeth1)
This parameter specifies the media type of the primary destination for archived logs. See section “Log management” for details about the options available.

Secondary log archive method (logarchmeth2)
This parameter specifies the media type of the secondary destination for archived logs. If this parameter is specified, log files will be archived using both this method and the method specified by logarchmeth1.

Track modified pages (trackmod)
When this parameter is set to "Yes", the database manager tracks database modifications so that the backup utility can detect which subsets of the database pages must be examined by an incremental backup and potentially included in the backup image. After setting this parameter to "Yes", you must take a full database backup in order to have a baseline against which incremental backups can be taken.

Using DB2 backup and restore

Basic examples for both offline and online backup of the database are described in the following sections. The examples shown are for the AIX operating system, and may need to be modified for other operating systems. These examples also incorporate name of the days in a week abbreviation in the naming of the backup locations.

Offline backup and offline restore procedures for Tivoli Directory Server database using DB2 backup and restore

Backing up the directory database:

1. Determine a secure location to store the files to be used for backup and recovery, such as a backup machine, separate media, etc. In the examples listed, the /safeplace directory is used as a location to store files. The DB2 instance owner must have write permission for the /safeplace directory.
2. Save Tivoli Directory Server configuration and schema files in a secure location. These files need to be updated only if you change the topology, configuration parameters, or schema. In the examples, the Tivoli Directory Server instance and database are named ldapdb2.

   ```
   cp /home/ldapdb2/idsslapd-ldapdb2/etc/* /safeplace/etc
   ```

3. Make sure that ibmslapd is not running.

   ```
   ibmslapd –I ldapdb2 -k
   ```

4. Create a full database offline backup. You must run all DB2 commands as the DB2 instance owner.

   ```
   db2 force applications all
   db2 backup db ldapdb2 to /safeplace/sun-full-ldapdb2
   ```

Restoring the directory database on a different machine:

1. If necessary, install Tivoli Directory Server.
2. Configure a database, using the same information that was specified on the backup machine.
3. Copy or ftp the configuration, schema, and backup image files from the backup machine to /safeplace on this machine.
4. Copy the backed-up configuration and schema files to this machine.

   ```
   cp /safeplace/etc/* /home/ldapdb2/idsslapd-ldapdb2/etc
   ```

5. Restore the directory database.

   ```
   db2 restore db ldapdb2 from /safeplace/sun-full-ldapdb2 replace existing
   ```

Note: In some versions, DB2 supports cross-platform backup and restore operations and mixed version backup and restore operations. From a Tivoli Directory Server perspective, you cannot back up a database on one version of Tivoli Directory Server and then restore that database on another version of Tivoli Directory Server. It is advisable to use the same version of db2 backup and db2 restore for both DB2 operations.

DB2 online backup and offline restore procedures for Tivoli Directory Server

Setting up online backup for the directory database (without change log)

1. Use a secure location to store files to be used for backup and recovery, such as a backup machine, separate media, etc. In the examples listed, the /safeplace directory is used as a location to store files. The DB2 instance owner must have write permission for the /safeplace directory. In the examples, the Tivoli Directory Server instance and database are named ldapdb2.
2. Save Tivoli Directory Server configuration and schema files in the secure location. These files need to be updated only if you change the topology, configuration parameters, or schema.

   ```
   cp /home/ldapdb2/idsslapd-ldapdb2/etc/* /safeplace/etc
   ```

3. Make sure that ibmslapd is not running.

   ```
   ibmslapd –I ldapdb2 -k
   ```

4. For recovery purposes, log files should be kept on a different physical drive than the database. In this example, the /safeplace/db2logs-ldapdb2 directory is used as the secure location. You must run all DB2 commands as the DB2 instance owner.

   ```
   db2 update db config for ldapdb2 using newlogpath /safeplace/db2logs-ldapdb2
   ```

5. Update the directory server database for online backup support with log archiving on.

   ```
   db2 update db config for ldapdb2 using logarchmeth1 logretain
   db2 force applications all
   db2stop
   db2start
   ```

6. After archival logging is set, you must make a full offline backup. Create a full offline backup of database.

   ```
   db2 backup db ldapdb2 to /safeplace/sun-full-ldapdb2
   ```

7. Start the directory server instance.

   ```
   ibmslapd –I ldapdb2
   ```

Creating full online backup for the directory database

1. On a nightly basis (or more frequently if necessary), create full backup and copy log files from the log file path.

   Note: You can use an online backup image for recovery only if you have the logs that span the time during which the backup operation was running.

   ```
   db2 backup db ldapdb2 online to /safeplace/mon-ldapdb2
   ```

2. Verify the log path. DB2 appends the node to the path specified.

   ```
   db2 get db config for ldapdb2 | grep –i "Path to log files"
   ```

   The following is an example of the information returned:

   ```
   Path to log files        = /safeplace/db2logs-ldapdb2/NODE0000/
   ```

Restoring the directory database

Suppose that a disk drive failed on Wednesday morning on the machine being used, since the /safeplace directory is used to back up the files and logs was not affected, it can be used for restore.

If a different machine is being used to restore the database, the /safeplace directory on the backed up machine must be set up on the new machine to a local /safeplace directory. This must include all backup directories being used, as well as the log files in the /safeplace/db2log-ldapdb2/NODE0000 directory.

1. If necessary, install Tivoli Directory Server.
2. Configure a database, using the same information that was specified on the backup machine.
3. Copy or tar the configuration and schema files backed up previously.

   ```
   cp  /safeplace/etc/*  /home/ldapdb2/idsslapd-ldapdb2/etc
   ```

4. Restore the directory database from Tuesday.

   ```
   db2 restore db ldapdb2 from /safeplace/tues-ldapdb2 taken
                                       at <timestamp_of_backup>
   ```

   Note: The <*timestamp_of_backup*> option is only required if there are more than one backup image in the specified directory path.

If you are restoring on a new machine, the following warning message is displayed:

```
SQL2523W  Warning!  Restoring to an existing database that
is different from the database on the backup image, but
have matching names. The target database will be
overwritten by the backup version. The Roll-forward
recovery logs associated with the target database will be deleted.
Do you want to continue ? (y/n) y
DB20000I  The RESTORE DATABASE command completed successfully.
```

5. Set the new database's log path to the same path that was used for the log files. If you are restoring on a new system, you must copy the log files from the old system to the new.

   ```
   db2 update db config for ldapdb2 using
                  newlogpath /safeplace/db2logs-ldapdb2
   ```

6. .Roll forward all logs located in the log directory, which include changes since the Tuesday night backup.

   ```
   db2 rollforward db ldapdb2 to end of logs and stop
   ```

Note: In this case, recovery requires only the last full backup image and the logs spanning the time since the backup was made.

Setting up incremental online backup for both the directory and change log database to be used for recovery

This section and the following sections are based on a backup strategy with a weekly schedule of doing full backups on Sundays, and then using incremental backups during the week.

