IBM Tivoli Directory Server
Administration and Use for z/OS
IBM Tivoli Directory Server
Administration and Use for z/OS
Note

Before using this information and the product it supports, be sure to read the general information under Notices.

Acknowledgements

Some of the material contained in this document is a derivative of LDAP documentation provided with the University of Michigan LDAP reference implementation (Version 3.3). Copyright © 1992-1996, Regents of the University of Michigan, All Rights Reserved.

This product includes software developed by the University of California, Berkeley and its contributors.

This product includes software developed by NEC Systems Laboratory.


This is a major revision of SC23-5191-02. This edition, SC23-5191-03, applies to Version 1 Release 11 of z/OS (5694-A01) and to all subsequent releases and modifications until otherwise indicated in new editions.

IBM welcomes your comments. A form for readers' comments may be provided at the back of this document, or you may address your comments to the following address:

International Business Machines Corporation
MHVRCFS, Mail Station P181
2455 South Road
Poughkeepsie, NY 12601-5400
United States of America

FAX (United States & Canada): 1+845+432-9405
FAX (Other Countries):
  Your International Access Code +1+845+432-9405

IBMLink™ (United States customers only): IBMUSM10(MHVRCFS)
Internet e-mail: mhvrcls@us.ibm.com

If you would like a reply, be sure to include your name, address, telephone number, or FAX number.

Make sure to include the following in your comment or note:
• Title and order number of this document
• Page number or topic related to your comment

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 2009.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
Contents

Figures ............................................ xv
Tables ........................................... xvii
About this document ................................ xix
Who should use this document ....................... xix
How this document is organized ...................... xix
Conventions used in this document .................... xix
Where to find more information ....................... xix
   Softcopy publications ................................ xx
   z/OS online library ................................ xx
   The z/OS Basic Skills Information Center ........ xx
Summary of changes ................................ xxi

Part 1. Administration ................................ 1

Chapter 1. Introducing the LDAP server .................. 3
What is a directory service? ............................ 3
What is LDAP? ....................................... 4
   How is information stored in the directory? .......... 4
   How is the information arranged? ..................... 4
   How is the information referenced? ................... 5
   How is the information accessed? ...................... 6
   How is the information protected from unauthorized access? 6
   How does LDAP work? ................................ 6
What about X.500? .................................... 6
What are the capabilities of the z/OS LDAP server? .... 6
   Participation in multilevel security ................. 11
   RFCs supported by z/OS LDAP ....................... 11
   Superseded RFCs ................................... 12

Chapter 2. Planning and roadmap ......................... 13
Planning directory content ............................ 13
LDAP server roadmap ................................ 13

Chapter 3. Installing and setting up related products .... 15
   Required products ................................. 15
   Installing and setting up WLM (Workload Management) .... 15
   Installing a z/OS UNIX System Services file system for the schema backend 15
   Optional Products ................................. 16
   Installing and setting up DB2 for TDBM and GDBM (DB2-based) .......... 16
   Getting DB2 installed and set up for CLI and ODBC ........... 16
   Installing RACF for SDBM and native authentication .......... 18
   Installing a z/OS UNIX System Services file system for LDBM, GDBM (file-based), and CDBM ... 19
   backends ........................................ 19
   Installing and setting up Policy Director and SAF for z/OS Policy Director support .......... 19
   Installing System SSL .............................. 19
   Installing ICSF for encryption ....................... 19
   Installing Kerberos ................................ 20

Chapter 4. Configuring an LDAP server using the dsconfig utility .... 21
   Overview of the LDAP configuration utility .......... 21
Chapter 5. Configuring an LDAP server without the dsconfig utility

Chapter 6. Setting up the user ID and security for the LDAP server

Chapter 7. Preparing WLM, backends, sysplex, SSL/TLS, and encryption
Chapter 8. Customizing the LDAP server configuration

Creating the ds.conf file
   Locating ds.conf
   Configuration file format
   Specifying a value for filename
   Specifying a value for a distinguished name.
   Configuration file checklist
Configuration file options
   Deprecated options
CDBM backend configuration entries
   cn=configuration
   cn=Replication,cn=configuration
   cn=Log Management,cn=Configuration
   cn=Replication,cn=Log Management,cn=Configuration
   Configuration considerations
   Determining operational mode
      Operating in single-server mode
      Operating in multiple single-server mode
      Operating in multi-server mode
      Operating in PC callable support mode
   Establishing the administrator DN and basic replication replica server DN and passwords
       Example configuration scenarios
          Configuring a TDBM backend with SSL/TLS only and password encryption.
          Configuring SDBM and GDBM (DB2-based) backends
          Configuring SDBM and TDBM backends
          Configuring LDBM with native authentication and GDBM (file-based) backends
       Configuring LDBM and CDBM backends with advanced replication.
       Configuring an EXOP backend

Chapter 9. Running the LDAP server

Setting up the PDSE for the LDAP server DLLs
Setting up and running the LDAP server
   Defining the started task for the LDAP server.
   Running the LDAP server using the sample JCL
   LDAP server messages and debug output
   Running the LDAP server using data sets
Verifying the LDAP server
Finalizing setup of LDAP backends
Environment variables used by the LDAP server
Dynamic debugging
   CTRACE in-memory trace records
      Viewing LDAP server CTRACE output
Displaying performance information and server settings
   Size limitations
Activity logging
LDAP SMF auditing
   Auditing events
   Working with audit records
Monitoring LDAP server resources
   Server backends and plug-ins during startup
   DB2
   Network communications
Chapter 10. Migrating to z/OS IBM TDS

Actions required for migrations from previous releases of z/OS IBM TDS

Native authentication and SDBM operations

Migrating to changed format of SMF 83 audit records.

Actions required for migrations from z/OS ISS LDAP server

Migrating server configuration and started task procedure

Migrating schema from an ISS LDAP server to z/OS IBM TDS

Migrating TDBM from an ISS LDAP server to TDBM on z/OS IBM TDS

Migrating TDBM from an ISS LDAP server to LDBM on z/OS IBM TDS

Basic replication considerations when migrating TDBM from an ISS LDAP server to z/OS IBM TDS

Migrating GDBM (change log) from an ISS LDAP server to DB2-based GDBM on z/OS IBM TDS

Migrating GDBM (change log) from an ISS LDAP server to file-based GDBM on z/OS IBM TDS

Changing DB_VERSION setting for TDBM on z/OS IBM TDS

Schema updates

Migration considerations for applications

Coexistence with the ISS LDAP server

Sharing a TDBM or DB2-based GDBM backend between z/OS IBM TDS and an ISS LDAP server

Fallback procedures after migrating a TDBM backend to z/OS IBM TDS

Fallback from a TDBM backend on z/OS IBM TDS to a TDBM backend on an ISS LDAP server

Fallback from an LDBM backend in z/OS IBM TDS to a TDBM backend in an ISS LDAP server

Fallback procedures after migrating a GDBM backend to z/OS IBM TDS

Fallback from a DB2-based GDBM backend in z/OS IBM TDS to a GDBM backend in an ISS LDAP server

Fallback from a file-based GDBM backend in an z/OS IBM TDS to a GDBM backend in an ISS LDAP server

Migration roadmap

Chapter 11. Running and using the LDAP server utilities

Running the LDAP server utilities in the z/OS shell

Running the LDAP server utilities from JCL

Running the LDAP server utilities in TSO

SSL/TLS information for LDAP utilities

Using RACF key rings

Using PKCS #11 tokens

Using a Java keystore or RACF key ring for ldap diff

Utility programs.

db2pwden utility

ds2ldif utility

ldif2ds utility

ldapdiff utility

ldapexop utility

Chapter 12. Internationalization support

Translated messages

UTF-8 support

Part 2. Use

Chapter 13. Data model
<table>
<thead>
<tr>
<th>Initializing ACLs with TDBM or LDBM</th>
<th>346</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default ACLs with LDBM or TDBM.</td>
<td>346</td>
</tr>
<tr>
<td>Initializing ACLs with GDBM</td>
<td>347</td>
</tr>
<tr>
<td>Initializing ACLs with CDBM</td>
<td>347</td>
</tr>
<tr>
<td>Initializing ACLs with schema entry</td>
<td>347</td>
</tr>
<tr>
<td>Access determination</td>
<td>347</td>
</tr>
<tr>
<td>Search</td>
<td>349</td>
</tr>
<tr>
<td>Filter</td>
<td>349</td>
</tr>
<tr>
<td>Compare</td>
<td>349</td>
</tr>
<tr>
<td>Requested attributes</td>
<td>349</td>
</tr>
<tr>
<td>Propagating ACLs</td>
<td>350</td>
</tr>
<tr>
<td>Example of propagation</td>
<td>350</td>
</tr>
<tr>
<td>Examples of overrides</td>
<td>350</td>
</tr>
<tr>
<td>Other examples</td>
<td>351</td>
</tr>
<tr>
<td>Access control groups</td>
<td>352</td>
</tr>
<tr>
<td>Associating DNs and access groups with a bound user</td>
<td>352</td>
</tr>
<tr>
<td>Deleting a user or a group</td>
<td>353</td>
</tr>
<tr>
<td>Retrieving ACL information from the server</td>
<td>353</td>
</tr>
<tr>
<td>Creating and managing access controls</td>
<td>354</td>
</tr>
<tr>
<td>Creating an ACL</td>
<td>354</td>
</tr>
<tr>
<td>Modifying an ACL</td>
<td>356</td>
</tr>
<tr>
<td>Deleting an ACL</td>
<td>357</td>
</tr>
<tr>
<td>Creating an owner for an entry</td>
<td>358</td>
</tr>
<tr>
<td>Modifying an owner for an entry</td>
<td>359</td>
</tr>
<tr>
<td>Deleting an owner for an entry</td>
<td>360</td>
</tr>
<tr>
<td>Creating a group for use in ACLs and entry owner settings</td>
<td>361</td>
</tr>
</tbody>
</table>

### Chapter 23. Basic replication

| Basic replication in a sysplex       | 364 |
| ibm-entryuuid replication            | 364 |
| Complex modify DN replication        | 364 |
| Basic replication and ldif2ds        | 364 |
| Password encryption and basic replication | 364 |
| Replicating server                  | 365 |
| Replica entries                      | 365 |
| Adding replica entries in TDBM or LDBM | 367 |
| Searching a replica entry            | 368 |
| Displaying basic replication status  | 368 |
| Basic replication maintenance mode   | 368 |
| Replica server                       | 369 |
| Populating a replica                 | 369 |
| Configuring the replica              | 369 |
| LDAP update operations on read-only replicas | 370 |
| Changing a read-only replica to a master | 371 |
| Basic peer to peer replication       | 371 |
| Server configuration                 | 372 |
| Basic replication conflict resolution | 372 |
| Adding a peer replica to an existing server | 372 |
| Upgrading a read-only replica to be a peer replica of the master server | 372 |
| Downgrading a peer server to read-only replica | 373 |
| SSL/TLS and basic replication        | 373 |
| Replica server with SSL/TLS enablement | 373 |
| Replicating server with SSL/TLS enablement | 373 |
| Basic replication error log          | 374 |
| Troubleshooting basic replication    | 375 |
| Recovering from basic replication out-of-sync conditions | 376 |
| CDBM performance considerations | 494 |
| TDBM performance considerations | 494 |
| DB2 tuning | 495 |
| TDBM database tuning | 496 |
| Monitoring performance with cn=monitor | 497 |
| Monitor search examples | 502 |
| Large access groups considerations | 505 |
| LE heap pools considerations | 505 |
| GDBM (Changelog) performance considerations | 506 |
| SDBM performance considerations | 507 |

**Part 3. Messages**

**Chapter 30. LDAP server messages**
- LDAP server and Idif2ds messages (1000) | 511
- Utility messages (2000) | 616
- TDBM messages (3000) | 634
- LDBM messages (6000) | 652

**Appendix A. Initial LDAP server schema** | 721

**Appendix B. SPUFI files**
- The DSTDDBMDB SPUFI file | 743
- The TDBMMGRT SPUFI file | 755

**Appendix C. Supported server controls**
- authenticateOnly | 765
- Do Not Replicate | 765
- IBMLdapProxyControl | 765
- IBMModifyDNRealignDNAttributesControl | 766
- IBMModifyDNTimelimitControl | 767
- IBMSchemaReplaceByValueControl | 767
- manageDsaIT | 768
- No Replication Conflict Resolution | 768
- PersistentSearch | 768
- Refresh Entry | 771
- replicateOperationalAttributes | 771
- Replication Supplier ID Bind | 772
- Server Administration | 773

**Appendix D. Supported extended operations**
- Cascading control replication | 775
- changeLogAddEntry | 777
- Control replication | 779
- Control replication error log | 780
- Control replication queue | 781
- GetDnForUserid | 782
- GetPrivileges | 783
- Quiesce or unquiesce context | 784
- Replication topology | 785
- Start TLS | 787
- unloadRequest | 787

**Appendix E. SMF records**
- SMF Record Type 83, subtype 3 records | 791
<table>
<thead>
<tr>
<th>Appendix</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Appendix F.</td>
<td>Guidelines for interoperability between non-z/OS TDS and z/OS TDS</td>
<td>805</td>
</tr>
<tr>
<td></td>
<td>Schema considerations</td>
<td>805</td>
</tr>
<tr>
<td></td>
<td>Import or export of directory entries</td>
<td>807</td>
</tr>
<tr>
<td></td>
<td>Functional considerations</td>
<td>807</td>
</tr>
<tr>
<td>Appendix G.</td>
<td>Accessibility</td>
<td>809</td>
</tr>
<tr>
<td></td>
<td>Using assistive technologies</td>
<td>809</td>
</tr>
<tr>
<td></td>
<td>Keyboard navigation of the user interface</td>
<td>809</td>
</tr>
<tr>
<td></td>
<td>z/OS information</td>
<td>809</td>
</tr>
<tr>
<td>Notices</td>
<td>Policy for unsupported hardware</td>
<td>811</td>
</tr>
<tr>
<td></td>
<td>Trademarks</td>
<td>812</td>
</tr>
<tr>
<td>Bibliography</td>
<td>IBM z/OS Security Server publications</td>
<td>813</td>
</tr>
<tr>
<td></td>
<td>IBM Tivoli Directory Server for z/OS</td>
<td>813</td>
</tr>
<tr>
<td></td>
<td>IBM C/C++ language publications</td>
<td>813</td>
</tr>
<tr>
<td></td>
<td>IBM DB2 publications</td>
<td>813</td>
</tr>
<tr>
<td></td>
<td>IBM z/OS Cryptographic Service publications</td>
<td>813</td>
</tr>
<tr>
<td></td>
<td>Other IBM publications</td>
<td>813</td>
</tr>
<tr>
<td>Glossary</td>
<td></td>
<td>815</td>
</tr>
<tr>
<td>Index</td>
<td></td>
<td>819</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>1</td>
<td>Directory hierarchy example</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>Sample DSNAOINI file</td>
<td>18</td>
</tr>
<tr>
<td>3</td>
<td>LDAP configuration utility overview</td>
<td>22</td>
</tr>
<tr>
<td>4</td>
<td>Sample portion of ds.profile</td>
<td>24</td>
</tr>
<tr>
<td>5</td>
<td>LDAP configuration utility roles and responsibilities</td>
<td>28</td>
</tr>
<tr>
<td>6</td>
<td>General format of ds.conf</td>
<td>70</td>
</tr>
<tr>
<td>7</td>
<td>Sample Schema Entry</td>
<td>229</td>
</tr>
<tr>
<td>8</td>
<td>Object class hierarchy example</td>
<td>233</td>
</tr>
<tr>
<td>9</td>
<td>Before Modify DN operation</td>
<td>250</td>
</tr>
<tr>
<td>10</td>
<td>After Modify DN operation</td>
<td>251</td>
</tr>
<tr>
<td>11</td>
<td>Before Modify DN operation</td>
<td>251</td>
</tr>
<tr>
<td>12</td>
<td>After Modify DN operation</td>
<td>252</td>
</tr>
<tr>
<td>13</td>
<td>Before Modify DN operation</td>
<td>253</td>
</tr>
<tr>
<td>14</td>
<td>After Modify DN operation</td>
<td>253</td>
</tr>
<tr>
<td>15</td>
<td>Before Modify DN operation</td>
<td>258</td>
</tr>
<tr>
<td>16</td>
<td>After Modify DN operation</td>
<td>259</td>
</tr>
<tr>
<td>17</td>
<td>Suffix rename with no new superior</td>
<td>261</td>
</tr>
<tr>
<td>18</td>
<td>Suffix rename with new superior</td>
<td>262</td>
</tr>
<tr>
<td>19</td>
<td>Overlapping suffix rename A</td>
<td>263</td>
</tr>
<tr>
<td>20</td>
<td>Overlapping suffix rename B</td>
<td>264</td>
</tr>
<tr>
<td>21</td>
<td>Suffix rename to non-suffix entry</td>
<td>265</td>
</tr>
<tr>
<td>22</td>
<td>Rename non-suffix entry to suffix entry</td>
<td>266</td>
</tr>
<tr>
<td>23</td>
<td>RACF namespace hierarchy (Part 1 of 2)</td>
<td>281</td>
</tr>
<tr>
<td>24</td>
<td>RACF namespace hierarchy (Part 2 of 2)</td>
<td>281</td>
</tr>
<tr>
<td>25</td>
<td>Kerberos directory example</td>
<td>308</td>
</tr>
<tr>
<td>26</td>
<td>Native authentication example</td>
<td>318</td>
</tr>
<tr>
<td>27</td>
<td>CRAM-MD5 and DIGEST-MD5 authentication example</td>
<td>325</td>
</tr>
<tr>
<td>28</td>
<td>Group hierarchy and membership for the examples</td>
<td>335</td>
</tr>
<tr>
<td>29</td>
<td>Example of adding propagating ACL to existing entry in directory</td>
<td>354</td>
</tr>
<tr>
<td>30</td>
<td>Example of adding propagating ACL to existing entry in the directory</td>
<td>355</td>
</tr>
<tr>
<td>31</td>
<td>Example of setting up a non-propagating ACL</td>
<td>356</td>
</tr>
<tr>
<td>32</td>
<td>Example of adding an aclEntry attribute value</td>
<td>356</td>
</tr>
<tr>
<td>33</td>
<td>Example of modifying aclPropagate attribute</td>
<td>357</td>
</tr>
<tr>
<td>34</td>
<td>Example of removing a single aclEntry attribute value</td>
<td>357</td>
</tr>
<tr>
<td>35</td>
<td>Example of deleting an ACL from an entry</td>
<td>357</td>
</tr>
<tr>
<td>36</td>
<td>Example of adding a propagating set of entry owners to existing entry in the directory</td>
<td>358</td>
</tr>
<tr>
<td>37</td>
<td>Example of setting up a non-propagating entry owner</td>
<td>359</td>
</tr>
<tr>
<td>38</td>
<td>Example of adding an entryOwner attribute value</td>
<td>359</td>
</tr>
<tr>
<td>39</td>
<td>Example of modifying the ownerPropagate attribute</td>
<td>360</td>
</tr>
<tr>
<td>40</td>
<td>Example of removing a single entryOwner Attribute value</td>
<td>360</td>
</tr>
<tr>
<td>41</td>
<td>Example of deleting an entry owner set from an entry</td>
<td>361</td>
</tr>
<tr>
<td>42</td>
<td>Example of adding a group to access control information</td>
<td>361</td>
</tr>
<tr>
<td>43</td>
<td>Example of adding a group to entry owner information</td>
<td>361</td>
</tr>
<tr>
<td>44</td>
<td>Master-replica replication</td>
<td>380</td>
</tr>
<tr>
<td>45</td>
<td>Cascading replication</td>
<td>381</td>
</tr>
<tr>
<td>46</td>
<td>Peer-to-peer replication</td>
<td>382</td>
</tr>
<tr>
<td>47</td>
<td>Gateway replication</td>
<td>383</td>
</tr>
<tr>
<td>48</td>
<td>Master-replica topology</td>
<td>402</td>
</tr>
<tr>
<td>49</td>
<td>Peer-to-peer topology</td>
<td>406</td>
</tr>
<tr>
<td>50</td>
<td>Master-forwarder-replica topology</td>
<td>410</td>
</tr>
<tr>
<td>51</td>
<td>Gateway topology</td>
<td>415</td>
</tr>
<tr>
<td>52</td>
<td>Alias example</td>
<td>442</td>
</tr>
<tr>
<td>53</td>
<td>Example using ref attribute</td>
<td>456</td>
</tr>
</tbody>
</table>
### Tables