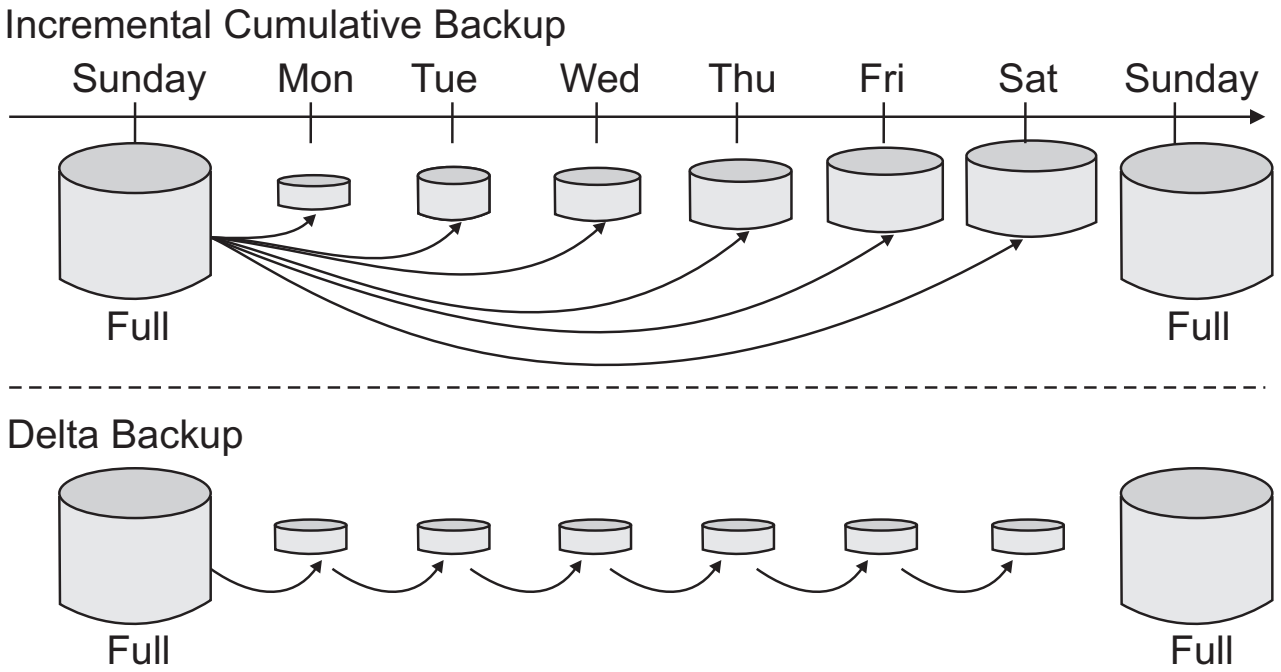


Figure 22. Incremental cumulative and Delta backup

1. Use a secure location to store files to be used for backup and recovery, such as a backup machine, separate media, etc. In the examples listed, the directory /safeplace is used as a location to store files. If the change log is not configured, all commands containing ldapclog can be ignored.

2. Save the Tivoli Directory Server configuration and schema files in a secure location. These files need to be updated only if you change the topology, configuration parameters, or schema. In this example, ldapdb2 is used as the Tivoli Directory Server instance and database instance name.

   ```
   cp /home/ldapdb2/idsslapd-ldapdb2/etc/*  /safeplace/etc
   ```

3. Make sure that ibmslapd is not running.

   ```
   ibmslapd –I ldapdb2 –k
   ```

   Note: In this example, the path of the log files has not been modified from the default locations. Here, the default log path locations are used for both directory and change log databases. For recovery purposes, log files should be kept on a different physical drive than the databases.

4. Update the directory server database and change log database for online backup support with archival logging on, and incremental backup with trackmod on.

   Note: Setting trackmod on for incremental backup support can have an impact on the runtime performance for database update or insert operation.

   ```
   db2 update db cfg for ldapdb2 using logarchmeth1 logretain trackmod on
   db2 update db config for ldapclog using logarchmeth1 logretain trackmod on
   db2 force applications all
   db2stop
   db2start
   ```

Creating full offline backups for both the directory and change log databases

1. Create full database offline backups for both directory and change log databases on Sunday.

   ```
   db2 backup db ldapdb2 to /safeplace/sun-full-ldapdb2
   db2 backup db ldapclog to /safeplace/sun-full-ldapclog
   ```

2. Start the directory server instance.

   ```
   ibmslapd –I ldapdb2
   ```

Creating incremental online backups for both the directory and change log databases

1. On a daily basis or more frequently if determined necessary, create incremental backups.

   Note: You can only use an online backup image for recovery if you have the logs that span the time during which the backup operation was running. Note that the directory and change log database logs are kept in different paths with identical names, for example, S0000000.LOG and S0000001.LOG, so they need to be saved in different directories if the change log is configured.

   ```
   db2 backup db ldapdb2 online incremental to /safeplace/mon-ldapdb2
   ```

2. Verify the path to the log files for the directory database.

   ```
   db2 get db config for ldapdb2 | grep –i "Path to log files"
   ```

   An example of the output displayed:

```
Path to log files = /home/ldapdb2/ldapdb2/NODE0000/SQL00001/SQLOGDIR/

cp /home/ldapdb2/ldapdb2/NODE0000/SQL00001/SQLOGDIR/*
                                /safeplace/db2logs-ldapdb2
db2 backup db ldapclog online incremental to /safeplace/mon-ldapclog
```

3. Verify the path to the log files for the change log database.

   ```
   db2 get db config for ldapclog | grep "Path to log files"
   ```

   An example of the output displayed:

   ```
   Path to log files   = /home/ldapdb2/ldapdb2/NODE0000/SQL00002/SQLOGDIR/

   cp /home/ldapdb2/ldapdb2/NODE0000/SQL00002/SQLOGDIR/*
                                      /safeplace/db2logs-ldapclog
   ```

Restoring the directory and change log databases

Suppose a disk drive failed on Wednesday morning on the machine being used, since the /safeplace directory used to backup the files was not affected, it can be used for restore.

If a different system is being used to restore the database, the /safeplace directories on the backed up system must be set up on the new system to the local /safeplace directory. This must include all backup directories being used, as well as the log files in the /safeplace/db2log-ldapdb2/NODE0000 and the /safeplace/db2log-ldapclog/NODE0000 directories.

1. If necessary, install Tivoli Directory Server. Configure a new database, using the same information that was specified earlier. Copy the configuration and schema files backed up previously.

   ```
   cp  /safeplace/etc/*  /home/ldapdb2/idsslapd-ldapdb2/etc
   ```

2. Make sure that ibmslapd is not running.

   ```
   ibmslapd –I ldapdb2 –k
   ```

3. Restore the directory database. The last backup image to be restored is called the target image. The target image must be restored twice, once at the start of the restore procedure and again at the end. In order to restore Tuesday's incremental backup.

   ```
   db2 restore db ldapdb2 incremental from /safeplace/tues-ldapdb2
   db2 restore db ldapdb2 incremental from /safeplace/sun-full-ldapdb2
   db2 restore db ldapdb2 incremental from /safeplace/tues-ldapdb2
   ```

4. Copy the log files backed up previously to the default log path locations.

   ```
   cp /safeplace/db2logs-ldapdb2/*
                     /home/ldapdb2/ldapdb2/NODE0000/SQL00001/SQLOGDIR

   db2 rollforward db ldapdb2 to end of logs and stop
   ```

5. Restore the change log database.

   ```
   db2 restore db ldapclog incremental from /safeplace/tues-ldapclog
   db2 restore db ldapclog incremental from /safeplace/sun-full-ldapclog
   db2 restore db ldapclog incremental from /safeplace/tues-ldapclog
   ```

6. Copy the log files backed up previously to the default log path locations.

   ```
   cp /safeplace/db2logs-ldapclog/*
                    /home/ldapdb2/ldapdb2/NODE0000/SQL00002/SQLOGDIR

   db2 rollforward db ldapclog to end of logs and stop
   ```

Note: In this case, recovery requires a full backup image and the last incremental backup. Note that the Monday incremental backup is not needed to restore up through Tuesday.

Using incremental delta backups

In the examples using incremental backup, the incremental backup increases in size until the next full backup. This is because the backup contains accumulated changes over time, so there are many more changes saved for Saturday than there were for Monday. DB2 also allows "delta" backups, which save only changes made since the last backup of any kind. These delta backups are much smaller and can be done in lesser time. When restoring, you must have all deltas since the last full or incremental backup.

The commands to perform online delta backups for the ldapdb2 database on a daily basis are listed:

```
db2 backup db ldapdb2 online incremental delta to /safeplace/mon-delta-ldapdb2
db2 backup db ldapdb2 online incremental delta to /safeplace/tues-delta-ldapdb2
db2 backup db ldapdb2 online incremental delta to /safeplace/wed-delta-ldapdb2
db2 backup db ldapdb2 online incremental delta to /safeplace/thurs-delta-ldapdb2
db2 backup db ldapdb2 online incremental delta to /safeplace/fri-delta-ldapdb2
db2 backup db ldapdb2 online incremental delta to /safeplace/sat-delta-ldapdb2
```

When using delta backups, the log files for the database must be kept in a secure location. If you are using the default log paths, you must copy them to the /safeplace/db2logs-ldapdb2 directory or modify the database configuration to save them directly in the /safeplace/db2logs-ldapdb2 directory.