1. LDAP server roadmap ........................................... 13
2. LDAP configuration utility roles and responsibilities .................. 27
3. LDAP server configuration roadmap ................................ 35
4. Configuration variable interactions ................................ 38
5. TDBM value definitions for DSTDBMDB ................................ 47
6. TDBM table space partitioning indexes and values ...................... 48
7. TDBM table space partitioning using EID range .......................... 49
8. Supported ciphers .............................................. 60
9. Configuration file options checklist .................................. 73
10. **cn=configuration** entry attribute descriptions ................... 110
11. **cn=Replication,cn=configuration** entry attribute descriptions ...... 112
12. **cn=Replication,cn=Log Management,cn=Configuration** entry attribute descriptions 113
13. Configuration considerations ...................................... 113
14. Sample checklist (using TDBM, SSL/TLS, and password encryption) .......... 121
15. Sample checklist (using SDBM and GDBM) ................................ 122
16. Sample checklist (using SDBM and TDBM) ................................ 123
17. Sample checklist (using LDBM and GDBM) ................................ 124
18. Sample checklist (using CDBM and LDBM) ................................ 126
19. Sample checklist (using EXOP) ......................................... 127
20. Debug levels .................................................. 131
21. Summary of IBM TDS updates for z/OS V1R11 ......................... 173
22. Summary of IBM TDS server updates for z/OS V1R10 .................. 176
23. db2pwden options .............................................. 185
24. Considerations for using ldif2ds or ldapadd .......................... 200
25. ldapdiff general options ......................................... 205
26. ldapdiff supplier server options .................................. 206
27. ldapdiff consumer server options .................................. 207
28. ldapxop general options ......................................... 212
29. Mapping between Unicode and UTF-8 .................................. 219
30. Syntax and default EQUALITY, ORDERING, and SUBSTR matching rules .......... 230
31. Syntax and acceptable matching rules (EQUALITY, ORDERING, and SUBSTR) ......... 230
32. Character representations .......................................... 234
33. Supported LDAP syntaxes - general use ................................ 235
34. Supported LDAP syntaxes - server use ................................ 236
35. Supported matching rules .......................................... 237
36. The errno values returned by __passwd() ............................ 270
37. Mapping of LDAP attribute names to RACF fixed fields (user) ............... 271
38. Mapping of LDAP attribute names to RACF fixed fields (group) .............. 275
39. Mapping of LDAP attribute names to RACF fixed fields (connection) .......... 275
40. Mapping of LDAP attribute names to RACF fixed fields (resource) ............ 276
41. Mapping of LDAP attribute names to RACF fixed fields (setropts) ............. 278
42. RACF backend behavior .......................................... 283
43. SDBM search filters .............................................. 289
44. Kerberos attributes and object classes ................................ 304
45. Operating modes for native authentication binding ....................... 312
46. The errno values returned by __passwd() when binding .................... 313
47. Operating modes for updating native password or password phrases .......... 314
48. The errno values returned by __passwd() when updating password or password phrase 316
49. Behavior of native authentication in example 1 ................................ 318
50. Behavior of native authentication in example 2 ................................ 319
51. Behavior of CRAM-MD5 and DIGEST-MD5 authentication in example ........... 325
52. ACL and entry owner attributes .................................... 341
53. Permissions which apply to an entire entry ................................ 344

© Copyright IBM Corp. 2009
About this document

This document supports z/OS® (5694-A01) and explains the LDAP server. The LDAP server supports Lightweight Directory Access Protocol (LDAP) and runs as a stand-alone daemon. It is based on a client/server model that provides client access to an LDAP server. The LDAP server provides an easy way to maintain directory information in a central location for storage, updating, retrieval, and exchange.

Who should use this document

This document is intended to assist LDAP administrators. LDAP administrators should be experienced and have previous knowledge of directory services. It is also intended for anyone that will be implementing the directory service.

How this document is organized

This document is divided into the following parts:
- Part 1, “Administration”
- Part 2, “Use”
- Part 3, “Messages”
- Part 4, Appendixes, Appendix A, “Initial LDAP server schema”

Conventions used in this document

This document uses the following typographic conventions:

**Bold** words or characters represent API names, attributes, status codes, environment variables, parameter values, and system elements that you must enter into the system literally, such commands, options, or path names.

*Italic* words or characters represent values for variables that you must supply.

**Example Font**
Examples and information displayed by the system appear in constant width type style.

[ ] Brackets enclose optional items in format and syntax descriptions.

{ } Braces enclose a list from which you must choose an item in format and syntax descriptions.

| A vertical bar separates items in a list of choices.

< > Angle brackets enclose the name of a key on the keyboard.

... Horizontal ellipsis points indicate that you may repeat the preceding item one or more times.

\ A backslash is used as a continuation character when entering commands from the shell that exceed one line (255 characters). If the command exceeds one line, use the backslash character \ as the last nonblank character on the line to be continued, and continue the command on the next line.

Where to find more information

Where necessary, this document references information in other documents. For complete titles and order numbers of the documents for all products that are part of z/OS, refer to [z/OS Information Roadmap](#).

For a list of titles and order numbers of the documents that are useful for z/OS LDAP, see [Bibliography](#).
Softcopy publications
The IBM® Tivoli® Directory Server and LDAP libraries are available on a CD-ROM collection, z/OS Collection. The CD-ROM online library collections include Softcopy Reader, which is a program that enables you to view the softcopy documents.

z/OS online library
The softcopy z/OS publications are also available for web browsing and for viewing or printing PDFs using the following URL:

http://www.ibm.com/systems/z/os/zos/bkserv/e0zlib

You can also provide comments about this document and any other z/OS documentation by visiting that URL. Your feedback is important in helping to provide the most accurate and high-quality information.

The z/OS Basic Skills Information Center
The z/OS Basic Skills Information Center is a Web-based information resource intended to help users learn the basic concepts of z/OS, the operating system that runs most of the IBM mainframe computers in use today. The Information Center is designed to introduce a new generation of Information Technology professionals to basic concepts and help them prepare for a career as a z/OS professional, such as a z/OS system programmer.

Specifically, the z/OS Basic Skills Information Center is intended to achieve the following objectives:
• Provide basic education and information about z/OS without charge
• Shorten the time it takes for people to become productive on the mainframe
• Make it easier for new people to learn z/OS.

To access the z/OS Basic Skills Information Center, open your Web browser to the following Web site, which is available to all users (no login required): http://publib.boulder.ibm.com/infocenter/zos/basics/index.jsp
Summary of changes

Summary of changes
for SC23-5191-03
z/OS Version 1 Release 11

This document contains information previously presented in *IBM Tivoli Directory Server Administration and Use for z/OS SC23-5191*, which supports z/OS Version 1 Release 10.

The following summarizes the changes to that information:

New information

- The following configuration options have been added:
  - `enableResources`
  - `serverCompatLevel`
  - `useAdvancedReplication`
  - `wlmExcept`

  The `dsconfig` utility has been updated to support the new configuration options.

- The following utilities have been added:
  - `ldapdiff`
  - `ldapexop`

- The following reason codes have been added:
  - R000146 - R000149
  - R000207
  - R004177 - R004183
  - R006067 - R006070
  - R001102
  - R010067
  - R010500 - R010519
  - R010750 - R010805

- The following messages have been added:
  - GLD1254E to GLD1279A
  - GLD1866E to GLD1868E
  - GLD1871E to GLD1873E
  - GLD2285E
  - GLD3350E
  - GLD8501E
  - GLD8503W to GLD8504E
  - GLD8510E
  - GLD8516E to GLD8536E
  - GLD8538E to GLD8543W
  - GLD8545E to GLD8547E
  - GLD8551E
  - GLD8553E
  - GLD8556E
  - GLD8559E to GLD8560E
  - GLD8563E to GLD8572I
  - GLD8578W to GLD8584E
  - GLD8586E to GLD8596I
  - GLD8598E
  - GLD8601I to GLD8604W
  - GLD8608I

© Copyright IBM Corp. 2009
The following server controls have been added:
- Do Not Replicate
- No Replication Conflict Resolution
- Refresh Entry
- Server Administration
- Replication Supplier ID Bind

The following extended operations have been added:
- Cascading replication
- Control replication
- Control replication queue
- Replication error log
- Replication topology
- Quiesce or unquiesce subtree

RACF general resource profiles and class options can be managed using SDBM operations. Many new attributes have been added to the LDAP schema in support of this.

Change log entries can be created for changes to RACF general resource profiles.

A new backend, CDBM, has been added. This is a file-based backend which stores configuration information.

Appendix F, “Guidelines for interoperability between non-z/OS TDS and z/OS TDS” has been added.

Chapter 24, “Advanced replication,” on page 377 has been added.

Workload manager (WLM) support has been added.

An ID component has been added to searchStats and searchIPStats attribute values on the cn=operations.cn=monitor entry.

A new operator modify command, WLMEXCEPT, has been added.

Updates to Root DSE information for ibm-supportedCapabilities and ibm-enabledCapabilities have been changed.

Changed information
- The following configuration option has been changed:
  - database
  - db2Terminate
- The following reason codes have been changed:
  - R000205
The following messages have been changed:

- GLD1040E
- GLD1158E
- GLD1159E
- GLD1257E
- GLD2004I is now GLD2004D
- GLD2262A
- GLD2273R is now GLD2273D
- GLD3342I
- GLD3343E
- GLD3344E
- GLD3345E
- GLD3346E

- The `ds2ldif` and `ldif2ds` utilities have been updated.
- The `RESET` operator modify command has been updated.

Deleted information

- The following messages have been deleted:
  - GLD2412A
  - GLD2413A
  - GLD2415A
  - GLD2416A
  - GLD2418A through GLD2423I
  - GLD2430A through GLD2434A

Summary of changes for SC23-5191-02
z/OS Version 1 Release 10

This document contains information previously presented in *IBM Tivoli Directory Server Administration and Use for z/OS SC23-5191*, which supports z/OS Version 1 Release 9.

The following summarizes the changes to that information:

New information

- The following messages have been added:
  - GLD1243I
  - GLD1244E
  - GLD1245E
  - GLD1246E
  - GLD1247E
  - GLD1248W
  - GLD1249E
  - GLD1250E
  - GLD1251E
  - GLD1252E
  - GLD1253W
  - GLD3342E
- GLD3343E
- GLD3344E
- GLD3345E
- GLD3346E
- GLD3347W

- The following reason codes have been added:
  - R000100
  - R000101
  - R000102
  - R000105
  - R001100
  - R001101
  - R002020
  - R002021
  - R004166
  - R004176
  - R006066
  - R007082
  - R007083
  - R008126
  - R008127

- The following configuration options have been added:
  - `db2StartUpRetryInterval`
  - `db2StartUpRetryLimit`
  - `operationsMonitor`
  - `operationsMonitorSize`
  - `plugin`
  - `pwSearchOutput`
  - `sslMapCertificate`

The `dsconfig` utility has been updated to support the new configuration options.

- AES keys can now be stored in ICSF CKDS.
- `racfPassPhraseEnvelope` and `racfHavePassPhraseEnvelope` attributes have been added in support of RACF® password phrase support.
- `RACFFIELD` keyword in `ibmAttributeTypes` definition has been added to support RACF custom fields.
- A new DB2® table (`DIR_EID`) and a new partitioning algorithm have been added to improve TDBM performance when using a partitioned database.
- The recommended parameters to use when running the DB2 `RUNSTATS` utility for a TDBM database have been updated.
- Information about configuring the LDAP server to wait for DB2 to start has been added.
- A password phrase can be specified instead of a password when binding using native authentication or SDBM.
- SDBM operations can be performed after a SASL EXTERNAL bind or a native authentication bind.
- Tagged encrypted password values can be used for the `userPassword` attribute.
- Subentry `cn=operations` has been added to `cn=monitor` to store search data.

**Changed information**
• The following messages have been changed:
  – GLD1037E
  – GLD1040E
  – GLD1102E
  – GLD1103E
  – GLD1106E
  – GLD1115E
  – GLD1116E
  – GLD1171E
  – GLD1241W
  – GLD1245E
  – GLD1246E
  – GLD1247E
  – GLD2401E
  – GLD3305E
  – GLD3339W
  – GLD3340I
  – GLD3341W
  – GLD6036E (now GLD6036W)
• The following reason codes have been changed on a failed SDBM bind:
  – R004109 changed to R000100
  – R004128 changed to R000101
  – R004110 changed to R000102
  – R004112 changed to R000105
• The following reason codes have been changed on a native authentication bind:
  – R004062 changed to R004111
  – R004108 changed to R004176
  – R004118 changed to R004108
• The following configuration options have been changed:
  – db2Terminate
  – pwEncryption
  – secretEncryption

Summary of changes
for SC23-5191-01
z/OS Version 1 Release 9

This document contains information previously presented in IBM Tivoli Directory Server Administration and Use for z/OS SC23-5191, which supports z/OS Version 1 Release 8.