Restoring from incremental delta backups

In the examples, the log files for the database from the backup machine must be available on the machine being used for restoring the delta backups. If you are using the default log paths, you must copy them from the /safeplace/db2logs-ldapdb2/NODE0000 directory on the backup machine to the default log path on the machine being restored, or modify the database configuration newlogpath on the new machine and copy them directly to the /safeplace/db2logs-ldapdb2/NODE000 directory. When restoring from delta backups, you must have ALL deltas since the last full or incremental backup.

The commands to restore online delta backups for the ldapdb2 database are as listed:

```
db2 restore db ldapdb2 incremental from /safeplace/sat-delta-ldapdb2
db2 restore db ldapdb2 incremental from /safeplace/sun-full-ldapdb2
db2 restore db ldapdb2 incremental from /safeplace/mon-delta-ldapdb2
db2 restore db ldapdb2 incremental from /safeplace/tues-delta-ldapdb2
db2 restore db ldapdb2 incremental from /safeplace/wed-delta-ldapdb2
db2 restore db ldapdb2 incremental from /safeplace/thurs-delta-ldapdb2
db2 restore db ldapdb2 incremental from /safeplace/fri-delta-ldapdb2
db2 restore db ldapdb2 incremental from /safeplace/sat-delta-ldapdb2
```

Note: The target image must be restored twice, at the beginning and at the last restore.

Copy the logs and do rollforward:

```
cp /safeplace/db2logs-ldapdb2/*
                /home/ldapdb2/ldapdb2/NODE0000/SQL0001/SQLOGDIR/

db2 rollforward db ldapdb2 to end of logs and stop
```

Pros and cons of different backup and restore strategies

If a database is used for high write activity, an online full backup may be more efficient. Although minimal, the tracking of updates to the database can have an impact on the runtime performance of transactions that update or insert data.

Incremental backup can be used as a way to protect a database that is mostly read-only and has some write activity, which makes it important to be recoverable. An incremental backup image is a copy of all database data that has changed since the most recent, successful, and full backup operation. This is also known as a cumulative backup image. The predecessor of an incremental backup image is always the most recent successful full backup of the same object. With this approach, you must save the last full backup and the last cumulative incremental backup because both will be used for restoring the database.

An incremental delta backup image is a copy of all database data that has changed since the last successful backup, such as full, incremental, or incremental delta. This is also known as a differential or noncumulative backup image. While delta backups are smaller, all deltas since the last full or cumulative incremental backup are required to restore the database.

Managing the archived logs

When using online backup, you need to keep archived logs for as long as they might be required for restoring a database, which depends on your backup methodology and goals. This applies even if you have configured one of the log archival options that "automates" log archiving, you still must have a plan to delete old log files as they become expendable so that your archive space does not get full. One key decision that you must make is whether you want to recover your data up to the most recent backup, or you want to recover data right up to the time of the system failure. In case is a disk fails and you have to restore a database from a backup, you must have the log files that were taken during the backup. After the restore activity, the log files are rolled forward to bring the database to a consistent state that existed after the last backup. If you have saved all the log files generated since the last backup, you can replay the logs right to a time just before the crash. This helps reduce loss of updates to the directory considerably. The next main factor is your backup methodology and schedule. Consider the following examples:

1. If you perform daily full online backups, then you must at least keep the log files that were active during the last backup operation. If you have saved all logs generated since the beginning of the last backup, then you would have all the data necessary to restore the database to the point in time immediately before an event, such as disk or system failure. Any log files archived before the last backup can be deleted to free disk space.
2. If you perform a full online backup once a week and thereafter daily incremental backups in between, then you must at least save the logs that were active during the latest backup, full or incremental. Also, with this approach all the archived logs before the last full backup are no longer needed and can be deleted.
3. If you perform a full online backup once a week and thereafter daily incremental delta backups, then you need to save the logs that were active during the latest backup, full or delta. In order to restore data up to the point in time of the data loss, you must save all the logs since the last backup operation. Any log files archived before the last full backup can be deleted.

Other examples of DB2 backup, restore, and rollforward command options

In cases where you want to restore a database to a specific point in time and not roll forward any changes made after than point in time, the "without rolling forward" option will prevent DB2 from changing the restored database in rollforward pending state.

```
db2 restore db ldapdb2 from /safeplace taken at 20040405154705 without rolling forward
```

To restore a database without prompting for a path that has only one backup database image stored, use the following command:

```
db2 restore db ldapclog from /safeplace/full-backup-ldapclog without
                      rolling forward without prompting
```

The command for offline rollforward database to a point in time:

```
db2 rollforward database ldapdb2 to 2004-04-22-14.54.21.253422 and stop
```

This command rolls forward all logs located in the log folder specified in the database configuration file up to and including the point-in-time as stated in the example. The "and stop" key phrase completes the rollforward recovery process by rolling back incomplete transactions and tuning off the rollforward pending state of the database.

Common problems that may occur during DB2 backup, restore, or rollforward

In the scenarios listed below, the database name ldapdb2 is used. For change log, the change log database ldapclog can be used.

Scenario 1

When you try updating database configuration for online backup parameters while ibmslapd is running:

```
db2 update db cfg for ldapdb2 using logarchmeth1 logretain trackmod on

DB20000I  The UPDATE DATABASE CONFIGURATION command completed successfully.

SQL1363W One or more of the parameters submitted for immediate modification
were not changed dynamically. For these configuration parameters, all
applications must disconnect from this database before the changes become
effective.
```

If you receive the displayed message, you must stop and restart ibmslapd for the changes to take effect. Use the following commands:

```
ibmslapd –I ldapdb2 –k
ibmslapd –I ldapdb2
```

Scenario 2

When you try performing online backup without setting logretain:

```
db2 backup database ldapdb2 online to /safeplace

SQL2413N Online backup is not allowed because either logretain or userexit
for roll-forward is not activated, or a backup pending condition is in
effect for the database.
```

To set the archival logging parameters to enable rollforward recovery for the database ldapdb2 the following DB2 command must be run:

```
db2 update db config for ldapdb2 using logarchmeth1 logretain
```

After archival logging is configured, the user must make a full backup of the database. This state is indicated by the backup_pending flag parameter.

If a full backup has not been made, the following message will be displayed when the user connects to the database:

```
db2 connect to ldapdb2
SQL1116N A connection to or activation of database <ldapdb2>
cannot be made because of a BACKUP PENDING.
```

The database will be in backup pending state until an offline backup is performed. This could cause a server to fail when it connects to the database and will start in configuration mode only.

Scenario 3

Taking a full backup:

```
db2 backup database ldapdb2 to /safeplace
```

If the backup is successful, the following message is displayed:

```
Backup successful.  The timestamp for this backup image is : 20040308170601
```

Scenario 4

When you try to restore a database while ibmslapd is running, the following message is displayed:

```
db2 restore db ldapdb2 from /safeplace
SQL1035N The database is currently in use.
```

Scenario 5

If rollforward must be done following a restore:

```
db2 connect to ldapdb2
SQL1117N A connection to or activation of database "LDAPDB2" cannot be made
because of ROLL-FORWARD PENDING.  SQLSTATE=57019
```

The database will be in rollforward pending state until a rollforward command is issued. This could cause a server to fail when it connects to the database and will start in configuration mode only.