The following summarizes the changes to that information:

New information
• The following messages have been added:
  – GLD2284A
  – GLD3339W
  – GLD3340I
GLD3341W

- The ds2ldif utility has been updated to no longer rewrite or commit the LDBM checkpoint and database files while initializing the LDBM backend to unload.
- TDBM has been updated to use DB2 catalog statistics to improve performance of the DB2 requests generated by TDBM.
- Information has been added to describe the RACF SMF unload utility output for audit records created by the LDAP server.
- The use of PKCS #11 tokens when using SSL is now supported.

You may notice changes in the style and structure of some content in this document—for example, headings that use uppercase for the first letter of initial words only, and procedures that have a different look and format. The changes are ongoing improvements to the consistency and retrievability of information in our documents.

This document contains terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.
Part 1. Administration
Chapter 1. Introducing the LDAP server

The z/OS Lightweight Directory Access Protocol (LDAP) server, part of IBM Tivoli Directory Server for z/OS (IBM TDS), is based on a client/server model that provides client access to an LDAP server. An LDAP directory provides an easy way to maintain directory information in a central location for storage, update, retrieval, and exchange.

The LDAP server provides the following functions:
- Interoperability with any Version 2 or Version 3 LDAP directory client
- Access controls on directory information, using static, dynamic, and nested groups
- Secure Sockets Layer (SSL) communication (SSL V3 and TLS V1)
- Start TLS (Transport Layer Security) activation of secure communication
- Client and server authentication using SSL/TLS
- Password encryption
- Basic replication
- Advanced replication
- Referrals
- Aliases
- Change logging
- LDAP Version 2 and Version 3 protocol support
- Schema publication and update
- Kerberos authentication
- Native authentication
- CRAM-MD5 (Challenge-Response Authentication Method) and DIGEST-MD5 authentication
- Root DSE information
- LDAP access to information stored in RACF®
- Support for sharing directory data in a sysplex
- Plug-in support to extend the LDAP server

This information describes how to install, configure, and run the stand-alone LDAP server and other LDAP programs. It is intended for newcomers and experienced administrators alike. This section provides a basic introduction to directory services, and the directory service provided by the LDAP server in particular.

IBM Tivoli Directory Server Client Programming for z/OS describes the LDAP client application programming interfaces (APIs) you can use to develop LDAP applications.

IBM Tivoli Directory Server Plug-in Reference for z/OS describes the LDAP server application programming interfaces (APIs) you can use to develop a plug-in for the LDAP server.

What is a directory service?

A directory is like a database, but tends to contain more descriptive, attribute-based information. The information in a directory is generally read much more often than it is written. As a consequence, directories do not typically implement the complicated transaction or rollback schemes that relational databases use for doing high-volume complex updates. Directory updates are typically simple all-or-nothing changes, if they are allowed at all. Directories are tuned to give quick-response to high-volume lookup or search operations. They may have the ability to replicate information widely in order to increase availability and reliability, while reducing response time. When directory information is replicated, temporary inconsistencies between the replicas are considered acceptable, as long as they get in sync eventually.

There are many different ways to provide a directory service. Different methods allow different kinds of information to be stored in the directory, place different requirements on how that information can be referenced, queried and updated, how it is protected from unauthorized access, and so on. Some directory services are local, providing service to a restricted context (for example, the finger service on a single
machine). Other services are global, providing service to a much broader context (for example, the entire Internet). Global services are typically distributed, meaning that the data they contain is spread across many machines, all of which cooperate to provide the directory service. Typically a global service defines a uniform namespace which gives the same view of the data no matter where you are in relation to the data itself.

What is LDAP?

The LDAP server’s model for the directory service is based on a global directory model called LDAP, which stands for the Lightweight Directory Access Protocol. LDAP Version 2 (V2) and LDAP Version 3 (V3), both supported in z/OS, are directory service protocols that run over TCP/IP. The details of LDAP V2 are defined in Internet Engineering Task Force (IETF) Request for Comments (RFC) 1777 The Lightweight Directory Access Protocol and the details of LDAP V3 are defined in the set of IETF RFCs 2251 - 2256. RFCs supported by z/OS LDAP shows the entire list of supported RFCs.

This section gives an overview of LDAP from a user’s perspective.

How is information stored in the directory?

The LDAP directory service model is based on entries. An entry is a collection of attributes that has a name, called a distinguished name (DN). The DN is used to refer to the entry unambiguously. Each of the entry’s attributes has a type and one or more values. The types are typically mnemonic strings, like cn for common name, or mail for e-mail address. The values depend on what type of attribute it is. For example, a mail attribute might contain an e-mail address with an attribute value of thj@vnet.ibm.com. A jpegPhoto attribute would contain a photograph in binary JPEG format.

How is the information arranged?

In LDAP, directory entries are arranged in a hierarchical tree-like structure that sometimes reflects political, geographic or organizational boundaries. Entries representing countries appear at the top of the tree. Below them are entries representing states or national organizations. Below them might be entries representing people, organizational units, printers, documents, or just about anything else you can think of. Figure 1 shows an example LDAP directory tree, which should help make things clear.
In addition, LDAP allows you to control which attributes are required and allowed in an entry through the use of a special attribute called **object class**. The values of the **objectClass** attribute determine the attributes that can be specified in the entry.

How is the information referenced?

An entry is referenced by its distinguished name, which is constructed by taking the name of the entry itself (called the relative distinguished name, or RDN) and concatenating the names of its ancestor entries. For example, the entry for Tim Jones in the example above has an RDN of `cn=Tim Jones` and a DN of `cn=Tim Jones, o=IBM, c=US`. The full DN format is described in IETF RFC 2253, *LDAP (V3): UTF-8 String Representation of Distinguished Names*.

The z/OS LDAP server supports different naming formats. While naming based on country, organization, and organizational unit is one method, another method is to name entries based on an organization's registered DNS domain name. Names of this form look like: `cn=Tim Smith,dc=vnet,dc=ibm,dc=com`. These naming formats can be mixed as well, for example: `cn=Tim Brown,ou=Sales,dc=ibm,dc=com`.

---

**Figure 1. Directory hierarchy example**

All entries have attributes (and values)
- **objectclass** is an attribute in all entries
- Attributes grouped into required and allowed
How is the information accessed?

LDAP defines operations for interrogating and updating the directory. Operations are provided for adding/deleting an entry to/from the directory, changing an existing entry, and changing the name of an entry. Most of the time, though, LDAP is used to search for information in the directory. The LDAP search operation allows some portion of the directory to be searched for entries that match some criteria specified by a search filter. Information can be requested from each entry that matches the criteria. The LDAP compare operation allows a value to be tested in an entry without returning that value to the client.

An example of search is, you might want to search the entire directory subtree below IBM for people with the name Tim Jones, retrieving the e-mail address of each entry found. LDAP lets you do this easily. Or you might want to search the entries directly below the c=US entry for organizations with the string Acme in their name, and that have a FAX number. LDAP lets you do this too. The section How does LDAP work? describes in more detail what you can do with LDAP and how it might be useful to you.

How is the information protected from unauthorized access?

LDAP client requests can be performed using an anonymous identity or the LDAP bind operation can be used to supply an authentication identity. The LDAP server can use the identity to perform authorization checking when accessing entries in the directory. An Access Control List (ACL) provides a means to protect information stored in an LDAP directory. An ACL is used to restrict access to different portions of the directory, to specific directory entries, or to information within an entry. Access control can be specified for individual users or for groups. This authentication process can be used by distributed applications which need to implement some form of authentication.

How does LDAP work?

LDAP directory service is based on a client/server model. One or more LDAP servers contain the data making up the LDAP directory tree. An LDAP client application connects to an LDAP server using LDAP APIs and asks it a question. The server responds with the answer, or with a pointer to where the application can get more information (typically, another LDAP server). With a properly constructed namespace, no matter which LDAP server an application connects to, it sees the same view of the directory; a name presented to one LDAP server references the same entry it would at another LDAP server. This is an important feature of a global directory service, which LDAP servers can provide.

What about X.500?

LDAP was originally developed as a front end to X.500, the OSI directory service. X.500 defines the Directory Access Protocol (DAP) for clients to use when contacting directory servers. DAP has been characterized as a heavyweight protocol that runs over a full OSI stack and requires a significant amount of computing resources to run. LDAP runs directly over TCP and provides most of the functionality of DAP at a much lower cost.

An LDAP server is meant to remove much of the burden from the server side just as LDAP itself removed much of the burden from clients. If you are already running an X.500 service and you want to continue to do so, you can probably stop reading this guide, which is all about running LDAP through an LDAP server without running X.500. If you are not running X.500, want to stop running X.500, or have no immediate plans to run X.500, read on.

What are the capabilities of the z/OS LDAP server?

You can use the z/OS LDAP server to provide a directory service of your very own. Your directory can contain just about anything you want to put in it. Some of the z/OS LDAP server's more interesting features and capabilities include:

- **Multiple concurrent database instances** (referred to as backends): The LDAP server can be configured to serve multiple databases at the same time. This means that a single z/OS LDAP server
can respond to requests for many logically different portions of the LDAP tree. A z/OS LDAP server can be configured to provide access to RACF, as well as store application-specific information.

- **Robust general-purpose databases** The LDAP server comes with TDBM and LDBM backends. There are no restrictions on the types of information that these backends can contain. The TDBM backend is based on DB2 and is a highly scalable database implementation. The LDBM backend keeps its entries in memory for quick access and requires a minimum amount of setup. When the LDAP server is not running, LDBM stores its directory information in z/OS UNIX System Services files.

  **Note:** DB2 is required to use TDBM, but is not required for LDBM.

- **Access to RACF data** The LDAP server can be configured to provide read/write access to RACF user, group, connection, and general resource profiles using the LDAP protocol. The LDAP server can also be used to manage RACF options that affect classes. (RACF is a component of the Security Server for z/OS.) If the RACF data is shared across the sysplex, then users, groups, connections, and resource profiles in the sysplex can be managed using LDAP. The LDAP server's access to RACF is managed by an additional configurable backend called SDBM. See Chapter 16, “Accessing RACF information” for more information.

  **Note:** To use SDBM for ONLY authentication (LDAP bind processing), any security manager implementing the SAF service required by the __passwd() function call can be used. A password or a password phrase can be used for authentication. To use SDBM for accessing and updating user, group, connection, and resource profile information, and to set class options, RACF is required.

- **Configuration backend** The LDAP server can be configured with a CDBM backend. The CDBM backend is used to store configuration information. See Setting up for CDBM for more information.

- **Loading and unloading data** The LDAP server can load a large number of entries into a TDBM directory using the ldif2ds utility. See ldif2ds utility for more information. The LDAP server can also unload a large number of entries from a TDBM, LDBM, or CDBM directory using the ds2ldif utility. See ds2ldif utility for more information.

- **Access control** The LDAP server provides a rich and powerful access control facility, allowing you to control access to the information in your database or databases. You can control access to entries based on LDAP authentication information, including users and groups. Group membership can be either static, dynamic, or nested. Access control is configurable down to individual attributes within entries. Also, access controls can be set up to explicitly deny access to information. See Chapter 22, “Using access control” on access control and Chapter 21, “Static, dynamic, and nested groups” for more information about groups.

- **Threads** The LDAP server is threaded for optimal performance. A single multi-threaded z/OS LDAP server process handles all incoming requests, reducing the amount of system overhead required.

- **Multiple concurrent servers** The LDAP server can be configured to permit multiple LDAP servers to serve the TDBM, LDBM, CDBM, or GDBM directory at the same time. The multiple servers may run on the same z/OS image, and they may run on multiple z/OS images in a Parallel Sysplex®. This improves availability and may offer improved performance in certain configurations. See Determining operational mode for more information.

- **Basic replication** The LDAP server can be configured to maintain replica copies of its database. Master/consumer replication is vital in high-volume environments where a single LDAP server just does not provide the necessary availability or reliability. Peer to peer replication is also supported. See Chapter 23, “Basic replication” for more information. This feature is contrasted with multiple concurrent servers.

- **Advanced replication** The LDAP server can be configured to act as a supplier, consumer, cascading, or gateway server in an advanced replication environment. An advanced replication environment allows for only certain subtrees or entries in a TDBM, LDBM, or CDBM backend to be replicated to other servers. See Chapter 24, “Advanced replication” for more information.

- **Referrals** The LDAP server provides the ability to refer clients to additional directory servers. Using referrals you can distribute processing overhead, distribute administration of data along organizational
boundaries, and provide potential for widespread interconnection beyond an organization's own boundaries. See Chapter 27, “Referrals” for more information.

- **Aliases**: An alias entry can be created in the directory to point to another entry in the directory. During search operations, an alias entry can provide a convenient public name for an entry or subtree, hiding the more complex actual name of the entry or subtree. It can also avoid the need to duplicate an entry in multiple subtrees. See Chapter 25, “Alias” for more information.

- **Change Logging**: The LDAP server can be configured to create change log entries in the GDBM backend. Each change log entry contains information about a change to an entry in a TDBM, CDBM, or LDBM backend, to the LDAP server schema, or to a RACF user, group, connection, or resource profile. The GDBM backend can be configured to store its entries in DB2 (similar to TDBM) or in z/OS UNIX System Services files (similar to LDBM). See Chapter 26, “Change logging” for more information.

- **Configuration**: The LDAP server configuration process can be simplified by using the `dsconfig` configuration utility. This utility requires minimal user interaction and allows novice LDAP users to quickly configure an LDAP server. See Chapter 4, “Configuring an LDAP server using the dsconfig utility” for more information.

  If you do not use the `dsconfig` utility, the LDAP server is highly configurable through a single configuration file which allows you to change just about everything you would ever want to change. Configuration options have reasonable defaults, making your job much easier. See Creating the ds.conf file for more information.

- **Secure communications**: The LDAP server can be configured to encrypt data to and from LDAP clients using the z/OS Cryptographic Services System SSL. The LDAP server supports the Start TLS extended operation to switch a non-secure connection to a secure connection. It has a variety of ciphers for encryption to choose from, all of which provide server and optionally client authentication through the use of X.509 certificates. See Setting up for SSL/TLS for more information.

- **Dynamic workload management**: The LDAP server can be configured to participate in dynamic workload management in a Parallel Sysplex by exploiting TCP/IP connection optimization. With multiple concurrent server instances configured in this way, availability is improved, as is resource utilization. In addition, performance improvements may be experienced as sysplex resource utilization is more evenly balanced across z/OS systems in the sysplex. See Determining operational mode for more information.

- **Retrieve Policy Director data**: The z/OS LDAP server, when using the EXOP backend, supports two LDAP extended operations, GetDnForUserid and GetPrivileges, that retrieve Policy Director data from any LDAP server. See Chapter 20, “Using extended operations to access Policy Director data” for more information.

- **Native authentication**: The z/OS LDAP server allows clients to bind to entries in a TDBM, LDBM, or CDBM backend by using the system for verifying the authentication attempt. The client can perform a simple bind supplying an LDAP DN of an entry in a TDBM, LDBM, or CDBM backend along with a security manager-maintained password or password phrase. Password or password phrase authentication is then performed by the security manager. See Chapter 18, “Native authentication” for more information.

  Note: To use native authentication, any security manager implementing the SAF service required by the __passwd() function call can be used.

- **LDAP Version 3 protocol support**: The LDAP server provides support for Version 3 of the LDAP protocol in addition to the LDAP Version 2 protocol. Version 3 includes:
  - All protocol operations
  - Implicit bind
  - Certificate (or Simple Authentication and Security Layer) bind
  - Version 3 referrals
  - Aliases
  - Controls
  - Root DSE support
  - Internationalization (UTF-8) support
  - Modify name supported for all entries including subtree move
  - Schema publication
– Additional syntax support
– Online schema update capability

**Dynamic schema:** The LDAP server allows the schema to be changed dynamically through the LDAP protocol. See Chapter 14, “LDAP directory schema” for more information.

**Internationalization (UTF-8) support:** The LDAP server allows storage, update and retrieval, through LDAP operations, of national language data using LDAP Version 3 protocol. See UTF-8 support for more information.

**SASL external bind and client and server authentication:** The LDAP server allows client applications to use a certificate when communicating with the server using SSL/TLS communications. In order to use a certificate on bind, the server must be configured to perform both client and server authentication. This ensures both entities are who they claim to be. See Setting up for SSL/TLS for more information.

**SASL GSS API Kerberos bind with mutual authentication:** The LDAP server allows clients to bind to the server using Kerberos credentials. Mutual authentication is used to verify both the client and server identities. See Chapter 17, “Kerberos authentication” for more information.

**SASL CRAM-MD5 and DIGEST-MD5 authentication:** The LDAP server allows clients to bind to the server using DIGEST-MD5 [RFC 2831: Using Digest Authentication as a SASL Mechanism] and CRAM-MD5 [RFC 2195: IMAP/POP AUTHorize Extension for Simple Challenge/Response] authentication bind methods. See Chapter 19, “CRAM-MD5 and DIGEST-MD5 authentication” for more information.