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 6.3 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 with Embedded WebSphere Application Server Version 7.x

The embedded version of WebSphere Application Server version 7.x has HTTPS set up on port 12101 by default. To use HTTPS, you must change your login Web address to the following:

`https://<hostname>:12101/IDSWebApp/IDSjsp/Login.jsp`

For non-HTTPS connections, use the following Web address:

`http://<hostname>:12100/IDSWebApp/IDSjsp/Login.jsp`

Additionally, if you want to change the SSL certificate of application server, 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 *<WAS_HOME>/profiles/TDSWebAdminProfile/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 adding or modifying the following entries (highlighted in bold) in the *<WAS_HOME>/profiles/TDSWebAdminProfile/config/cells/DefaultNode/security.xml* file to use your new file names, passwords, and file formats.

```
<keyStores xmi:id="KeyStore_DefaultNode_10"
 name="DummyServerKeyFile"
 password="{xor}CDo9Hgw="
 provider="IBMJCE"
 location="${WAS_HOME}/profiles/TDSWebAdminProfile/etc/DummyServerKeyFile.jks"
 type="JKS"
 fileBased="true"
 hostList=""
 managementScope="ManagementScope_DefaultNode_1"/>
<keyStores xmi:id="KeyStore_DefaultNode_11"
 name="DummyServerTrustFile"
 password="{xor}CDo9Hgw="
 provider="IBMJCE"
 location="${WAS_HOME}/profiles/TDSWebAdminProfile/etc/DummyServerTrustFile.jks"
 type="JKS"
 fileBased="true"
 hostList=""
 managementScope="ManagementScope_DefaultNode_1"/>
```

© Copyright IBM Corp. 2002, 2010

Creating secure connections between IBM Tivoli Directory Server and the IBM Tivoli Directory Server Web Administration Tool

Create a key pair and certificate request for self-signing key store file (.jks) and a key database file (.kdb).

Notes:

1. See the appendix "Setting up GSKit to support CMS key databases" in *IBM Tivoli Directory Server Version 6.3 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 613.

The only requirements are that you create the key store file and 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 153.

1. Do the following to create a key database (.kdb) file on the system that has Tivoli Directory Server installed:
   a. Type `ikeyman` to start the Java utility.
   b. Select **Key Database File**.
   c. Select **New**, or **Open** if the key database already exists.
   d. From the **Key database type** list, select **CMS**.
   e. 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.
   f. When prompted, supply the password for the key database file. Click **OK**.
   g. Go to **Create**->**New Self-Signed Certificate**.
   h. 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.

i. Click **OK**.

2. Do the following to create a self-signing key store file (.jks) on the system on which Web Administration Tool is installed:
   a. Type `ikeyman` to start the Java utility.
   b. Select **Key Database File**.
   c. Select **New**, or **Open** if the key database already exists.
   d. From the **Key database type** list, select **JKS**.
   e. Specify a key store file name and location. Click **OK**.
   f. When prompted, supply the password for the key store file. Click **OK**.
   g. Go to **Create->New Self-Signed Certificate**.
   h. Supply the following:
      - User-assigned label for the key pair. The label identifies the key pair and certificate in the key database file.

        Note: Be sure that you do not use the same label that you used in step 1g.
      - 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.

   i. 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 type, key database (.kdb) file name, and location.

      Note: This is the key database file you created previously.
   d. When prompted, specify the password.
   e. Click **OK**.
   f. Select **Personal Certificates**.

g. Click **Extract Certificate**.
h. Select **Data type**. For this scenario, select **Binary DER data**.

   Note: The Data type can also be Base-64 encoded ASCII data, which creates .arm file.

i. Provide a filename and location.

   Notes:

   1) Remember this filename and location.
   2) Transfer the extracted server certificate from the server system to the client system, if required.

j. Select **Key Database File**.
k. Select **Open**.
l. Select the key database type, key store (.jks) file name, and location.

   Note: This is the key store file you created previously.

m. When prompted, specify the password.
n. Click **OK**.
o. Go to **Signer Certificates**.
p. Click **Add**.
q. Select the **Binary DER data** (.der) file created previously for the key database (.kdb) file.
r. Click **OK**.
s. Enter a label for the 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 database type, key store (.jks) file name, and location.

      Note: This is the key store file you created previously.

   d. When prompted, specify the 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.

      Notes:

      1) Remember this filename and location.
      2) Transfer the extracted client certificate from the client system to the server system, if required.

   j. Select **Key Database File**.
   k. Select **Open**.
   l. Select the key database type, key database (.kdb) file name, and location.

      Note: This is the key database file you created previously.

   m. When prompted, specify the password.
   n. Click **OK**.

o. Go to **Signer Certificates**.
p. Click **Add**.
q. Select the **Binary DER data** (.der) file created previously for the key store (.jks) file.
r. Click **OK**.
s. Enter a label for the certificate.
t. Select the added certificate and click **View/Edit**. Make sure that the **Set the certificate as a trusted root** check box is selected.
u. Click **OK.**

5. Start the directory server instance, if not started already. See "Starting the directory server instance" in *IBM Tivoli Directory Server Version 6.3 Installation and Configuration Guide* .
6. Start the Web application server. See "Starting the Web application server to use the Web Administration Tool" in *IBM Tivoli Directory Server Version 6.3 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 administration login** is displayed.
      2) In the **User ID** 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) Specify a unique name that identifies a registered IBM Tivoli Directory Server instance running on a specified host name or IP address and server port. The server name is displayed in the LDAP Hostname list on the Directory server login panel. If a name is not provided in the Server name field, the hostname:port combination would be displayed for the server instance in the LDAP Hostname list on the Directory server login panel.
      5) Type the hostname or the IP address of the server in the **Hostname** field; for example, myserver.mycity.mycompany.com.
      6) Specify the server port number in the **Port** field.

7) Select the **Admin server supported** check box to enable the Administration port control.
8) Specify the administration server port number in the **Administration port** field.
9) Ensure the **Enable SSL encryption** check box is not checked.
10) 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 IBM Tivoli Directory Server here apply to the directory administration server as well.

   f. Select the **Server and client authentication** radio button.

      Note: You must distribute the server certificate to the client. For server and client authentication, you also must add the certificate of the 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 6.3 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 the panel.
      - Click **OK** to apply your changes and exit the panel.
      - Click **Cancel** to exit this panel without making any changes.
   i. You must stop and restart both IBM Tivoli Directory Server and the administration server 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) In the **User ID** field, type `superadmin`.
   2) 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 ikeyman" on page 152 and "Secure Sockets Layer" on page 141 for information about key databases and SSL.

   Note: The LDAP server and the administration server 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. Specify a unique name that identifies a registered IBM Tivoli Directory Server instance running on a specified host name or IP address and server port. The server name is displayed in the LDAP Hostname list on the Directory server login panel. If a name is not provided in the Server name field, the hostname:port combination would be displayed for the server instance in the LDAP Hostname list on the Directory server login panel.
    e. Type the hostname or the IP address of the server in the **Hostname** field; for example, myserver.mycity.mycompany.com
    f. Specify the server secure port number in the **Port** field.
    g. Select the **Admin server supported** check box to enable the Administration port control.
    h. Specify the admin server 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.

i. Select the **Enable SSL encryption** check box.

j. Click **OK**, and then click **OK** again on the confirmation panel.

k. Click **Logout** in the navigation area.

Note: You need to restart 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 IBM Tivoli Directory Server and the administration server 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 IBM Tivoli Directory Srever C-based client and IBM Tivoli Directory Server

1. Do the following to create a key database (.kdb) file and self-signed certificate on the server using the ikeyman utility:
    a. Type `ikeyman` 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 type, key database file name (for example, *<server_file>*.kdb), and location. 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 `ikeyman` to start the Java utility.
   b. Select **Key Database File**.
   c. Select **New**, or **Open** if the key database already exists.
   d. Specify a ley database type, key database file name (for example, <*client_file*>.kdb), and location. 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:
   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. Transfer the extracted server's self-signed certificate from the server machine to the client machine.
5. Do the following on the client machine:
   a. Open the <*client_file*>.kdb file.
   b. Go to **Signer Certificates**.
   c. Click **Add**.
   d. Click **Browse** to find the server's self-signed certificate that you transferred to the client machine.
   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 617.

h. Select the certificate and click **View/Edit**. Make sure the **Set the certificate as a trusted root** box is selected.

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.

  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.

k. Click **OK**.

l. Click **Extract Certificate**.

m. Provide a filename and location.

  Note: Remember this filename and location.

n. Click **OK**.

6. Transfer the extracted client's self-signed certificate from the client machine to the server machine.

7. Do the following on the server machine:

   a. Open the <*server_file*>.kdb file.

   b. Go to **Signer Certificates**.

   c. Click **Add**.

   d. Click **Browse** to find the client's self-signed certificate that you transferred to the server machine.