**Support for root DSE:** The LDAP server supports search operations, including subtree search, against the root of the directory tree as described in RFC 2251: Lightweight Directory Access Protocol (v3). The so-called Root DSE can be accessed using LDAP V3 search operations. See Chapter 17, “Root DSE” and IBM Tivoli Directory Server Client Programming for z/OS for more information.

**Extended group membership searching:** The LDAP server supports extended group membership searching which allows the LDAP server to find a DN that may be a member of static and nested groups in a backend (TDBM, LDBM, or CDBM) where the DN does not reside. The LDAP server can find the group memberships for the DNs in the other backends that are configured. See the extendedGroupSearching configuration file option on page 83 for more information.

**Supported server controls:** The LDAP server supports the following:

- authenticateOnly
- Do NotReplicate
- IBMldapProxyControl
- IBMModifyDNRealignDNAttributesControl
- IBMModifyDNTimelimitControl
- IBMSchemaReplaceByValueControl
- manageDsasIT
- No Replication Conflict Resolution
- PersistentSearch
- Refresh Entry
- replicateOperationalAttributes
- Replication Supplier ID Bind
- Server Administration

See Appendix C, “Supported server controls” for more information.

**Supported extended operations:** The LDAP server supports the following:

- Cascading control replication
- changeLogAddEntry
- Control replication
- Control replication error log
- Control replication queue
- GetDnForUserid
- GetPrivileges
- Quiesce or unquiesce context
- Replication topology
Start TLS
unloadRequest

See Appendix D, “Supported extended operations” for more information.

- **Attribute encryption**: The LDAP server supports encryption of the values of several critical attributes to prevent unauthorized access to these attribute values in TDBM, LDBM, and CDBM backends. The attributes that can be encrypted are as follows:
  - ibm-replicaKeyPwd
  - ibm-slapdMasterPw
  - replicaCredentials
  - secretKey
  - userPassword

See Configuring for encryption for more information.

- **Multiple socket ports**: The LDAP server can be configured to listen for secure and nonsecure connections from clients on one or more IPv4 or IPv6 interfaces on a system. With the `listen` configuration option on the LDAP server, the hostname or the IPv4 or IPv6 address, along with the port number, can target one or multiple IPv4 or IPv6 interfaces on a system. See the `listen` configuration option on page 86 for more information.

- **Persistent search**: The LDAP server provides an event notification mechanism for applications, directories and meta-directories that need to maintain a cache of directory information or to synchronize directories when changes are made to an LDAP directory. Persistent search will allow these applications to be notified when a change has occurred. See Appendix C, “Supported server controls,” on page 765 for more information.

- **ibm-entryuuid attribute**: The LDAP server generates a unique identifier for any entry that is created or modified and does not already have a unique identifier assigned. The unique identifier is stored in the `ibm-entryuuid` attribute. The `ibm-entryuuid` attribute is replicated to servers that support the `ibm-entryuuid` attribute. See the Configuration file options to configure the `serverEtherAddr` option in the LDAP server configuration file.

- **ibm-entryCheckSum and ibm-entryCheckSumOp attributes**: The LDAP server supports the querying of a checksum of all non-operational attributes with the `ibm-entryCheckSum` operational attribute. The LDAP server also supports the `ibm-entryCheckSumOp` operational attribute, which is a checksum of the following operational attributes: aclEntry, aclPropagate, entryOwner, ownerPropagate, creatorsName, modifiersName, createTimestamp, modifyTimestamp, and `ibm-entryuuid`. The `ibm-entryCheckSum` and `ibm-entryCheckSumOp` operational attributes are used by the `ldapdiff` utility to simplify comparisons of entries between two different servers. See `ldapdiff utility` for more information on the `ldapdiff` utility.

- **ibm-allMembers and ibm-allGroups attributes**: The LDAP server supports the querying of the members of static, dynamic, and nested groups in a TDBM, LDBM, or CDBM backend by using the `ibm-allMembers` operational attribute. The LDAP server also supports the querying of the static, dynamic, and nested groups that a user belongs to with the `ibm-allGroups` operational attribute.

- **Plug-in support**: The LDAP server can be configured to provide access to extensions to the LDAP server. The extensions are supplied by other products or created by you. Plug-in extensions are invoked before, during, or after the LDAP server processes a client request. See the `plugin` configuration option in Chapter 8, “Customizing the LDAP server configuration” for more information about configuring a plug-in extension. See IBM Tivoli Directory Server Plug-in Reference for z/OS for more information about creating a plug-in extension.

- **Workload Management**: The LDAP server can be configured to use transaction names configured in Workload Manager (WLM) to use different performance goals for LDAP operations based upon the client’s IP address or the bound user’s distinguished name (DN). See Workload manager (WLM) for more information on using WLM with the LDAP server.

- **Comparing directories**: The `ldapdiff` utility is provided to compare the directory contents of two different LDAP servers. This utility is useful in determining and optionally synchronizing data between a master and replica server before configuring basic or advanced replication. See `ldapdiff utility` for more information.
Extended operations utility: The ldapexop utility provides a command line interface for the following extended operations: Cascading control replication, Control replication, Control replication error log, Control replication queue, Quiesce or unquiesce context, and Replication topology, and Quiesce or unquiesce subtree. See ldapexop utility for more information.

Participation in multilevel security

Multilevel security is an enhanced security environment that can be configured on a z/OS system from the SECLABEL class and various SECLABEL-related options. In this environment, the security server and trusted resource managers enforce mandatory access control policies in addition to the usual discretionary access control policies. Resource managers that do not support mandatory access control processing can participate in a multilevel secure system, if they are physically managed, to guarantee that all information made available by that resource manager has the same single security label and all users of the resource manager are permitted to that security label. These servers are referred to as single-level security. Each LDAP server can participate in a multilevel security environment by being configured as a single-level server. For more information about configuring a z/OS system for multilevel security and how to configure an LDAP server in that environment, see z/OS Planning for Multilevel Security and the Common Criteria.

RFCs supported by z/OS LDAP

The z/OS LDAP server supports the following Internet Engineering Task Force (IETF) request for comments (RFCs):

- RFC 1738: Uniform Resource Locators (URL)
- RFC 1823: The LDAP Application Program Interface
- RFC 2052: A DNS RR for specifying the location of services (DNS SRV)
- RFC 2104: HMAC: Keyed-Hashing for Message Authentication
- RFC 2195: IMAP/POP AUTHorize Extension for Simple Challenge/Response
- RFC 2222: Simple Authentication and Security Layer (SASL)
- RFC 2247: Using Domains in LDAP/X.500 Distinguished Names
- RFC 2254: The String Representation of LDAP Search Filters
- RFC 2255: The LDAP URL Format
- RFC 2256: A Summary of the X.500 (96) User Schema for use with LDAPv3
- RFC 2279: UTF-8, a transformation format of ISO 10646
- RFC 2373: IP Version 6 Addressing Architecture
- RFC 2714: Schema for Representing CORBA Object References in an LDAP Directory
- RFC 2732: Format for Literal IPv6 Addresses in URLs
- RFC 2743: Generic Security Service Application Program Interface Version 2, Update 1
- RFC 2744: Generic Security Service API Version 2: C-bindings
- RFC 2820: Access Control Requirements for LDAP
- RFC 2829: Authentication Methods for LDAP
- RFC 2831: Using Digest Authentication as a SASL Mechanism
- RFC 2849: The LDAP Data Interchange Format (LDIF)

Note that although the LDAP V3 protocol RFCs are listed as supported, the specific function that z/OS LDAP supports is listed in “LDAP Version 3 protocol support” on page 8.

LDAP RFCs rely on the International Telecommunication Union (ITU-T) standards to define many of the basic constructs:

- X.500 The Directory: Overview of Concepts, Models and Services
- X.501 The Directory: Models
Superseded RFCs

The following obsolete RFCs were implemented by the z/OS LDAP client and server but have been superseded by current RFCs.

- RFC 1778: The String Representation of Standard Attribute Syntaxes
- RFC 1779: A String Representation of Distinguished Names
- RFC 1959: An LDAP URL Format
- RFC 1960: A String Representation of LDAP Search Filters
Chapter 2. Planning and roadmap

This topic:
- Shows you where to find information in this information that will help you plan your directory content.
- Contains a roadmap that points you to information that may be helpful in preparing for your LDAP server configuration.

Planning directory content

Before configuring and populating your database, determine:
- What type of data you are going to store in the directory.
  You should decide on what sort of schema you need to support the type of data you want to keep in your directory. The directory server is shipped with a standard set of attribute type and object class definitions.
  Before you begin adding entries to the directory, you might need to add new attribute type and object class definitions that are customized to your data.
  Schema definition styles are specific to the backend or data store being configured. Once you have determined which backends to configure, refer to Chapter 14, “LDAP directory schema” or Setting up for SDBM for more information.
- How you want to structure your directory data.
  Refer to Chapter 13, “Data model” for more information.
- A set of policies for access permissions.
  Refer to Chapter 22, “Using access control” for more information.

LDAP server roadmap

Table 1 is a roadmap that points you to information that may be helpful in preparing for your LDAP server configuration.

See Chapter 10, “Migrating to z/OS IBM TDS” if you have a previous release of the LDAP server installed on your system.

For complete instructions for installing the LDAP server product, see z/OS Program Directory which comes with the LDAP server tape or cartridge. Be sure to read the license agreement in z/OS Licensed Program Specifications which is also included in the box.

Important

Before you proceed, review the Memo to Users, which describes any late changes to the procedures in this information. A printed copy is included with the LDAP server tape or cartridge.

<table>
<thead>
<tr>
<th>Complete?</th>
<th>Task</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>If you are configuring your LDAP server for z/OS WebSphere™ Application Server (z/OS Component Broker), you must:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>See the WebSphere® documentation for details on LDAP server requirements.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>If you are configuring your LDAP server for z/OS Policy Director, you must:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>See the z/OS Policy Director documentation for details on LDAP server requirements.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 1. LDAP server roadmap (continued)

<table>
<thead>
<tr>
<th>Complete?</th>
<th>Task</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>If you are configuring your LDAP server for z/OS Enterprise Identity Mapping (EIM), you must:</td>
<td>See the z/OS EIM documentation for details on LDAP server requirements.</td>
<td></td>
</tr>
<tr>
<td>If you want to see how a working LDAP server looks, you must:</td>
<td>See the <code>/usr/lpp/ldap/examples/sample_server</code> directory which contains everything necessary to set up a sample LDAP server. See the <code>ds.README</code> file in this directory for complete instructions.</td>
<td></td>
</tr>
<tr>
<td>If you are migrating from a previous release of the LDAP server, you must:</td>
<td>See the migration information.</td>
<td>Chapter 10, “Migrating to z/OS IBM TDS.”</td>
</tr>
</tbody>
</table>
| If this is the first time you are installing the z/OS LDAP server, you must: | Read the following documents that are included in the box with the z/OS LDAP server tape or cartridge:  
  - [z/OS Program Directory](/usr/lpp/ldap/examples/sample_server) which contains the complete install instructions.  
  - [z/OS Licensed Program Specifications](/usr/lpp/ldap/examples/sample_server) which contains the license agreement.  
  - [Memo to Users](/usr/lpp/ldap/examples/sample_server), which describes any late changes to the procedures in this information. | |
| Choose a configuration method from the following options: | The `dsconfig` configuration utility uses a profile file as input to generate jobs to set up the system environment and configuration. Check [Restrictions](/usr/lpp/ldap/examples/sample_server) to decide if this method will work for your installation. | Chapter 4, “Configuring an LDAP server using the `dsconfig` utility.” |
| | If you do not use the `dsconfig` utility, use the instructions for customizing your configuration. | Chapter 8, “Customizing the LDAP server configuration.” |
Chapter 3. Installing and setting up related products

This topic discusses the required and optional products that are necessary to install or set up prior to configuring the z/OS LDAP server product.

Required products

This section describes the products that must either be installed or configured for the LDAP server to work properly.

To run the LDAP server: You must:

<table>
<thead>
<tr>
<th>To run the LDAP server:</th>
<th>You must:</th>
<th>See:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workload Manager (WLM)</td>
<td>Install and configure Workload Manager (WLM).</td>
<td>Installing and setting up WLM (Workload Management)</td>
</tr>
<tr>
<td>Schema backend (based on z/OS UNIX System Services file system)</td>
<td>Ensure that a z/OS UNIX System Services file system exists and has enough space to store the schema backend and that it can be written to by the LDAP server.</td>
<td>Installing a z/OS UNIX System Services file system for the schema backend</td>
</tr>
</tbody>
</table>

Installing and setting up WLM (Workload Management)

The LDAP server supports Workload Manager (WLM) to allow an installation to set performance goals for work within the LDAP server. WLM has reserved LDAP as a subsystem type. The WLM ISPF panels allow the LDAP subsystem type to be associated with a default service or report class. The service and report classes must be defined before being used with the LDAP subsystem type. See z/OS MVS Planning: Workload Management for more information about WLM.

It is strongly recommended that a default service or replace class be configured for the LDAP subsystem type even if WLM is not being used for setting performance goals within the LDAP server. If a default service or replace class is not specified in WLM, all LDAP server operations run under the discretionary profile and receive a low priority.

The following WLM classification parameters are accepted on the LDAP subsystem type:

- **Sysplex Name (PX):** Identifies the sysplex name where the LDAP server(s) reside. This allows an installation to establish the same performance goals for all LDAP servers running within the same sysplex.
- **Subsystem Instance (SI):** Identifies the started task name of the LDAP server. This allows an installation to establish performance goals for an individual LDAP server that is running within the sysplex.
- **Transaction Name (TN):** Identifies a WLM transaction name that has been configured in WLM. These transaction names can be used within any subsystem type in WLM. For the LDAP server, the default transaction name is GENERAL.

See Setting up for WLM (workload management) for more information about setting up additional transaction names.

Installing a z/OS UNIX System Services file system for the schema backend

The LDAP server must have write access to a z/OS UNIX System Services file system to store the schema backend. The schema backend is required to run the LDAP server.

Refer to the following documentation for more information about how to install zFS:

© Copyright IBM Corp. 2009
Optional Products

This section describes the products that may need to be installed and configured depending on how you want to set up your LDAP server.

<table>
<thead>
<tr>
<th>If you plan to use:</th>
<th>You must:</th>
<th>See:</th>
</tr>
</thead>
<tbody>
<tr>
<td>TDBM or GDBM backend (based on DB2)</td>
<td>Install the DB2 product and set up CLI and ODBC.</td>
<td>Installing and setting up DB2 for TDBM and GDBM (DB2-based) below</td>
</tr>
<tr>
<td></td>
<td>Note that if your LDAP server will be used only for accessing RACF information, it is not necessary to install DB2 or set up a DB2 database. See Setting up for SDBM for information about configuring the LDAP Server for accessing RACF information.</td>
<td></td>
</tr>
<tr>
<td>SDBM backend (based on RACF)</td>
<td>Install RACF.</td>
<td>Installing RACF for SDBM and native authentication</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LDBM, CDBM, or GDBM backend</td>
<td>Ensure that a z/OS UNIX System Services file system exists and has enough space to store the LDBM, CDBM, or GDBM (file-based) backend and that it can be written to by the LDAP server.</td>
<td>Installing a z/OS UNIX System Services file system for LDBM, CDBM (file-based), and CDBM backends</td>
</tr>
<tr>
<td>(based on z/OS UNIX System Services file system)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Program call (PC) support and the EXOP backend to support Policy Director extended operations</td>
<td>Install Policy Director and use SAF.</td>
<td>Installing and setting up Policy Director and SAF for z/OS Policy Director support</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Protect access to your LDAP server with Secure Sockets Layer (SSL) security or Transport Layer Security (TLS)</td>
<td>Install z/OS Cryptographic Services System SSL.</td>
<td>Installing System SSL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Encryption setup</td>
<td>Install ICSF (optional).</td>
<td>Installing ICSF for encryption</td>
</tr>
<tr>
<td>Kerberos authentication</td>
<td>Install Kerberos.</td>
<td>Installing Kerberos</td>
</tr>
<tr>
<td>Native authentication</td>
<td>Install a security server.</td>
<td>Installing RACF for SDBM and native authentication</td>
</tr>
</tbody>
</table>

Installing and setting up DB2 for TDBM and GDBM (DB2-based)

This section describes how to get DATABASE 2™ (DB2), Version 8 or higher, running and how to run the LDAP server using the TDBM and GDBM (DB2-based) backend. You should also have or have access to DB2 ODBC Guide and Reference, SC18-7423, DB2 Installation Guide, GC18-7418, and DB2 Application Programming and SQL Guide, SC18-7415.