   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 618.

   h. Select the certificate and click **View/Edit**. Make sure the **Set the certificate as a trusted root** box is selected.

8. Issue the following command, on the server machine to modify the cn=SSL,cn=Configuration entry in the ibmslapd.conf file:

```
idsldapmodify -p <port> -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 directory server and the administration server for the changes to take effect.
10. Issue the following command, from the client to verify that the server is functioning as an SSL server:

```
idsldapsearch -h <hostname> -p <server_secure_port> -D <admin_dn> -w <admin_pw>
 -K <keyfile> -b "cn=localhost" objectclass=*
```

Note: You do not need to specify the **-P** option here because the keyfile password was attached to a stash file.

Appendix O. High Availability Scenarios

IBM Tivoli Directory Server is widely deployed in high availability (HA) configurations. In a typical HA configuration, a load balancer is configured in front of several peer masters. Load balancing function, also called virtual IP support or layer 4 routing is usually implemented using network switches. Many network switches from Cisco, F5, Nortel, and other switch vendors have this capability.

For HA configurations, a load balancer is configured only for the purpose of a failover. If a primary master goes down, all traffic to that master is redirected to one of the peer masters. Usually, failback to the original peer is not automatic. However, this is appropriate as the failback is desired only when the replication queue to the newly restarted peer becomes empty. The load balancer sends health check messages to the LDAP servers frequently. For most load balancers, the default health check message is very basic such as a TCP SYN packet. If the target server responds with an ACK, then it is regarded as up. However, the SYN packet is not a very accurate measure of availability, because an ACK is returned even if the target server is in a hung state.

In larger configurations, both load balancing and failover may be desired. Typically, load balancing of write traffic is unwise, because it leads to a possibility of an update conflict. So, one common approach is to configure read and write applications to use a virtual IP address in the load balancer which is configured for failover, and to have read-only applications point to a different virtual IP address, which is configured for load balancing. For write access, the load balancers are configured to failover between peer masters. For read access, failover and load balancing may occur between read-only replicas or between a combination of peer masters and read-only replicas.

The licensed version of Tivoli Directory Server also includes the proxy server. The proxy server has the ability to distinguish between LDAP reads and writes, and so it can failover writes and load balance reads. However, it is advisable to have several proxies so that there is no single point of failure. The proxies are typically fronted by one or several load balancers.

Many LDAP applications use persistent sessions. If persistent sessions are used, the failover process may not be fast. While new sessions are redirected to the backup server, the existing sessions may take several minutes to time out, resulting in a loss of service for that period. This problem can be resolved by using Tivoli Directory Server V6.1 or above proxy server, which fails over existing sessions without disruption. Some load balancers, such as the software load balancer included in IBM WebSphere Application Server Network Deployment, can be configured to send a reset (RST) packet to any persistent sessions, so that they can be quickly re-established on the failover server.

There are several other characteristics of an HA configuration. For instance, in an HA configuration scenario, if one system goes down, the remaining systems must be able to bear the load. Also, it is a good idea to build redundancy into the network configuration, so that if one LAN segment or switch goes down, traffic can still flow from LDAP clients to LDAP servers. In an HA configuration, it is advisable to store LDAP data on RAID arrays, so that no server outage is caused by a physical disk failure. It is also advisable to use system monitoring tools to poll the availability of the servers, so that recovery procedures can be initiated if

© Copyright IBM Corp. 2002, 2010

any of the servers go down. Some scenarios may also have HA support to include multiple redundant sites, so that if an entire site is lost, the other one takes over.

Another important characteristic of HA configuration involves the ability to accomplish maintenance without system downtime. IBM Tivoli directory server supports incremental upgrade of a server topology, so that service can be applied to one server at a time without downtime for the directory service. Updates for the server that is down are queued, so that it comes back into full synch when it is restarted. Tivoli Directory Server also supports online backup of an existing server, using either DB2 or RAID facilities. This allows new servers to be added or existing servers to be replaced in the topology without downtime.

Appendix P. Referential integrity plug-in

Tivoli Directory Server provides a plug-in named libdelref which is a pre-operation plug-in that enables referential integrity constraints for LDAP Delete operation. The libraries are available at the location:<TDS_HOME>/lib or lib64, and library name varies for different platforms as libdelref.dll (Windows), libdelref.a (AIX), libdelref.so (Solaris and Linux). Also, a sample configuration file *tdsdelref.conf* is available in the /etc directory of the Tivoli Directory Server install location. When an instance is created, the *tdsdelref.conf* file becomes available in the etc directory of the instance location.

You can enable the plug-in using the attribute *ibm-slapdReferentialIntegrityPlugin* defined in the *imbslapd.conf* file. By default, the value of this attribute is false. To enable the plug-in you must modify the attribute value to true and restart the server.

The following lines in the ibmslapd.conf file define the libdelref plugin:

```
dn: cn=Directory, cn=RDBM Backends, cn=IBM Directory, cn=Schemas, cn=Configuration
ibm-slapdPlugin: preoperation libdelref.so DeleteReferenceInit
file=/home/nuser/idsslapd-nuser/etc/tdsdelref.conf dn=o=sample
ibm-slapdReferentialIntegrityPlugin: FALSE
```

Note: By default the plug-in entry function is "DeleteReferenceInit". However, for debugging purposes the function "DeleteReferenceInitDebug" may be substituted in the <init-function> specification in the *ibmslapd.conf* file to generate more verbose logging in ibmslapd.log.

Here, the ibm-slapdPlugin attribute defines that the plugin is a pre-operation plugin whose library is libdelref.so. The *file* parameter takes the default value as the complete path of the sample *tdsdelref.conf* file in the etc directory and the *dn* parameter takes the default value for the dn under which you want to search for the entries as o=sample.

To enable the plug-in, issue the following command:

```
idsldapmodify –D <bindDN> –w <password>

dn: cn=Directory, cn=RDBM Backends, cn=IBM Directory, cn=Schemas, cn=Configuration
ibm-slapdReferentialIntegrityPlugin: True
```

The plugin is initialized by reading referential integrity constraint information from the file specified by *file* parameter and the dn specified by the *dn* parameter in the *tdsdelref.conf* file.The *tdsdelref.conf* file is included for reference purposes. You can use any file as long as it adheres to the following format:

```
file=<absolutePathToFile>
dn=<searchDN>

OR

oc=<deleteObjectClass:referenceObjectClass:referenceAttribute>
dn=<searchDN>

where:

absolutePathToFile: is the absolute path to a file containing oc and dn parameters
```

© Copyright IBM Corp. 2002, 2010

deleteObjectClass: is the objectclass name of the deleted object for which the referential integrity is to be maintained

referenceObjectClass: is the objectclass name of the reference object which might contain reference to the deleted object

referenceAttribute: is the attribute name in the referenceObjectClass whose value is the reference to the object being deleted

searchDN: is the base DN, where objects need to searched (for references to the object being deleted)

The file may contain multiple attributes and search base DN specifications in any order. Each specification is treated literally, so white space before and after a specification is not allowed, and will lead to undesirable results.

Note: There can be multiple instances of "oc" and "dn", separated by spaces.

Let us consider an example of how referential integrity works for a delete operation. Consider an example where the entry in *tdsdelref.conf* is :

```
oc=inetOrgPerson:inetOrgPerson:manager
```

Let us assume there are two users in the DIT, namely: cn=testmanager and cn=testuser. Also, let us assume that the manager of cn=testuser is cn=testmanager. For instance:

```
dn: cn=testmanager,o=sample
objectclass: inetOrgPerson
sn: manager

dn: cn=testuser,o=sample
objectclass: inetOrgPerson
sn: testuser
manager: cn=testmanager,o=sample
```

Now, if referential integrity plugin is enabled and you delete cn=testmanager, then all the references to cn=testmanager for manager attribute in cn=testuser will also get deleted.

Appendix Q. Guidelines for interoperability between IBM Tivoli Directory Server and z/OS IBM Tivoli Directory Server

This section contains information to consider when setting up a mixed platform environment where an LDAP directory is being replicated between Tivoli Directory Server on Linux, Unix, or Windows platforms and z/OS IBM Tivoli Directory Server. This information also applies to migration of the schema and directory entries between these different platforms.

Note: Replication from a z/OS LDAP Server to a Distributed LDAP Server on a distributed platform depends on the following conditions:

- The data stored or modified on the z/OS server and the operations used to update them are limited to the subset that is supported on both directory servers.
- The schema definitions are equivalent between the two servers.

The Distributed platforms include AIX, Windows, Solaris, Linux, and HP-UX. For optimal performance, it is better to use replication only between Distributed LDAP Server on Distributed platforms.

Schema considerations

1. **Syntax and matching rules:**

   Tivoli Directory Server supports more syntaxes and matching rules than z/OS IBM Tivoli Directory Server.

   Additional syntaxes and matching rules must be removed from the Tivoli Directory Server schema before it can be used in z/OS IBM Tivoli Directory Server. Attributes using these syntaxes or matching rules must either be removed from the schema or changed to use syntaxes and matching rules supported by z/OS IBM Tivoli Directory Server. If the attributes are in use in an entry, either remove the attribute values from the entry if the attribute is being removed from the schema or ensure that the attribute values conform to the changed attribute definition in the schema.

2. **Schema LDIF format:**

   The format of the schema LDIF obtained from Tivoli Directory Server or z/OS IBM Tivoli Directory Server by publishing the schema (using **ldapsearch –L**) might not be acceptable input for a schema modification.

   a. When modifying the Tivoli Directory Server schema, break up the **attributetypes** and **objectclasses** in the schema file into separate schema modifications, each including a single **attributetypes** value or **objectclasses** value. Also, include an **ibmattributetypes** value (if any) in the modification for its associated **attributetypes** value. If the attribute or object class already exists in the schema, make the modification a modify-replace; otherwise, make the modification a modify-add.

      When modifying the z/OS IBM Tivoli Directory Server schema, the entire LDIF can be processed in a single modify-replace operation, whether or not the attributes or object classes already exist in the schema.

      For example, assume that attribute **attr1** and object class **objclass1** already exist in the schema. For z/OS IBM Tivoli Directory Server, the following schema modification replaces those schema elements and adds new attribute **attr2** and object class **objclass2**:

© Copyright IBM Corp. 2002, 2010

```
dn: cn=schema
changetype: modify
replace: attributetypes
attributetypes: (
  1.3.18.0.2.4.11111
  NAME 'attr1'
  DESC 'Description for attribute attr1'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
  USAGE userApplications
  )
IBMAttributetypes: (
  1.3.18.0.2.4.11111
  ACCESS-CLASS normal
  )
attributetypes: (
  1.3.18.0.2.4.22222
  NAME 'attr2'
  DESC 'Description for attribute attr2'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
  USAGE userApplications
  )
IBMAttributetypes: (
  1.3.18.0.2.4.22222
  ACCESS-CLASS normal
  )
-
replace: objectclasses
objectclasses: (
  1.3.18.0.2.6.33333
  NAME 'objclass1'
  DESC 'Description for object class objclass1'
  SUP top
  STRUCTURAL
  MUST ( cn )
  MAY ( attr1 )
  )
objectclasses: (
  1.3.18.0.2.6.44444
  NAME 'objclass2'
  DESC 'Description for object class objclass2'
  SUP top
  STRUCTURAL
  MUST ( cn )
  MAY ( attr1 $ attr2 )
  )
```

For Tivoli Directory Server, this schema modification has to be reformatted into separate schema modifications and modify-add used instead of modify-replace for the new schema elements, as follows:

```
dn: cn=schema
changetype: modify
replace: attributetypes
attributetypes: (
  1.3.18.0.2.4.11111
  NAME 'attr1'
  DESC 'Description for attribute attr1'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
  USAGE userApplications
  )
IBMAttributetypes: (
  1.3.18.0.2.4.11111
  ACCESS-CLASS normal
  )

dn: cn=schema
changetype: modify
add: attributetypes
```

```
attributetypes: (
  1.3.18.0.2.4.22222
  NAME 'attr2'
  DESC 'Description for attribute attr2'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
  USAGE userApplications
  )
IBMAttributetypes: (
  1.3.18.0.2.4.22222
  ACCESS-CLASS normal
  )

dn: cn=schema
changetype: modify
replace: objectclasses
objectclasses: (
  1.3.18.0.2.6.33333
  NAME 'objclass1'
  DESC 'Description for object class objclass1'
  SUP top
  STRUCTURAL
  MUST ( cn )
  MAY ( attr1 )
  )

dn: cn=schema
changetype: modify
add: objectclasses
objectclasses: (
  1.3.18.0.2.6.44444
  NAME 'objclass2'
  DESC 'Description for object class objclass2'
  SUP top
  STRUCTURAL
  MUST ( cn )
  MAY ( attr1 $ attr2 )
  )
```

b. Ensure that the object classes do not precede the attributes that they reference.

Import or export of directory entries

1. **Exporting data from Tivoli Directory Server to z/OS IBM Tivoli Directory Server:**
   a. Tivoli Directory Server includes certain suffixes, such as `cn=configuration`, `cn=ibmPolicies`, and `cn=localhost`, that contain special entries used to manage LDAP configuration, policies, and replication. z/OS IBM Tivoli Directory Server only supports some of these special entries. You must remove the other special entries from the LDIF or use **db2ldif –s** <**subtreeDN**> **-x** to avoid unloading these suffixes. The `cn=configuration` suffix contains entries that are used to configure advanced replication support. When the server is first started, the following advanced replication configuration entries under the `cn=configuration` suffix are automatically created:
      - cn=configuration
      - cn=Replication,cn=configuration
      - cn=Log Management,cn=Configuration
      - cn=Replication,cn=Log Management,cn=Configuration

See the "*Enabling advanced replication*" section in *Chapter 24: Advanced replication* of the *IBM Tivoli Directory Server Administration and Use for z/OS* guide for more information about the special entries in z/OS IBM Tivoli Directory Server.

b. User passwords must be in clear text, SHA, or CRYPT. Other forms are not compatible with z/OS IBM Tivoli Directory Server. If using CRYPT, make sure to specify **pwCryptCompat off** in the z/OS IBM Tivoli Directory Server's configuration file.

c. z/OS IBM Tivoli Directory Server does not support the use of filtered ACLs in **aclEntry** attribute values (**ibm-filterAclEntry** attribute). You must remove these before importing to z/OS IBM Tivoli Directory Server.

2. **Exporting data from z/OS IBM Tivoli Directory Server to Tivoli Directory Server:**

   a. For Tivoli Directory Server, **aclEntry** and **entryOwner** attribute values must begin with the following format:

   ```
   "access-id:|group:|role:"
   ```

   This is not required for z/OS IBM Tivoli Directory Server, therefore, it might need to be added to these attribute values before importing to Tivoli Directory Server. Always specify these on z/OS IBM Tivoli Directory Server to avoid this issue.

   b. User passwords must be in clear text, SHA, or CRYPT. Other forms are not compatible with Tivoli Directory Server. If using CRYPT, make sure to specify **pwCryptCompat off** in the z/OS IBM Tivoli Directory Server's configuration file. Use **ds2ldif –t** to unload passwords in the tagged format used by Tivoli Directory Server.

Functional considerations

1. For Tivoli Directory Server, deleting a person entry results in removing the DN of the entry from groups and ACLs. This is not done in z/OS IBM Tivoli Directory Server. Instead, applications need to do this themselves.
2. Similarly, Tivoli Directory Server supports a control that allows deletion of all entries in a subtree. z/OS IBM Tivoli Directory Server does not support this control, therefore, applications need to do this themselves.
3. There are some capabilities that only Tivoli Directory Server or only z/OS IBM Tivoli Directory Server supports. Restrict usage to capabilities supported on both platforms to facilitate replication of operations and migration of entries.

Appendix R. Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

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

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

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

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement might not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

© Copyright IBM Corp. 2002, 2010

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

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

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to

IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

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

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Adobe, Acrobat, Portable Document Format (PDF), and PostScript® are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

Cell Broadband Engine™ and Cell/B.E. are trademarks of Sony Computer Entertainment, Inc., in the United States, other countries, or both and is used under license therefrom.

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

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

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

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the U.S., other countries, or both.

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

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

Other company, product, and service names may be trademarks or service marks of others.

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 multivalue 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.

bulkload
A command line utility that is used for bulk-loading large amounts of data in LDIF format.

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

© Copyright IBM Corp. 2002, 2010

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.

DMS (Database Managed Space)
A tablespace where the database manager controls the storage space.

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.

iKeyman

A tool supplied with the Gateway for maintaining digital certificates for SSLight and JSSE.

The ikeyman tool is a user-friendly GUI tool for managing key files. This tool allows creating of public-private key pairs and certificate requests, receiving certificate requests into a key database, and managing keys in a key database.

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 50.

ldapadd
The LDAP modify-entry and LDAP add-entry tool ldapmodify is a shell-accessible interface to the ldap_modify and ldap_add library calls. **ldapadd** is implemented as a renamed version of **ldapmodify**. When invoked as ldapadd the **-a** (add new entry) flag is turned on automatically.

ldapdelete
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.

ldapmodify
The LDAP modify-entry and LDAP add-entry tools ldapmodify is a shell-accessible interface to the ldap_modify and ldap_add library calls. **ldapadd** is implemented as a renamed version of **ldapmodify**. When invoked as ldapadd the **-a** (add new entry) flag is turned on automatically.

ldapmodrdn
LDAP modify-entry RDN tool ldapmodrdn is a shell-accessible interface to the ldap_modrdn library call. **ldapmodrdn** 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.

ldapsearch
The LDAP search tool ldapsearch is a shell-accessible interface to the ldap_search library call. **ldapsearch** 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.

LDAP Data Interchange Format (LDIF)
A format used by the LDAP import-export tools as well as ldapmodify, ldapadd, and ldapsearch command-line utilities to represent LDAP entries or changes to entries in a standard portable text form. See RFC 2849.

ldif2db
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. **ldif2db** can be used to add entries to an empty directory database or to a database that already contains entries.

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=sample, 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 (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 might 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
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.

SMS (System Managed Space)
A tablespace where the operating system's file system manager allocates and manages the space where the table is stored.

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
```