Getting DB2 installed and set up for CLI and ODBC

Following are the steps to get DB2 installed:

1. Have your database system administrator install DB2, version 8 or later. If you will be running your LDAP server in multi-server mode on multiple images in a parallel sysplex, your administrator must
configure a DB2 data sharing group with members on each of the z/OS images on which an LDAP server instance will run. (See [Determining operational mode](#) for a description of the various operating modes in which the LDAP server may run.)

Make sure that the SMP/E jobs are a part of the DB2 installation. See the section about installing DB2 CLI in [DB2 ODBC Guide and Reference](#). Also, specify the user ID (for example, suxxxx) that should be granted database system administrator authority. This should be the ID you will log on with to run the SPUFI jobs to create the DB2 tables for the LDAP server. You need to find out the following information from your database administrator:

- DB2 subsystem name. For example, DSN8.
- DB2 server location (or data source). For example, LOC1.

In order to use a local or remote DB2 database, you must include a DDF record in your Bootstrap Data Set (BSDS). That DDF record must include a LOCATION keyword and an LUNAME keyword. If you are using a DB2 database that is on the local system (including a database that is set up for DB2 data sharing) the DDF component need not be started. If you are using a DB2 database that is on a remote system, the DDF component of DB2 must be configured and started on systems using the DB2 Call Level Interface (CLI). CLI is used by the LDAP server for requesting services from DB2. (The DB2 Call Level Interface is IBM’s callable SQL interface used by the DB2 family of products, based on the ISO Call Level Interface Draft International Standard specification and the Microsoft® Open Database Connectivity specification.)

It may be necessary to have your DB2 administrator set up buffer pools, TEMP space, and TEMP datasets for additional buffer pool sizes. By default, the LDAP server DB2-based backends use bufferpool BP0. The bufferpool choice and size (4K, 32K or other sizes) should be examined by your database system administrator to ensure they are large enough to meet the additional needs of the LDAP server, once you have loaded data into its database. The **DB2 RUNSTATS** utility should be used once data is loaded so that DB2 queries are optimized. See [Chapter 29, “Performance tuning”](#) for detailed information about running the DB2 **RUNSTATS** utility.

It may also be necessary to have your DB2 administrator increase the configured DB2 limits for MAX USERS and MAX BATCH CONNECT settings to accommodate the resources required by the LDAP server. These settings are controlled by the DB2 subsystem parameters CTHREAD and IDBACK, respectively, by way of installation panel DNSTIPE. These parameters are discussed in more detail in the [DB2 Installation Guide](#). The LDAP server requires the following connections to DB2:

- two connections for miscellaneous functions
- one connection for each communication thread, as defined by the **commThreads** option in the LDAP server configuration file
- one connection for each program call thread, as defined by the **pcThreads** option in the LDAP server configuration file
- one connection for each defined replica object, when basic replication is being used
  - if advanced replication is being used:
    - one connection for each defined replication agreement
    - one connection for each advanced replication plugin utility operation
    - five connections for system communication for each configured TDBM backend that is used with sysplex data sharing (sysplex threads for TDBM function shifting)

2. Enter:
   
   - `dsn start db2`

   from the image console and wait for DB2 to finish the DB2 initialization. The `dsn` is the DB2 subsystem name.

You can stop DB2 by entering:

   - `dsn stop db2`

   from the console.

**Note:** This may already be done when the system is re-ipled.
3. Edit and Submit DSNHLQ.SDSNSAMP(DSNTIJCL) where DSNHLQ is the high-level qualifier used during DB2 installation. See the section on setting up DB2 CLI runtime environment in DB2 ODBC Guide and Reference. You must run this from the user ID that has been granted the appropriate database authorities. This step establishes the environment needed for the LDAP server to use the CLI. It is often referred to as "binding the CLI plan". When binding the CLI plan, it must be bound using the bind option RELEASE(COMMIT), either by default (when no RELEASE option is specified on the bind statement) or by explicitly specifying the option on the bind statement (see DB2 Command Reference for information about bind options and syntax). Note the plan name used when editing this job, for example DSNACLI.

4. Create (Allocate) the DB2 CLI initialization file. A sample of the CLI initialization file can be found at DSNHLQ.SDSNSAMP(DSNAOINI). Create your own CLI initialization file and copy DSNHLQ.SDSNSAMP(DSNAOINI) into it. If a dataset is used for the CLI initialization file, it must not contain sequence numbers. Refer to the section on the DB2 CLI initialization file in DB2 ODBC Guide and Reference for more information about the contents of this file. Figure 2 shows a sample file. The example in Figure 2 shows a DSNAOINI file with values based on the examples used in this section. Items in the file that may need to be customized to your DB2 installation are in bold type. See your DB2 administrator for the values of these items for your installation.

```
; This is a comment line...
; Example COMMON stanza
[COMMON]
MVSDEFAULTSSID=DSN8

; Example SUBSYSTEM stanza for your DB2 subsystem name
[DSN8]
MVSATTACHTYPE=CAF
MVSATTACHTYPE=RRSAF
PLANNAME=DSNACLI

; Example DATA SOURCE stanza for your data source
[LOC1]
AUTOCOMMIT=0
CURSORHOLD=0
CONNECTTYPE=1
```

Figure 2. Sample DSNAOINI file

Choosing the MVSATTACHTYPE

The LDAP server can be set up to use either the Call Attachment Facility (CAF) or the Resource Recovery Services attachment facility (RRSAF) to access DB2. See DB2 ODBC Guide and Reference for more information about these choices.

Setting AUTOCOMMIT

To prevent data corruption, the LDAP server always uses a value of 0 for AUTOCOMMIT, regardless of the value specified in the DSNAOINI file.

Installing RACF for SDBM and native authentication

In order for your LDAP server to have access to RACF data, you must have RACF installed on your system and have a license for the z/OS Security Server. RACF is part of the z/OS Security Server. Refer to the following documentation for information about installing and configuring RACF:

- z/OS Program Directory
- z/OS Security Server RACF Security Administrator’s Guide
- z/OS Migration
The RACF Subsystem function of RACF must be defined and activated to allow the LDAP server to communicate with RACF through the SDBM backend. See \textit{z/OS Security Server RACF System Programmer's Guide} for information.

### Installing a z/OS UNIX System Services file system for LDBM, GDBM (file-based), and CDBM backends

The LDAP server must have write access to a z/OS UNIX System Services file system to store each LDBM, GDBM (file-based), and CDBM backend that is configured.

Refer to the following documentation for more information about how to install zFS:
- \textit{z/OS Program Directory}
- \textit{z/OS Distributed File Service zSeries File System Administration}

### Installing and setting up Policy Director and SAF for z/OS Policy Director support

In order for your LDAP server to provide z/OS Policy Director support, you must install and set up Policy Director. See \textit{Policy Director Authorization Services for z/OS and OS/390 Customization and Use} for instructions. Policy Director support also uses the System Authorization Facility (SAF) interface which is part of the z/OS environment and is always present on your z/OS system. \textit{z/OS Security Server RACF System Programmer's Guide} provides more information about SAF.

### Installing System SSL

In order for your LDAP server to provide SSL/TLS support, you must install z/OS Cryptographic Services System SSL and use STEPLIB, LPALIB, or LNKLST to make their libraries available. See \textit{z/OS Cryptographic Services System SSL Programming} for more information regarding SSL/TLS. Also, see \textit{Setting up for SSL/TLS} for details on configuring and using SSL/TLS with your LDAP server.

### Installing ICSF for encryption

The z/OS LDAP server supports AES, crypt, DES, MD5, and SHA for encryption of \texttt{userPassword} attribute values in the TDBM, LDBM, and CDBM backends. The \texttt{secretKey}, \texttt{replicaCredentials}, \texttt{ibm-replicaKeyPwd}, and \texttt{ibm-slapdMasterPw} attribute values in the TDBM, LDBM, and CDBM backends can be encrypted in either AES or DES. AES or DES keys can be stored in either an LDAPKEYS dataset or in Integrated Cryptographic Service Facility (ICSF). If ICSF is not used by the z/OS LDAP server, the encryption is done either by software or hardware assist (if it is available) depending upon the encryption method selected. For additional information about the LDAPKEYS dataset and the other encryption methods in the z/OS LDAP server, refer to \textit{Configuring for encryption}.

If using ICSF to perform AES or DES encryption, the processor must have hardware cryptographic support. All new processors have hardware cryptographic support, while some older processors optionally provide this support. Two other services of ICSF needed for AES or DES encryption in the z/OS LDAP server are the Key Generator Utility Program (KGUP) and the Cryptographic Key Data Set (CKDS). These are needed to generate and store the key and key label needed for AES or DES encryption of \texttt{userPassword}, \texttt{secretKey}, \texttt{replicaCredentials}, \texttt{ibm-replicaKeyPwd}, and \texttt{ibm-slapdMasterPw} attribute values. Refer to the information about managing cryptographic keys and using the Key Generator Utility Program in \textit{z/OS Cryptographic Services ICSF Administrator's Guide} for instructions on how to generate and store into CKDS a single-length data-encrypting key (also referred to as data key) for AES or DES encryption and how to set up the necessary security authorizations when using RACF to protect use of the key. It is important to remember to refresh both CKDS and RACF after you make the changes. ICSF must be configured so that the user ID under which the LDAP server runs can use ICSF services. Other portions of the ICSF book may be useful for general background information about ICSF and Cryptographic Keys.
Installing Kerberos

In order for your LDAP server to provide Kerberos support, you must install the Security Server Network Authentication and Privacy Service for z/OS which is the IBM implementation of Kerberos Version 5. See [z/OS Integrated Security Services Network Authentication Service Administration](#) for more information regarding Kerberos.

A sample Kerberos configuration file is provided in `/etc/skrb`. Refer to [z/OS Integrated Security Services](#) for details on setting up this file.
Chapter 4. Configuring an LDAP server using the dsconfig utility

The LDAP configuration utility, dsconfig, simplifies and automates the LDAP server configuration process for GDBM (DB2-based or file-based), TDBM, LDBM, EXOP, CDBM, and SDBM backends. The following table shows where to find specific information about the LDAP configuration utility within this topic.

<table>
<thead>
<tr>
<th>Description</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview of how the LDAP configuration utility works and information for determining if the utility is appropriate for your LDAP server configuration</td>
<td>Overview of the LDAP configuration utility</td>
</tr>
<tr>
<td>Details describing how to use the dsconfig utility</td>
<td>dsconfig utility</td>
</tr>
<tr>
<td>Step-by-step instructions describing how to configure an LDAP server</td>
<td>Steps for configuring an LDAP server</td>
</tr>
<tr>
<td>Configuration confirmation</td>
<td>Configuration confirmation</td>
</tr>
<tr>
<td>Advanced configuration options</td>
<td>Specifying advanced configuration options with the dsconfig utility</td>
</tr>
<tr>
<td>How to set the time zone</td>
<td>Setting the time zone</td>
</tr>
</tbody>
</table>

Overview of the LDAP configuration utility

The LDAP configuration utility helps you configure new LDAP server instances with minimal user interaction.

The LDAP configuration utility takes a profile file as input and generates a set of output members in a data set to facilitate an LDAP server configuration. The profile file is targeted for the System Administrator (or System Programmer) and the LDAP Administrator and it contains statements that must be updated with appropriate values. The LDAP configuration utility generates a series of JCL members, configuration files, and a procedure to start the LDAP server. The JCL jobs are segregated based on typical administrative roles in a z/OS installation and contain the required commands to configure the z/OS components used by the LDAP server. Each administrator is responsible for reviewing and submitting their JCL job. After all JCL jobs are submitted, each administrator is responsible for reviewing their job’s output and addressing any errors that may have occurred. Once all JCL jobs have completed successfully, the LDAP server can be started.
The minimal user interaction with the utility and the jobs it produces to update the required z/OS components results in a simplified approach to LDAP configuration. This approach allows novice LDAP users and administrators and even novice z/OS users to quickly deploy an LDAP server. In addition, the utility does not restrict the configuration of advanced LDAP features, such as referrals, replication, password encryption, and sysplex setup. See Specifying advanced configuration options with the dsconfig utility for more information.

Capabilities

- Allows the following backends to be configured:
  - LDBM - file-based
  - TDBM - DB2-based
  - GDBM - DB2-based for change log
  - GDBM - file-based for change log
  - SDBM - RACF-based for security
  - CDBM - file-based for configuration
  - EXOP - for extended operations

- Generates JCL jobs to accomplish the updates of all the z/OS components required for an LDAP server.

- Can configure advanced LDAP server features, including:
  - Encryption (dsconfig does not generate encryption key labels or data)
  - Referrals
  - Advanced replication
  - Basic replication
  - Change logging
  - Secure Sockets Layer (SSL) or Transport Layer Security (TLS) (dsconfig does not generate certificates or passwords)
  - Kerberos authentication
  - Native authentication
  - Extended operations (EXOP) backend (used for accessing Policy Directory information)
  - Plug-in extensions to the server

Restrictions

- Assumes that RACF is the security server in use. However, if RACF is not the security server in use, dsconfig could still be used. The resulting RACF JCL job will need to be converted to properly update the security server in use.
Does not allow configuring multiple backends of the same database type.
All values in the input files must be less than 66 bytes in length and must contain only printable
caracters in the IBM-1047 code page.
Cannot extend or enhance an existing LDAP server configuration.
Any manual updates to the output that the utility produces will be lost if you run the utility again with the
same output data set.
The database name value can be manually updated. Edit and update the value in the LDAP server
configuration file, generated by dsconfig.

If you cannot use the dsconfig utility because of one or more of these restrictions, see [Chapter 5, "Configuring an LDAP server without the dsconfig utility"] for information about alternate methods you can use to configure your LDAP server.

Running the dsconfig utility
In order to run the dsconfig utility in the shell, some environment variables need to be set properly, as
follows:

- STEPLIB - If SYS1.SIEALNKE is not in LNKLST, a STEPLIB will be required prior to running the
dsconfig utility. Issue:
explan STEPLIB=SYS1.SIEALNKE:$STEPLIB
- PATH - Ensure that /usr/lpp/ldap/sbin is added to the PATH environment variable. Issue:
explan PATH=/usr/lpp/ldap/sbin:$PATH
- NLSPATH - Ensure that /usr/lpp/ldap/lib/nls/msg/%L/%N is added to the NLSPATH environment
variable. Issue:
explan NLSPATH=/usr/lpp/ldap/lib/nls/msg/%L/%N:$NLSPATH
- LANG - Ensure that the LANG environment variable is set properly. Issue:
explan LANG=En_US.IBM-1047

dsconfig utility

Purpose
dsconfig takes as input a profile containing a set of options that, when updated, are used to create a set
of JCL members, configuration files, and the LDAP server start-up procedure. This output set configures
an LDAP server and sets up the z/OS system to run the server. Once the JCL jobs are submitted and are
successful, the LDAP server can be started.

Format
dsconfig -i profile_file [-s ds_file] [-a yes|no] [-d debug_level] [-?]

Parameters

- `-?` Specifies the usage.
- `-a yes|no` Specifies a preemptive answer for the dsconfig prompted question to
  overwrite an existing output dataset, from a previous execution of
dsconfig.
- `-d debug_level` Specifies the debug level of tracing for dsconfig. The debug level is
  specified in the same fashion as the debug level for the LDAP server, as
described in Table 20. The default is no debug messages.
- `-l profile_file` Specifies the path and file name of the input profile file. This file contains
  the base set of configuration options for the LDAP server as well as
  options necessary for the JCL output members.
dsconfig utility

| -s ds_file | Specifies the path and file name of the output server configuration file. **Note:** The LDAP configuration file generated with this option can be used with the method referred to on page Chapter 5, "Configuring an LDAP server without the dsconfig utility," on page 35. When this option is specified, no output dataset members are generated. |

Examples

To execute `dsconfig` in its most typical usage, where `ds.profile` is the input profile file and located in the current directory, enter:

```
dsconfig -i ds.profile
```

This example looks for the profile file in the `/home/u` directory and only generates a configuration file in the `/home/u` directory.

```
dsconfig -i /home/u/ds.profile -s /home/u/ds.conf
```

To enable error and trace debugging and preemptively skip the "overwrite dataset" question, enter:

```
dsconfig -i ds.profile -d error+trace -a yes
```

Input file description

The input file, `ds.profile`, must be modified before executing the LDAP configuration utility, `dsconfig`. Make a copy of the shipped `ds.profile` file from `/usr/lpp/ldap/etc` to a directory where the modifications can be made.

In this file there are statements containing a keyword and value which must have the appropriate value for the target system being configured. There is a brief description given for each keyword and value in the file. Figure 4 shows a sample portion of the `ds.profile` file:

```plaintext
# LDAPUSRID <user_id>
# Description:
#   User ID for the LDAP server to run under.
# ---------------------------------------
LDAPUSRID = GLDSRV
```

Figure 4. Sample portion of ds.profile

The LDAPUSRID statement, as shown above, has a preassigned value of GLDSRV. Above the statement there is some commentary describing the statement and its usage.