VLV (Virtual List View)

A GUI technique that may be employed where ordered lists containing a large number of entries need to be displayed. VLV provides a scrollable view of large sorted data set through a window containing a small number of visible entries.

Index

Numerics

24/7 338

A

access control lists 419
access controls
  dynamic schema 65
access evaluation
  combinatory rule 429
  specificity rule 429
access permissions
  LDAP operations 426
access rights 425
accessibility xi
ACI mechanisms
  OIDs 501
ACL
  propagation of 427
ACL cache size 112
ACLs 419
  filter-based 420
  filtered 433
  non-filtered 432
  syntax 422
adding an auxiliary object class 412
adding servers 244, 292
administration
  name 81
  password 81
administration server 19
  audit logs 364
    disabling 367
  error logs 362
  SSL security 610
  Starting an instance of the directory administration server 20
  Stopping an instance of the directory administration server 20
administration server audit logs 364
  disabling 367
administration server error logs 362
administrative group 201
  adding members 207
  modifying members 208
  removing members 209
administrator
  administrator group 201
  realms 471
agreements
  replication 229
application servers
  apache tomcat 25
  embedded version of IBM WebSphere Application Server - Express 25
ASCII characters
  33 to 126 515
  allowable in encryption seed string 515
associating
  servers with referrals 212
attribute
  cache 134
  MAY 77
  MUST 77
  syntax 60
attribute cache 134
  adding attributes 133
  removing attributes 133
attribute types
  group 448
  schema file 37
attributes 46
  adding 52
  binary 403
  copying 55
  deleting 57
  editing 53
  multiple values 403
  unique 61
  viewing 51
audit
  error logs 368
    disabling 375
Audit error logs 368
  disabling 375
authentication
  client 146
  server 140
  server and client 140
auxiliary object class
  adding 412
  deleting 413

B

backup and restore 591
binary attributes 403
books
  see publications ix, x
browsing the directory tree 401
bulkload
  error logs 378

C

certificate authority 152
  distinguished names 158
certificate requests 156
certificates 152
changing ports 110
checking
  entries 77
client
  SSL security 616
client authentication 146
common schema 39
complex topology with peer-to-peer
  creating 256
configuration only mode 23
  how to start 23
  requirements 23
  using Web Administration to start 23
  verifying that the server is running in configuration only mode 24
configuration tools
  error logs 380
configuration tools log 380
connections 104
  preventing denial of service 106
  properties 106
console 27
  adding servers to 33
  changing login 33
  changing password 33
  changing properties 34
  logging off of 28
  logging on to 27
  managing 33
  modifying servers 34
  removing servers from 34
controls
  OIDs 505
conventions
  typeface xii
copying an entry 410

D

data interchange format 509
database
  recovering 281
    catastrophic failure 284
    single-server failure 283
database connections
  number of 112
DB2
  error logs 381
DB2 error logs 381
default log paths 359
default log settings
  modifying 361
defining a directory 3
  Directory clients and servers 3
  Directory security 4
deleting
  keys 155
deleting an auxiliary object class 413
deleting an entry 408
DIGEST-MD5
  configuring 191
directories
  distributed 313
directory management 401
  attributes 403
  browsing the directory tree 401
  copying an entry 410
  deleting an entry 408
  directory entries 401
  editing entry ACLs 412

© Copyright IBM Corp. 2002, 2010

directory management *(continued)*
entries 402
modifying an entry 409
directory names, notation xiii
directory server
error logs 384
directory server backup and restore 349
directory server error logs 384
disallowed changes
schema 65
attributes 66
matching rules 76
object classes 66
syntaxes 76
distinguished name 13
pseudo 424
distributed directories 313
back-end servers 322
backup replication 338
server groups 339
creating 322, 324
distributed directory setup tool 319
DN Partition plug-in 318
fail over & load balancing 334
global policies topology
creating 342
LDIF file
creating 340
monitor search 344
partition entries 321
partitioned data
loading 343
partitioning the data 343
proxy 313, 317, 318, 322, 324, 338, 339
proxy server
health check status interval configuration 337
health check thread 336
proxy servers
creating 343
RDN hash 317
replication topology
creating 341
splitting data 317
starting replication 347
synchronizing information 320
transactions in a proxy 347
DN 13
pseudo 424
DN escape characters 14
dynamic
changes
schema 64
dynamic group
editing a memberURL 456
dynamic group entry
creating 450
dynamic groups 442
dynamic schema
access controls 65
changes 64
matching rules 48
replication 65
Dynamically-changed attributes 587

E

editing entry ACLs 412
education
see Tivoli technical training xi
encryption
levels of 164
one-way encrypting
crypt 168
SHA-1 168
SHA-2 168
SSL 163
two-way encrypting
AES128 168
AES192 168
AES256 168
enforcing minimum ulimits 112
Enhanced DN processing 15
entries 449, 450
adding 402
adding an auxiliary objectclass 412
deleting an auxiliary object class 413
entry checking
against schema 77
environment variables, notation xiii
error codes 485
error handling
replication 228
error numbers 485
errors
ldap 485
escaping rules 14
event notification
disabling 123
enabling 123
example
LDIF 509
Version 1 510
exporting
keys 157
extended operations
OIDs 501

F

filtered ACLs 420, 433
sample LDIF file 579

G

gateway topology
creating 265
generalized time 78
global administration group 6
global security
gskcapicmd 148
ikeyman 152
group
attribute types 448
membership 454
object classes 448
group task
verifying 452
groups 441
creating 475
dynamic 442
hybrid 443
groups *(continued)*
management of 481
members of 453
membership 444
nested 443
proxy authorization 463
copying 466
creating 463
modifying 465
removing 466
search limit 459
static 441
Guidelines for interoperability
interoperability
TDS and z/OS IBM TDS 625