Most of the statements in the `ds.profile` are required and those that are not required are labelled as optional. Some statements in the `ds.profile` have preassigned values; however, they may not be valid on the target system being configured. Values must be provided for all required statements in the `ds.profile` file.

These keywords are intentionally left blank in `ds.profile` and their values must be provided:

- `ADMINDN` - the distinguished name (DN) of the administrator for this LDAP server
- `PROG SUFFIX` - specifies a two character suffix for the 'PROG' output member. This member contains the APF authorizations required for the LDAP server.
- The following four JCL jobcard options must be provided:
  - `APF_JOBCARD_1`
Examples and descriptions are included in the file. It is recommended you change the value for OUTPUT_DATASET, the name of the data set where all configuration utility output will be placed.

The ds.profile file embeds three other advanced input files. Information about these files can be found in Specifying advanced configuration options with the dsconfig utility. A common mistake, when specifying advanced configurations, is to copy and modify the advanced profiles but forget to modify the embedded path and name references for the modified advanced profiles in ds.profile.

Usage notes
1. The dsconfig utility will automatically configure a CDBM backend, when CDBM_USEADVANCEDREPLICATION in ds.profile is set to on.
2. The output from dsconfig is written to a partitioned data set that you specify in ds.profile. If the data set does not exist, the utility allocates the output data set for you.
   The dsconfig utility will prompt to confirm overwriting an existing output data set. When you specify -a yes on the dsconfig command line, the prompt will be skipped when the output data set exists. When -a no is specified, the existing data set will not be overwritten and dsconfig will end.
   When using the -s parameter, the value specified for the parameter is the path and file name of the LDAP server configuration file that will be created. The -a parameter can also be used to avoid the prompt that confirms overwriting an existing file.
   For advanced LDAP configuration as described in Specifying advanced configuration options with the dsconfig utility, the ds.profile contains pointers to the advanced configuration profile files. dsconfig expects to find the default pointers in the ds.profile.
3. The utility allows the configuration of an LDAP server which uses SSL/TLS. (See Setting up for SSL/TLS for details.) It does not, however, automate the process of generating SSL/TLS certificates.
4. The utility allows the configuration of an LDAP server which uses encryption. It does not, however, automate the process of generating encryption keys. (See Configuring for encryption for more information.)
5. Verify that the SYS1.SIEALNKE data set containing the LDAP code is in the LNKLST. If it is not in LNKLST, then STEPLIB must be used to locate this data set.
   The dsconfig utility does not provide an interface for adding STEPLIB statements to the started task procedure that it generates. Therefore, if an administrator wants to add STEPLIB statements to the started task procedure, the started task procedure must be manually updated and the following must occur:
   - The data sets specified in the new STEPLIB statements must be APF authorized.
   - When submitting the PRGMCTRL JCL job, the data sets specified in the new STEPLIB statements must be in the program control data set list.
   - The user ID specified on the LDAPUSRID statement in the ds.profile file must have read access to the data sets specified in the new STEPLIB statements.
6. The APF JCL job will not work on a system using JES3. JES3 users need to manually enter the following operator command in place of submitting the APF JCL job:
   - In SDSF, enter:
     /SET PROG=PROGsuffix
   - From the operator’s console, enter:
     SET PROG=PROGsuffix
   The suffix above is specified on the PROG_SUFFIX statement in the ds.profile file.
dsconfig utility

7. The PRGMCTRL and RACF jobs that dsconfig generates require that the definitions listed below exist in RACF before submission. If the definitions do not exist, the jobs will contain RACF errors in their output.
   a. To ensure that all required data sets are program controlled, the PRGMCTRL job requires that the PROGRAM. ** definition exists in RACF.
   b. To ensure that the user ID that appears on the LDAPUSRID statement in the ds.profile file has read permission on all required data sets, the RACF job requires that data set definitions exist for the following data sets in RACF.
      - CEEHLQ.** (where CEEHLQ appears on the CEEHLQ statement in the ds.profile file)
      - GLDHLQ.** (where GLDHLQ appears on the GLDHLQ statement in the ds.profile file)
      - GSKHLQ.** (where GSKHLQ appears on the GSKHLQ statement in the ds.profile file)
      - DSNHLQ.** (where DSNHLQ appears on the DSNHLQ statement in the ds.profile file)
      - CBCHLQ.** (where CBCHLQ appears on the CBCHLQ statement in the ds.profile file)
      - OUTPUT_DATASET_HLQ (where OUTPUT_DATASET_HLQ is the first qualifier of the data set name that appears on the OUTPUT_DATASET statement in the ds.profile file)

   Note: The server will operate properly even if the definitions required by the RACF JCL job do not exist, given that the user ID that appears on the LDAPUSRID statement in the ds.profile file has read permission on all required data sets.

8. Administrators with the appropriate authorizations must submit the JCL jobs generated by dsconfig on the target system.

9. You may get an error when running dsconfig from an rlogin session. This error return code of 12 can be ignored when running under the rlogin environment. The error is caused when both dsconfig and the rlogin environment free the OUTPUT_DATASET. No data is lost.

10. If an error occurs when submitting and running a dsconfig output JCL job or, if before submission, an administrator considers a value within the JCL job unsatisfactory, the administrator should not modify the JCL job directly. Instead, the administrator should update the appropriate profile files and perform all of the steps outlined in Steps for configuring an LDAP server again. To help determine the statements within a profile file that the administrator may need to update, at the top of every file generated by dsconfig there is a listing of statements that dsconfig used to create the output file. The administrator can use this listing to determine the exact statement within a profile file that should be updated. Note that when resubmitting all the JCL jobs dsconfig creates, many times JCL jobs for other components may have errors because of the duplication of a previous update. These messages may be ignored.

11. If a statement's value requires a length greater than 65 within the generated DSCONFIG member, the LDAP Administrator can move the DSCONFIG member out of the output data set into a data set where the record length is greater than 80 bytes and update the member in the new data set. Then, the System Administrator must update the CONFIG DD card in the generated procedure to point to the new data set.

12. The profile files do not replace the LDAP server configuration file; they are used to create an LDAP server configuration file to run the LDAP server.

13. It is recommended that you make all updates through the input files, running the utility again to recreate the jobs. Otherwise, if the generated JCL jobs are manually updated, those updates will be lost if the utility is run again using the same output data set.

14. Be sure to use a different output data set than is currently being used by other LDAP servers.

15. The DBCLI, DSNAOINI, and GDBSPUFI output members are created when the DB2-based TDBM backend is being configured.

16. The DBCLI, DSNAOINI, and GDBSPUFI output members are created when the DB2-based GDBM backend is being configured.

17. The output set of JCL jobs, configuration files and LDAP server startup procedure are rewritten whenever dsconfig is rerun. This only happens if the OUTPUT_DATASET option in the profile_file is
not changed. A common mistake is made when an LDAP Administrator makes changes to the output of dsconfig, particularly the configuration files, and then at some point later, reruns dsconfig and writes over those administrative changes.

Configuration roles and responsibilities

The output from the LDAP configuration utility consists of jobs and configuration files that finalize the LDAP server configuration. These jobs segregate z/OS updates based on typical administrative roles, allowing each administrator to control their component's updates. The typical administrative roles that are assumed to exist to configure an LDAP server are:

• System Administrator (or System Programmer)
• Database Administrator
• LDAP Administrator
• Security Administrator

Each administrator is responsible for updating input files in addition to reviewing and submitting jobs in the output members that the LDAP configuration utility produces for their component, as shown in Table 2.

Table 2. LDAP configuration utility roles and responsibilities

<table>
<thead>
<tr>
<th>Role</th>
<th>Responsibility</th>
<th>Input file name/type</th>
<th>Output members</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Administrator (or System Programmer)</td>
<td>APF authorization</td>
<td>ds.profile (main)</td>
<td>• APF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• PROGsuffix (suffix is</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>specified on the PROG_SUFFIX</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>statement in ds.profile)</td>
</tr>
<tr>
<td>Database Administrator</td>
<td>DB2, CLI</td>
<td>ds.db2.profile (advanced)</td>
<td>• TDBSPUFI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• GDBSPUFI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• DBCLI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• DSNAOINI</td>
</tr>
<tr>
<td>LDAP Administrator</td>
<td>LDAP server, Kerberos authentication, native authentication</td>
<td>ds.slapd.profile (advanced)</td>
<td>• user_id procedure (user ID is specified on the LDAPUSRID statement in ds.profile)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• DSENVVAR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• DSCONFIG</td>
</tr>
<tr>
<td>Security Administrator</td>
<td>RACF, SSL/TLS, password encryption</td>
<td>ds.racf.profile</td>
<td>• RACF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ds.slapd.profile (both</td>
<td>• PRGMCTRL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>files are advanced)</td>
<td></td>
</tr>
</tbody>
</table>

Figure 5 is a graphical representation showing the administrative roles, input files, and output members for dsconfig.
Steps for configuring an LDAP server

Use the following steps to configure with the configuration utility:

1. Copy the **ds.profile** file, found in `/usr/lpp/ldap/etc`, to a local directory and update it according to the commentary found in the file. (If you need to update the advanced configuration files mentioned in Table 2, you need to copy those files as well. See Specifying advanced configuration options with the **dsconfig** utility for more details on these files.)

Some statements in **ds.profile** do not have any preassigned values but are required for successful configuration. These are noted in the file. Assign values to all of these required statements, referring to information in the file above each statement for assistance. The intended audience of the **ds.profile** file is the System Administrator (or System Programmer) and the LDAP Administrator. The file contains information required from both administrators.

These **ds.profile** statements must be assigned values:

- ADMINDN
- PROG_SUFFIX
- APF_JOBCARD_1
- PRGCTRL_JOBCARD_1
- DB2_JOBCARD_1
- RACF_JOBCARD_1

Figure 5. LDAP configuration utility roles and responsibilities
The JOBCARD statements must be assigned regardless of their need. They are the job cards for the JCL output generated by `dsconfig`. The `dsconfig` utility determines what backends are configured at the same time it processes the JOBCARD statements. Note that the `dsconfig` utility will use only the appropriate job cards.

Refer to `ds.profile` for all statement descriptions and examples.

**Note:** Some statement values are case-sensitive and are denoted accordingly. Be sure to set up the editor to allow both upper and lower case letters to be specified.

Specify a distinguished name for ADMINDN that does not contain any of the suffixes that you are defining in the configuration. You will also need to specify a value for ADMINPW. This will enable the administrator to bind to the LDAP server and do the final setup of the LDAP server described in step [55](#) as part of that final setup, the adminDN option should be changed and the adminPW option removed from the configuration file.

2. Run the `dsconfig` utility. The utility will generate a set of members in a partitioned data set, as specified on the OUTPUT_DATASET statement in the `ds.profile` file.

   The `dsconfig` utility generates:
   - APF member: A JCL job which sets APF authorizations on libraries used by the LDAP server product.
   - DBCLI member: A JCL job which binds the CLI packages to DB2 and the DSNACLI plan.
   - DSCONFIG member: The LDAP server configuration file.
   - DSENVVAR member: The LDAP server environment variables file.
   - DSNAOINI member: The DB2 DSNAOINI initialization file.
   - GDBSPUFI member: A set of DB2 SQL statements, to be executed using the SPUFI tool, that defines database tables for the GDBM DB2-based backend.
   - PRGMCTRL member: A JCL job which will set Program Control on libraries used by the LDAP server product.
   - PROG_suffix member: A member needed for APF authorization.
   - RACF member: A JCL job which updates RACF to allow the LDAP server to run as a started task.
   - TDBSPUFI member: A set of DB2 SQL statements, to be executed using the SPUFI tool, that defines database tables for the TDBM backend.
   - A procedure member needed to start the LDAP server as a started task.

**Note:** DBCLI, DSNAOINI, TDBSPUFI and GDBSPUFI are conditionally generated when either of the DB2-based backends are configured.

3. Copy members and submit jobs.

   a. Copy the LDAP server started task procedure from the output data set to the target system’s procedure library. The name of the LDAP server started task procedure will be the name of the LDAP user ID specified on the LDAPUSRID statement in the `ds.profile` file. The preassigned name of the LDAP user ID is `GLDSRV`.

   b. Copy the generated PROG_suffix member (where `suffix` is specified on the PROG_SUFIX statement in the `ds.profile` file) from the output data set to the target system’s PARMLIB.

   c. Submit the following generated JCL jobs that can be found in the output data set in the following order:

      1) RACF member  
      2) APF member  
      3) DBCLI member, if TDBM or DB2-based GDBM is being configured

      **Note:** Be sure DB2 is started before submitting this job.

   4) PRGMCTRL member  
      The PRGMCTRL member is only required if an SDBM backend is being configured and Program Control is active.
dsconfig utility

4. Through the DB2 SPUFI interactive tool, submit the TDBSPUFI member, if a TDBM backend is being configured, and the GDBSPUFI member, if the DB2-based GDBM backend is being configured.

   **Note:** It is recommended that the table spaces in the GDBM database be set up for row level locking. See step 3 on page 47 in [Steps for configuring an LDAP server](#) for more information.

5. Start the LDAP server. The LDAP server can be started from SDSF or from the operator’s console.

   **Note:** The name of the LDAP server procedure is the same as the user ID specified on the LDAPUSRID statement. The preassigned value is `GLDSRV`.

   To start the LDAP server in SDSF, enter:
   ```
   /s user_id
   ```

   To start the LDAP server from the operator’s console, enter:
   ```
   s user_id
   ```

6. Finalize setup of the LDAP server.
   
   a. If TDBM, LDBM, or CDBM is configured, modify the LDAP server schema entry to contain the schema needed for your usage of these backends. The `schema.user.ldif` and `schema.IBM.ldif` files found in the `/usr/lpp/ldap/etc` directory may contain the schema you need. `schema.IBM.ldif` requires that you first load `schema.user.ldif`.

      **Note:** The distinguished name (DN) of the LDAP server schema is `cn=schema`. If the ldif file containing your schema has a DN of `cn=schema,suffix`, then update the file to change the DN to `cn=schema`.

      Use the `ldapmodify` utility to modify the schema entry.

      ```
      ldapmodify -h ldaphost -p ldapport -D binddn -w passwd -f file
      ```

      where:

      `ldaphost`
      Is the host name of the system where the LDAP server is running.

      `ldapport`
      Is the TCP/IP port on which the LDAP server is running. The port is specified in an advanced profile file, `ds.slapd.profile`, on the LISTEN statement. The preassigned value is 389.

      `binddn` Is the administrator DN of the LDAP server. The administrator DN is specified in the `ds.profile` file on the ADMINDN statement. This value is required and not preassigned.

      `passwd` Is the administrator password of the LDAP server. The administrator password is specified in the `ds.profile` file on the ADMINPW statement. The example value is "secret".

      `file` Is a file containing modifications to the schema entry in LDIF format. More information about the schema can be found in Chapter 14, “LDAP directory schema.”

      Following is an example of using `ldapmodify` to modify the schema entry:

      ```
      ldapmodify -h myhost -p 389 -D "cn=Admin" -w secret -f /usr/lpp/ldap/etc/schema.user.ldif
      ```

      [More information about `ldapmodify` can be found in IBM Tivoli Directory Server Client Programming for z/OS](#)

      Multiple schemas may need to be loaded before applications that use the directory will work. For example, in addition to the `schema.user.ldif` schema file, it is common for directory applications to require the elements defined in the `schema.IBM.ldif` schema file.

   b. Load the suffix entries for each configured TDBM and LDBM backend. Suffix entries are specified in the `ds.profile` file on the TDBM_SUFFIX and LDBM_SUFFIX statements.
Note: If you intend to load large amounts of data in LDIF format into a TDBM backend, see `ldif2ds` utility for instructions on using the `ldif2ds` utility. In this case, do not load the suffix entry separately. Include the suffix instead with the rest of the entries to be loaded by `ldif2ds`.

Use the `ldapadd` utility to load the suffix entry.

```
ldapadd -h ldaphost -p ldapport -D binddn -w passwd -f file
```

See descriptions of `ldaphost`, `ldapport`, `binddn`, and `passwd` above. `file` is a file containing the entries to be loaded in LDIF format.

For example, if `suffix.ldif` contains the following suffix entry:

```
dn: o=Your Company
objectclass: organization
o: Your Company
```

then the suffix entry can be added by `ldapadd` as follows:

```
ldapadd -h myhost -p 389 -D "cn=Admin" -w secret -f suffix.ldif
```

More information about `ldapadd` can be found in *IBM Tivoli Directory Server Client Programming for z/OS*.

c. Set an appropriate ACL for controlling access to change log entries for the GDBM backend, if configured. See Chapter 26, “Change logging” for more information.

d. After initial set up of the LDAP server, it is recommended that you remove the `adminPW` option from the LDAP server configuration file. See Establishing the administrator DN and basic replication replica server DN and passwords for more information.

To confirm the LDAP server is configured and ready for client requests, see Configuration confirmation.

To load the data in LDIF format into a TDBM backend, you can use `ldif2ds` or `ldapadd`. However, if you intend to load more than 100,000 directory entries, use `ldif2ds`.

**Configuration confirmation**

Following is an optional Installation Verification Procedure (IVP) to confirm that the LDAP server configuration is successful.

Run the `ldapsearch` utility to verify the configuration.

```
ldapsearch -h ldaphost -p ldapport -D binddn -w passwd -s base -b "" "objectclass=""
```

See descriptions of `ldaphost`, `binddn`, `passwd`, and `ldapport` above.

The `-s base` specifies the base scope for the search and the `-b ""` specifies the root DSE as the base.

The result of this search contains a list of all naming contexts supported by the LDAP server. For example, if both TDBM and SDBM are configured, the result of the search contains both naming contexts (suffixes) listed.

Following is an example using `ldapsearch` to verify a configuration:

```
ldapsearch -h myhost -p 389 -D cn=admin -w secret -s base -b "" "objectclass=""
```

If the naming context is not returned, an error message is returned indicating a problem.

More information about `ldapsearch` can be found in *IBM Tivoli Directory Server Client Programming for z/OS*.
Specifying advanced configuration options with the dsconfig utility

There are advanced configuration options specified in the following input files:

- **ds.db2.profile** (DB2 input file)
- **ds.racf.profile** (RACF input file)
- **ds.slapd.profile** (SLAPD input file)

These advanced profile files are located in the `/usr/lpp/ldap/etc` directory and are all included by the **ds.profile** file. Every statement contains a preassigned value in the advanced profile files.

To modify these optional statements:

1. Copy the desired files to a local directory and update them. Each input file should be modified by the appropriate administrator (see Table 2 for LDAP configuration utility roles and responsibilities).
2. Update the **ds.profile** to correctly include those modifications. Near the end of **ds.profile** there are three statements:

   ```
   SLAPD_PROFILE = /usr/lpp/ldap/etc/ds.slapd.profile
   DB2_PROFILE = /usr/lpp/ldap/etc/ds.db2.profile
   RACF_PROFILE = /usr/lpp/ldap/etc/ds.racf.profile
   ```

   Update these statements to show the new paths of the files you modified. Here is an example where the modified versions of **ds.db2.profile** and **ds.slapd.profile** are in a different directory (`/home/u`), and **ds.racf.profile** is not changed:

   ```
   SLAPD_PROFILE = /home/u/ds.slapd.profile
   DB2_PROFILE = /home/u/ds.db2.profile
   RACF_PROFILE = /usr/lpp/ldap/etc/ds.racf.profile
   ```

Advanced configuration options may require additional instructions not covered by the LDAP configuration utility. The following table provides references for those instructions.

<table>
<thead>
<tr>
<th>Configuration option</th>
<th>More information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Referrals</td>
<td>Chapter 27, “Referrals”</td>
</tr>
<tr>
<td>Basic replication</td>
<td>Chapter 23, “Basic replication”</td>
</tr>
<tr>
<td>Advanced replication</td>
<td>Chapter 24, “Advanced replication”</td>
</tr>
<tr>
<td>Password encryption</td>
<td>Configuring for encryption</td>
</tr>
<tr>
<td>Multi-server</td>
<td>Determining operational mode</td>
</tr>
<tr>
<td>Kerberos authentication</td>
<td>Chapter 17, “Kerberos authentication”</td>
</tr>
<tr>
<td>Native authentication</td>
<td>Chapter 18, “Native authentication”</td>
</tr>
<tr>
<td>CRAM-MD5 and DIGEST-MD5 authentication</td>
<td>Chapter 19, “CRAM-MD5 and DIGEST-MD5 authentication”</td>
</tr>
<tr>
<td>Extended operations to access Policy Director data</td>
<td>Setting up for Policy Director extended operations, serverEtherAddress option on page 100</td>
</tr>
<tr>
<td>Entry UUID support</td>
<td>Chapter 26, “Change logging”</td>
</tr>
<tr>
<td>Change logging</td>
<td>Chapter 26, “Change logging”</td>
</tr>
<tr>
<td>Plug-in extensions to the server</td>
<td><strong>IBM Tivoli Directory Server Plug-in Reference for z/OS</strong></td>
</tr>
<tr>
<td>Other LDAP server options</td>
<td>Chapter 8, “Customizing the LDAP server configuration”</td>
</tr>
</tbody>
</table>

Notes:

1. If the uid specified on the LDAPUID statement in the **ds.racf.profile** file is greater than 0 and the port specified with the LISTEN statement in the **ds.slapd.profile** file is less than 1024, the LISTEN statements generated in the DSCONFIG member commentary must be added to the **PROFILE.TCPIP** dataset on the target system. These LISTEN statements are located in the commentary directly above the `listen` option in the generated DSCONFIG member. See [z/OS Communications Server: IP Configuration Guide](https://www.ibm.com/support/knowledgecenter/en/SSLTBK_21.3.0/com.ibm.zos.v2r11.cicfg_iptc.pdf) for more information about the **PROFILE.TCPIP** dataset.
2. When using `dsconfig` to configure an LDAP server with Kerberos enabled, the user ID that the server runs under is created with a temporary password. This temporary password is then immediately removed. This is required to complete the configuration of an LDAP server with Kerberos enabled. See Chapter 17, “Kerberos authentication” for more details.

### Setting the time zone

The LDAP server uses time values returned by the operating system when it records server activity or when it generates LDAP trace records. The LDAP server assumes that time values are in Universal Time Coordinated (UTC) format. The UTC time value is mapped to a (local) time zone value as specified by the `TZ` environment variable. By default, `TZ` is set to GMT0.

To change the time zone value, you need to edit the `ds slapd.profile` file. See Specifying advanced configuration options with the dsconfig utility for more information about updating `ds slapd.profile`. In `ds slapd.profile`, uncomment the `TIMEZONE` variable and set the value as desired. For more information about time zones, see z/OS XL C/C++ Programming Guide.
dsconfig utility
Chapter 5. Configuring an LDAP server without the dsconfig utility

This topic lists the necessary steps involved in configuring your LDAP server if you do not use the `dsconfig` utility. It may be necessary for you to use this method instead of the `dsconfig` utility, as some LDAP configuration scenarios cannot be set up with the `dsconfig` utility. More information about the `dsconfig` utility is in Chapter 4, “Configuring an LDAP server using the dsconfig utility.”

This topic contains:
- A roadmap which provides LDAP server configuration steps based on which backends and options you choose to configure, such as:
  - SDBM backend (RACF-based)
  - TDBM backends (general purpose directory, DB2-based)
  - LDBM backends (general purpose directory, file-based)
  - CDBM backend (configuration directory, file-based)
  - GDBM backend (change log directory, DB2-based and file-based)
  - EXOP backend for accessing Policy Director data
  - Plug-in extensions to the server
  - Secure Sockets Layer (SSL) or Transport Layer Security (TLS)
  - Password encryption
  - Kerberos authentication
  - Native authentication
- A list of configuration variables and their interactions
- Setting the time zone

**LDAP server configuration roadmap**

Table 3 lists the set up and configuration tasks you must complete depending on which backends and options your LDAP server needs.

<table>
<thead>
<tr>
<th>Task</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>If you are configuring an SDBM (RACF-based) backend, you must:</td>
<td></td>
</tr>
<tr>
<td>Install RACF</td>
<td>Installing RACF for SDBM and native authentication</td>
</tr>
<tr>
<td>Set up the user ID and security for the LDAP server</td>
<td>Setting up a user ID for your LDAP server</td>
</tr>
<tr>
<td>Set up the LDAP server for SDBM</td>
<td>Setting up for SDBM</td>
</tr>
<tr>
<td>Configure the LDAP server</td>
<td>Chapter 8, “Customizing the LDAP server configuration”</td>
</tr>
<tr>
<td>Run the LDAP server and finalize setup</td>
<td>Chapter 9, “Running the LDAP server”</td>
</tr>
<tr>
<td>See other SDBM-specific information</td>
<td>Chapter 16, “Accessing RACF information”</td>
</tr>
<tr>
<td>If you are configuring a TDBM (DB2-based) backend, you must:</td>
<td></td>
</tr>
<tr>
<td>Install and set up DB2</td>
<td>Installing and setting up DB2 for TDBM and GDBM (DB2-based)</td>
</tr>
<tr>
<td>Set up the user ID and security for the LDAP server</td>
<td>Setting up a user ID for your LDAP server</td>
</tr>
<tr>
<td>Task</td>
<td>Topic</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>Create the DB2 database and table spaces for TDBM</td>
<td>Creating the DB2 database and table spaces for TDBM or GDBM</td>
</tr>
<tr>
<td>Set up the LDAP server for TDBM</td>
<td>Setting up for TDBM</td>
</tr>
<tr>
<td>Configure the LDAP server</td>
<td>Chapter 8, “Customizing the LDAP server configuration”</td>
</tr>
<tr>
<td>Run the LDAP server and finalize setup (including adding schema and</td>
<td>Chapter 9, “Running the LDAP server”</td>
</tr>
<tr>
<td>loading data)</td>
<td></td>
</tr>
<tr>
<td>If you are configuring an LDBM backend, you must:</td>
<td></td>
</tr>
<tr>
<td>Install a z/OS UNIX System Services file system</td>
<td>Installing a z/OS UNIX System Services file system for LDBM, GDBM</td>
</tr>
<tr>
<td></td>
<td>(file-based), and CDBM backends</td>
</tr>
<tr>
<td>Set up the user ID and security for the LDAP server</td>
<td>Setting up a user ID for your LDAP server</td>
</tr>
<tr>
<td>Set up the LDAP server for LDBM</td>
<td>Setting up for LDBM</td>
</tr>
<tr>
<td>Configure the LDAP server</td>
<td>Chapter 8, “Customizing the LDAP server configuration”</td>
</tr>
<tr>
<td>Run the LDAP server and finalize setup (including adding schema and</td>
<td>Chapter 9, “Running the LDAP server”</td>
</tr>
<tr>
<td>loading data)</td>
<td></td>
</tr>
<tr>
<td>If you are configuring a CDBM backend, you must:</td>
<td></td>
</tr>
<tr>
<td>Install a z/OS UNIX System Services file system</td>
<td>Installing a z/OS UNIX System Services file system for LDBM, GDBM</td>
</tr>
<tr>
<td></td>
<td>(file-based), and CDBM backends</td>
</tr>
<tr>
<td>Set up the user ID and security for the LDAP server</td>
<td>Setting up a user ID for your LDAP server</td>
</tr>
<tr>
<td>Set up the LDAP server for CDBM</td>
<td>Setting up for CDBM</td>
</tr>
<tr>
<td>Configure the LDAP server</td>
<td>Chapter 8, “Customizing the LDAP server configuration”</td>
</tr>
<tr>
<td>Run the LDAP server and finalize setup (including adding schema and</td>
<td>Chapter 9, “Running the LDAP server”</td>
</tr>
<tr>
<td>loading data)</td>
<td></td>
</tr>
<tr>
<td>If you are configuring a GDBM (DB2-based) backend, you must:</td>
<td></td>
</tr>
<tr>
<td>Install and set up DB2</td>
<td>Installing and setting up DB2 for TDBM and GDBM (DB2-based)</td>
</tr>
<tr>
<td>Set up the user ID and security for the LDAP server</td>
<td>Setting up a user ID for your LDAP server</td>
</tr>
<tr>
<td>Create the DB2 database and table spaces for GDBM</td>
<td>Creating the DB2 database and table spaces for TDBM or GDBM</td>
</tr>
<tr>
<td>Set up the LDAP server for GDBM</td>
<td>Configuring file-based GDBM</td>
</tr>
<tr>
<td>Configure the LDAP server</td>
<td>Chapter 8, “Customizing the LDAP server configuration”</td>
</tr>
<tr>
<td>Run the LDAP server and finalize setup (including setting access</td>
<td>Chapter 9, “Running the LDAP server”</td>
</tr>
<tr>
<td>control)</td>
<td></td>
</tr>
</tbody>
</table>
### Table 3. LDAP server configuration roadmap (continued)

<table>
<thead>
<tr>
<th>Task</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>If you are configuring a GDBM (file-based) backend, you must:</td>
<td></td>
</tr>
<tr>
<td>Install a z/OS UNIX System Services file system</td>
<td>Installing a z/OS UNIX System Services file system for LDBM, GDBM (file-based), and CDBM backends</td>
</tr>
<tr>
<td>Set up the user ID and security for the LDAP server</td>
<td>Setting up a user ID for your LDAP server</td>
</tr>
<tr>
<td>Set up the LDAP server for GDBM</td>
<td>Configuring file-based GDBM</td>
</tr>
<tr>
<td>Configure the LDAP server</td>
<td>Chapter 8, “Customizing the LDAP server configuration”</td>
</tr>
<tr>
<td>Run the LDAP server and finalize setup (including setting access control)</td>
<td>Chapter 9, “Running the LDAP server”</td>
</tr>
<tr>
<td>If you are configuring an EXOP (extended operation) backend, you must:</td>
<td></td>
</tr>
<tr>
<td>Install Policy Director</td>
<td>Installing and setting up Policy Director and SAF for z/OS Policy Director support</td>
</tr>
<tr>
<td>Set up the user ID and security for the LDAP server</td>
<td>Setting up a user ID for your LDAP server</td>
</tr>
<tr>
<td>Set up the LDAP server for EXOP</td>
<td>Setting up for Policy Director extended operations</td>
</tr>
<tr>
<td>Configure the LDAP server</td>
<td>Chapter 8, “Customizing the LDAP server configuration”</td>
</tr>
<tr>
<td>Run the LDAP server</td>
<td>Chapter 9, “Running the LDAP server”</td>
</tr>
<tr>
<td>If you are configuring a plug-in extension to the LDAP server, you must:</td>
<td></td>
</tr>
<tr>
<td>Create the extension</td>
<td>IBM Tivoli Directory Server Plug-in Reference for z/OS</td>
</tr>
<tr>
<td>Set up the extension</td>
<td>IBM Tivoli Directory Server Plug-in Reference for z/OS</td>
</tr>
<tr>
<td>Configure the LDAP server</td>
<td>IBM Tivoli Directory Server Plug-in Reference for z/OS and Chapter 8, “Customizing the LDAP server configuration”</td>
</tr>
<tr>
<td>Run the LDAP server</td>
<td>Chapter 9, “Running the LDAP server”</td>
</tr>
<tr>
<td>If your LDAP server is going to support Secure Sockets Layer (SSL) or Transport Layer Security (TLS), you must:</td>
<td></td>
</tr>
<tr>
<td>Install and set up System SSL</td>
<td>Installing System SSL</td>
</tr>
<tr>
<td>Set up the LDAP server for SSL/TLS</td>
<td>Setting up for SSL/TLS</td>
</tr>
<tr>
<td>Configure the LDAP server</td>
<td>Chapter 8, “Customizing the LDAP server configuration”</td>
</tr>
<tr>
<td>Run the LDAP server</td>
<td>Chapter 9, “Running the LDAP server”</td>
</tr>
<tr>
<td>If your LDAP server is going to use encryption of attribute values, you must:</td>
<td></td>
</tr>
<tr>
<td>Install ICSF if using ICSF to store DES or AES keys</td>
<td>Installing ICSF for encryption</td>
</tr>
</tbody>
</table>
Table 3. LDAP server configuration roadmap (continued)

<table>
<thead>
<tr>
<th>Task</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set up the LDAP server for encryption</td>
<td>Configuring for encryption</td>
</tr>
<tr>
<td>Configure the LDAP server</td>
<td>Chapter 8, “Customizing the LDAP server configuration”</td>
</tr>
<tr>
<td>Run the LDAP server</td>
<td>Chapter 9, “Running the LDAP server”</td>
</tr>
</tbody>
</table>

If your LDAP server is going to support Kerberos Authentication, you must:

| Install and configure Kerberos             | Installing Kerberos                           |
| Start the KDC                              | Setting up for Kerberos                       |
| Update the Kerberos segment of the LDAP server’s user ID | Setting up for Kerberos                       |
| Generate the LDAP server’s key table file (optional) | Setting up for Kerberos                       |
| Configure the LDAP server                 | Chapter 8, “Customizing the LDAP server configuration” |
| Run the LDAP server                        | Chapter 9, “Running the LDAP server”          |

If your LDAP server is going to support native authentication, you must:

| Install RACF or other security server      | Installing RACF for SDBM and native authentication |
| Configure the LDAP server                  | Chapter 8, “Customizing the LDAP server configuration” |
| Run the LDAP server                        | Chapter 9, “Running the LDAP server”              |

Preparing for configuration variable interactions

Some of the variables involved in configuring the LDAP server and its related products are used in more than one file or configuration step. It is essential that the same value be used each time the variable is referenced. The following table lists the interactions of each such variable, for each backend.