H

hybrid groups 443

I

IANA character sets 511
IBM Directory schema
managing 37
IBM Tivoli Directory Integrator 519
IBM Tivoli Directory server
SSL security 610
IBM Tivoli Directory Server 5
IBMAttributeTypes 47
ibmslapd options 24
ibmslapd.conf 127
IBMsubschema 64
identity mapping
Kerberos 189
idsbulkload 378
error logs 378
idsexop 109
idsldapmodify 109
idsslapd options 24
idsslapd.conf 127
importing
keys 158
inheritance
object class 40
iPlanet
compatibility 78
grammar 78
ipv4 517
Ipv6 517

K

Kerberos 187
key
certificate request for existing key 160
changing the database password 155
defaults 156
deleting 155
exporting 157
importing 158
self-signing 156
showing information about 155
trusted root 158
trusted root removal 159

key database
setting 161
key pairs 152
keyring file
migration 160
keys
private 152
public 152

L

language support 511
language tags 405
attributes cannot have associated language tags 406
attributes containing language tags searching 407
attributes language tag values 407
disabling 110
enabling 110
language tag descriptor removing 408
LDIF 509
load balancing 338
log paths
default 359
log settings
default
modifying 361
Logging in to the console 26
console administrator 27
logs
administration server audit 359
administration server error 359
audit 359
administration server 364, 367
bulkload 359
configuration tools 359
DB2 359
default settings 359
disabling 367, 375
errors
administration server 362
audit 368, 375
bulkload 378
configuration tools 380
DB2 381
directory server 384
idsbulkload 378
lost and found 382
viewing 386
idslogmgmt 360
log management tool 360
lost and found 359
server error 359
lost and found
error logs 382
lost and found log 382

M

managing
replication 307
manuals
see publications ix, x
master-replica topology
creating 234
master/replica
unconfiguring 263
matching rules 48
members
managing 453
memberships 454
messages
error 485
migration
keyring file 160
modifying an entry 409
monitor
service status 90
multi-threaded
replication 284

N

namespace 213
nested group entry
creating 451
nested groups 443
non-filtered ACLs 432
sample LDIF file 579
notation
environment variables xiii
path names xiii
typeface xiii
notification
event 123

O

object class
auxiliary 412
IBMAttributeTypes 47
IBMsubschema 64
object classes 39
adding 42
copying 44
deleting 46
editing 43
group 448
viewing 41
object identifier 39
OID 39
OIDs
ACI mechanisms 501
controls 505
extended operations 501
root DSE 491
supported and enabled capabilities 493
online publications
accessing x
operational attributes
password policy 533
ordering publications xi

P

password
administrator 81
console administrator 33
password *(continued)*
security 170
password policy
add/update for an entry 537
operational attributes 533, 536
overriding 534
queries 534
replicating 536
unlocking accounts 534
passwords
administration 81
path names, notation xiii
peer-to-peer
replication 256
performance 111
propagation
ACL 427
proxy authorization
groups 463
proxy server
backing up 338
failover 338
pseudo DNs 424
publications ix
accessing online x
ordering xi

Q

queries
schema 64
queues
replication 305

R

realms 471
adding 475
adding user 475
administrator 471
creating 471
management of 476
template 475
recovery
database 281
ref attribute 212
referral
object class 212
ref attribute 212
referrals 211
default
creating 215
distributing the namespace 213
entries 212
modifying 217
removing 218
server association 212
replica
creating servers 219
replicating
operational attributes 536
password policy 536
replicating servers 244, 292
replication
adding a subtree 286
adding credentials 288

replication *(continued)*
command line tasks 307
configuration information 307
creating gateway servers 310
monitoring status 308
supplier DN and password for a subtree 307
complex topology with peer-to-peer 256
creating a master-replica topology 234
credentials 288
demoting a master 297
dynamic schema 65
editing a subtree 286
editing an agreement 298
error handling 228
error table 285
managing credential ACLs 291
managing gateway servers 298
managing topologies 291
master server 236
master-forwarder-replica 250
modifying credentials 290
modifying properties 301
moving or promoting a server 296
multi-threaded 284
multiple password policy attributes 536
of subtrees 236
overview 222
cascading replication 223
gateway replication 224
peer-to-peer replication 223
replication conflict resolution 225
simple replication 222
partial replication 275
queues 305
quiescing a subtree 287
recovery procedures 281
removing a server 296
removing a subtree 287
removing credentials 291
replicas 237, 294
replication schedule 299
schedules 303
schema and password policy updates 234
server errors 300
server information 299
server roles 221
setting up a gateway topology 265
simple topology with peer replication 244
subtrees 286
supplier information 240, 301
terminology 219
unconfiguring a master/replica 263
viewing topologies 291
replication conflict resolution 225
replication method
multi-threaded 284
required attribute definitions 539
required permissions 426
roles 456
root DSE
attributes 491
rules
indexing 50
attributes 50

S

sample LDIF file
filtered ACLs 579
non-filtered ACLs 579
schedules
daily 304
weekly 303
schema
attribute types 37
attributes 46
adding 52
copying 55
deleting 57
editing 53
encrypted attributes 57
viewing 51
changes
disallowed 65
checking 76
common 39
support 39
dynamic
changes 64
file
attribute types 37
IBM Tivoli Directory Server 585
object classes 39
adding 42
attributes 41
copying 44
defining 40
deleting 46
editing 43
viewing 41
queries 64
subschema entries 63
search
paged results 119
paging 115
settings 115
size limits 115
sorted 115
time limits 115
search filter elements
number of 112
search limit group
copying 462
creating 459
modifying 461
removing 462
search limits
groups 459
searches
advanced 414
entries 414
extended controls 118
manual 416
paging and sorting 118
persistent search 122
simple 414
size limit 459
time limit 459
secure sockets layer 139
security 148, 152
Kerberos 187
password policy 170
self-signed server certificate 145
setting the key database 161
SSL 139, 141
TLS 141
self-signing keys 156
server
SSL security 616
starting 82
stopping 82
server and client authentication 140
server authentication 140
server certificates 143
server performance
settings 111
server properties
setting 109
server replication
gateway 244, 292
masters 244, 292
peer 244, 292
server startup
configuration only mode 23
server status 83
directory cached attributes 100
directory cached candidates 100
general server status 84
operation counts 85
work queue 87
worker status 87
setting searches 115
Simple Network Management Protocol 519
simple topology with peer replication
creating 244
SNMP 519
sorted search 118
SSL 139
SSL scenarios
administration server 610
client and server 616
IBM Tivoli Directory server 610
SSL security 609, 610, 616
Starting an instance of the directory administration server 20
starting the server 82
configuration only mode 23
static group entry
creating 449
static groups 441
status
connections 104
server 83
Stopping an instance of the directory administration server 20
stopping the server 82
subclassing 40
subschema entries 63
subtree replication considerations 440
suffixes 127
adding 127
removing 128
supplier information 240, 301

supported and enabled capabilities
OIDs 493
synchronizing
instances 577
two-way cryptography 577
syntax
ACL 422
attribute 60
Backus Naur Form 13
distinguished name 13
special characters 14

T

Tables
Filtering 31
Finding 30
Paging 30
reordering 31
Select Action drop-down menu 29
Sorting 30
table icons 28
Web Administration Tool 28
template
adding 475
realms 475
templates
creating 473
management of 477
time
generalized 78
UTC 78
Tivoli software information center x
Tivoli technical training xi
Tivoli user groups xi
TLS 139
topology
replication 219, 221
training, Tivoli technical xi
transaction layer security 139
transaction support
disabling 125
enabling 125
transactions
settings 125
trusted root 158
two-way cryptography
instances 577
synchronizing 577
typeface conventions xii

U

unique attributes 61
creating 61
removing an attribute 63
unlocking accounts
password policy 534
URL formats
ipv4 517
ipv6 517
user groups, Tivoli xi
users
management of 480
UTC time 78
UTF-8 511

V

variables, notation for xiii
viewing
error logs 386
viewing log 386

W

Web address protocols
ipv4 517
ipv6 517
Web admin server 19
Web administration console 27
Web Administration console
logs 359
Web Administration Tool
console 26
managing the console 33
setting up 33
starting 25
worker
server status 95

Printed in USA

SC27-2749-00