Table 4. Configuration variable interactions

<table>
<thead>
<tr>
<th>Variable</th>
<th>Used in:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TDBM or GDBM (DB2-based) Backend</td>
</tr>
<tr>
<td>DB2 subsystem ID</td>
<td>• SYSTEM(DSN) value in CLI bind JCL, DSNTIJCL (see Step 3)</td>
</tr>
<tr>
<td></td>
<td>• MVSDEFAULTSSID and SUBSYSTEM values in CLI initialization file, DSNAOINI (see Step 4)</td>
</tr>
<tr>
<td>Plan name</td>
<td>• PLAN value in CLI bind JCL, DSNTIJCL (see Step 3)</td>
</tr>
<tr>
<td></td>
<td>• PLANNNAME value in CLI initialization file, DSNAOINI (see Step 4)</td>
</tr>
<tr>
<td></td>
<td>• The zzz value in SQL commands to grant permissions to LDAP user (see Step 5)</td>
</tr>
<tr>
<td>Database name</td>
<td>• -DB_NAME- value in SPUFI script to create database, GLDHLQ.SGLDSAMP(DSTDBMDB) (see Step 3)</td>
</tr>
<tr>
<td></td>
<td>• The yyy value in SQL commands to grant permissions to LDAP user (see Step 5)</td>
</tr>
<tr>
<td>Database owner</td>
<td>• -DB_USERID- value in SPUFI script to create database, GLDHLQ.SGLDSAMP(DSTDBMDB) (see Step 2)</td>
</tr>
<tr>
<td></td>
<td>• The dbuserid value in LDAP server configuration file, ds.conf (see page 80)</td>
</tr>
</tbody>
</table>
### Table 4. Configuration variable interactions (continued)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Used in:</th>
</tr>
</thead>
</table>
| CLI initialization file name | • The dsnaoini value in LDAP server configuration file, ds.conf (see page 82)  
• DSNAOINI DD value in JCL for LDAP server and utilities |
| User ID running LDAP server, ds2ldif and ldif2ds utilities | • LDAPSrv value in RACF commands to create user ID (see Requirements for a user ID that runs the LDAP server)  
• The xxx value in SQL commands to grant permissions to LDAP user (see Step 5)  
• LDAPSrv value in RACF commands to create SSL/TLS key ring (see Creating and using key databases, key rings, or PKCS #11 tokens)  
• LDAPSrv value in RACF commands to create started task (see Defining the started task for the LDAP server) |
| Server name | • DATA SOURCE value in CLI initialization file, DSNAOINI (see Step 4)  
• The servername value in LDAP server configuration file, ds.conf (see page 101) |
| SDBM Backend |  
| User ID running LDAP server or utilities | • LDAPSrv value in RACF commands to create user ID (see Requirements for a user ID that runs the LDAP server)  
• LDAPSrv value in RACF commands to create SSL/TLS key ring (see Creating and using key databases, key rings, or PKCS #11 tokens)  
• LDAPSrv value in RACF commands to create started task (see Defining the started task for the LDAP server) |

### Setting the time zone

The LDAP server uses time values returned by the operating system when it records server activity or when it generates LDAP trace records. The LDAP server assumes that time values are in Universal Time Coordinated (UTC) format. The UTC time value is mapped to a (local) time zone value as specified by the TZ environment variable. By default, TZ is set to GMT0.

If you have not already done this, copy `/usr/lpp/ldap/etc/ds.envvars` to the `/etc/ldap` directory. Edit `/etc/ldap/ds.envvars`, uncomment the TZ environment variable and set the value as desired. For more information about time zones, see [z/OS XL C/C++ Programming Guide](https://www.ibm.com/support/knowledgecenter/SSTRZ7/latest/index.jsp?topic=/com.ibm.zos.r16.0.0/r16_inv/inv_chap5.htm).

When started, the LDAP server will read an environment variable file. The default file is `/etc/ldap/ds.envvars`. This default can be changed by setting the environment variable LDAP_DS_ENVVARS_FILE to the full path name of the desired environment variable file. LDAP server time stamps are then generated using a UTC value and the time zone value.
Chapter 6. Setting up the user ID and security for the LDAP server

This topic discusses how to configure and set up the products needed by your LDAP server.

In this section, some of the examples and descriptions reflect assumptions that may not apply to your environment. Following are descriptions of these assumptions, with guidance on how to use this information if they do not apply to your environment:

- Some examples use Resource Access Control Facility (RACF). You can use any z/OS external security manager that has equivalent support. You must substitute the appropriate procedures for any examples that use RACF.
- The default name /usr/lpp/ldap is used for the directory in which you installed the LDAP server product. If you used a different name, substitute that name in the examples and descriptions where applicable.
- The language setting En_US.IBM-1047 is used for the locale in which you are running the LDAP server. This setting is used in the names of several directories that are referred to in this information. If you are using a different language setting, substitute that setting in the examples and descriptions where applicable. You must also specify this setting as the value of the LANG parameter in the environment variables file as described in Chapter 12, “Internationalization support.” The default environment variables file already sets LANG to En_US.IBM-1047.
- The name LDAPSRV is used for the user ID that runs the LDAP server. If you use a different name, substitute that name in the examples and descriptions where applicable.
- The name of the production directory is /etc/ldap. If you use a different name, you must symbolic link the names of the appropriate files in your directory to the /etc/ldap directory.

Setting up a user ID for your LDAP server

It is recommended that a separate user ID be established to run the LDAP server. The user ID that runs the LDAP server must have the following attributes:

- The user ID must have read access to the BPX.WLMSERVER profile in the FACILITY class.
- If you defined the BPX.SERVER profile in the FACILITY class, the user ID must have update access to the profile.
- The user ID must have read access to the data sets defined in the startup procedure.
- The user ID must have read access to the /etc/ldap directory.
- If you have an LDBM, GDBM (file-based), or CDBM backend and the databaseDirectory configuration option is specified in the LDAP server configuration file, the user ID must be able to create the specified directory if it does not already exist. If the directory exists, then the user ID must have read/write access to it.
- If the schemaPath configuration option is specified in the LDAP server configuration file, the user ID must be able to create the specified directory if it does not already exist. If the directory exists, then the user ID must have read/write access to it.
- If the schemaPath configuration option is not specified or you have an LDBM, GDBM (file-based), or CDBM backend and have not specified the databaseDirectory configuration option, then the user ID must be able to create directories under /var or the /var/ldap directory must already exist. If the /var/ldap directory exists, then the user ID must have read/write access to the directory.

Requirements for a user ID that runs the LDAP server

Any user ID can be used to run the LDAP server. The examples in this topic use a user ID of LDAPSRV in the commands provided.

Note that if the UID of the user ID running the LDAP server is not zero, all console messages produced by the LDAP server are accompanied by a BPXM023I message identifying the user writing to the console.
The user ID performing the RACF commands in the following examples requires RACF SPECIAL authority.

You can use the RACF commands in the following example to define the user ID that will run the LDAP server (substitute appropriate values for UID and GID).

```
ADDCGROUP LDAPGRP SUPGROUP(SYS1) OMVS(GID(2))
ADDSUSER LDAPSRV DFLTGRP(LDAPGRP) OMVS(UID(1) PROGRAM('/bin/sh'))
```

The following RACF commands give the LDAP server access to the Workload Manager (WLM). This is required when starting the LDAP server even if WLM is not being used to classify or prioritize work within the LDAP server.

```
RDEFINE BPX.WLMSERVER CLASS(FACILITY) UACC(NONE)
PERMIT BPX.WLMSERVER CLASS(FACILITY) ID(LDAPSRV) ACCESS(READ)
SETROPTS RACLIST(FACILITY) REFRESH
```

The following RACF commands should only be entered if the BPX.SERVER profile is defined.

```
PERMIT BPX.SERVER CLASS(FACILITY) ID(LDAPSRV) ACCESS(UPDATE)
SETROPTS RACLIST(FACILITY) REFRESH
```

If the user ID has a nonzero UID value, issue these RACF commands so the LDAP server can perform `chmod` and `chown` commands on the LDBM, GDBM (file-based), CDBM, and schema backend files and directories:

```
RDEFINE UNIXPRIV SUPERUSER.FILESYS.CHOWN UACC(NONE)
PERMIT SUPERUSER.FILESYS.CHOWN CLASS_UNIXPRIV ID(LDAPSRV) ACC(READ)

RDEFINE UNIXPRIV SUPERUSER.FILESYS.CHANGEPARMS UACC(NONE)
PERMIT SUPERUSER.FILESYS.CHANGEPARMS CLASS_UNIXPRIV ID(LDAPSRV) ACC(READ)

SETROPTS CLASSACT(UNIXPRIV)
SETROPTS RACLIST(UNIXPRIV) REFRESH
```

If you are going to set up more than one LDAP server on the same system, a separate user ID should be used for each one.

**Additional setup when using SDBM**

If you plan to use an SDBM backend, the following RACF commands must be entered to set up the user ID that will run the LDAP server:

```
RDEFINE FACILITY IRR.RUSERMAP UACC(NONE)
PERMIT IRR.RUSERMAP CLASS(FACILITY) ID(LDAPSRV) ACCESS(READ)

SETROPTS RACLIST(FACILITY) REFRESH
```

The SDBM backend also supports the RACF functions that search for users and groups with a given UID or GID value, control sharing user UID and group GID values, and retrieve a user password or password phrase envelope. Usage of these functions requires additional RACF configuration and profiles, as described in the RACF documentation.

**Additional setup for RACF PROXY segment and SDBM**

The SDBM backend supports the PROXY segment within the RACF user profile. If you intend to use SDBM to set the BINDPW value in the PROXY segment, RACF requires that the KEYSMSTR class profile LDAP.BINDPW.KEY be created with the SSIGNON segment.

- To create the LDAP.BINDPW.KEY profile in the KEYSMSTR class, use the KEYMASKED sub-operand if no cryptographic product is installed on your system:

```
RDEFINE KEYSMSTR LDAP.BINDPW.KEY SSIGNON(KEYMASKED(key-value))
```

- Or, use the KEYENCRYPTED sub-operand if a cryptographic product is installed:

```
RDEFINE KEYSMSTR LDAP.BINDPW.KEY SSIGNON(KEYENCRYPTED(key-value))
```

`key-value` is a Secured Sign-on Application key and must be specified as a string of 16 hexadecimal characters.
Then, activate the KEYSMSTR class:

```
SETROPTS CLASSACT(KEYSMSTR)
```

See [z/OS Security Server RACF Command Language Reference](zos_command_language_reference) for details on using these RACF commands and [z/OS Security Server RACF Security Administrator's Guide](zos_security_administrator_guide) for information about creating and using profiles.

**Additional setup for sysplex**

If you plan to run the LDAP server in a sysplex group, the user ID that will run the LDAP server must have READ access to the GLD.XCF.GROUP.<group_name> resource in the FACILITY class, where <group_name> is the value of the `serverSysplexGroup` option in the LDAP server configuration file. For example, if the `serverSysplexGroup` value is LDAP, then issue the following RACF commands:

```
RDEFINE FACILITY GLD.XCF.GROUP.LDAP UACC(NONE)
PERMIT GLD.XCF.GROUP.LDAP CLASS(FACILITY) ID(LDAPSRV) ACCESS(READ)
SETROPTS RACLIST(FACILITY) REFRESH
```

**Defining the Kerberos identity**

If you plan to enable Kerberos support you need to associate a Kerberos identity with the server's user ID and generate a Kerberos key. The following RACF command must be entered:

```
ALTUSER LDAPSRV PASSWORD(password) NOEXPIRED KERB(KERBNAME("ldap_prefix/hostname")
ALTUSER LDAPSRV NOPASSWORD
```

Note that `ldap_prefix` must be either `ldap` or `LDAP`. It is recommended that `ldap` be used unless compatibility with earlier z/OS LDAP clients is needed. Also, the `hostname` needs to be the primary hostname for the system in DNS.

If the LDAP server is located on the same machine as the Key Distribution Center then a keytab file is not necessary to start the LDAP server. However the user ID that starts the server must have at least read access to IRR.RUSERMAP in the FACILITY class. This can be done by issuing the following RACF commands:

```
RDEFINE FACILITY IRR.RUSERMAP UACC(NONE)
PERMIT IRR.RUSERMAP CLASS(FACILITY) ID(LDAPSRV) ACCESS(READ)
SETROPTS RACLIST(FACILITY) REFRESH
```

If the LDAP server is not running on the same machine as the Kerberos KDC server, then a key table is required. The `kadmin` command is used to create the key table in a directory that can be accessed by the LDAP server. The name of the key table is then specified in the `krbKeytab` option in the LDAP server configuration file.

**Additional setup for generating audit records**

If you plan to generate LDAP SMF 83 audit records, the following RACF commands must be entered to set up the user ID that will run the LDAP server:

```
RDEFINE FACILITY IRR.RAUDITX UACC(NONE)
PERMIT IRR.RAUDITX CLASS(FACILITY) ID(LDAPSRV) ACCESS(READ)
SETROPTS RACLIST(FACILITY) REFRESH
```

**Additional setup for using securityLabel option**

If you plan on specifying `securityLabel` on in the global section of the LDAP server configuration file, the following RACF commands must be entered to set up the user ID that will run the LDAP server:

```
RDEFINE FACILITY BPX.POE UACC(NONE)
PERMIT BPX.POE CLASS(FACILITY) ID(LDAPSRV) ACCESS(READ)
SETROPTS RACLIST(FACILITY) REFRESH
```

See [z/OS UNIX System Services Planning](zos_unix_system_planning) for more information about setting up this profile. For other LDAP server considerations in a multilevel security environment, see [z/OS Planning for Multilevel Security and the Common Criteria](zos_planning_multilevel_security).
Protecting the environment for the LDAP server

The data set containing the LDAP server and the data sets containing any DLLs loaded by the LDAP server must be APF-authorized. The data sets can be authorized using SETPROG operator commands or in a PROGnn member in the system parameter library. Additionally, if program control is active on your system, the data sets containing the LDAP server and the data sets containing any DLLs loaded by the LDAP server must be program controlled. For example:

```
SETPROG APF,ADD,DSN=SYS1.SIEALNKE, VOL=valid
```

The following data sets need to be authorized and program controlled for use with the LDAP server (substitute the appropriate high-level data set name qualifier):

- SYS1.SIEALNKE
- SYS1.LINKLIB
- CEE.SCEERUN
- `DB2HLQ.SDSNLOAD`
- SYS1.CSSLIB

In addition, data sets used by plug-in extensions to the LDAP server must be similarly protected. See [IBM Tivoli Directory Server Plug-in Reference for z/OS](https://www.ibm.com/support/docview.wss?uid=swg27045671) for more information.
Chapter 7. Preparing WLM, backends, sysplex, SSL/TLS, and encryption

This topic discusses what you need to do to prepare WLM, backends, sysplex, SSL/TLS, and encryption.

<table>
<thead>
<tr>
<th>To run the LDAP server optimally:</th>
<th>See:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure WLM (Workload Manager)</td>
<td>Setting up for WLM (workload management)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>If you plan to use:</th>
<th>You must:</th>
<th>See:</th>
</tr>
</thead>
<tbody>
<tr>
<td>A backend, such as LDBM, TDBM, CDBM, GDBM, SDBM, or EXOP</td>
<td>Copy the configuration file.</td>
<td>Copying the configuration files</td>
</tr>
<tr>
<td>The sample server to set up an LDBM backend</td>
<td>Use the set of example files shipped with the code.</td>
<td>Creating a sample server with an LDBM backend</td>
</tr>
<tr>
<td>TDBM or GDBM (DB2-based) backend</td>
<td>Create the DB2 database and table spaces using SPUFI.</td>
<td>Creating the DB2 database and table spaces for TDBM or GDBM</td>
</tr>
<tr>
<td>TDBM backend</td>
<td>Set up your configuration file.</td>
<td>Setting up for TDBM</td>
</tr>
<tr>
<td>SDBM backend</td>
<td>Set up your configuration file.</td>
<td>Setting up for SDBM</td>
</tr>
<tr>
<td>LDBM backend</td>
<td>Set up your configuration file.</td>
<td>Setting up for LDBM</td>
</tr>
<tr>
<td>CDBM backend</td>
<td>Set up your configuration file.</td>
<td>Setting up for CDBM</td>
</tr>
<tr>
<td>GDBM backend</td>
<td>Set up your configuration file.</td>
<td>Setting up for GDBM</td>
</tr>
<tr>
<td>EXOP backend</td>
<td>Set up your configuration file.</td>
<td>Setting up for Policy Director extended operations</td>
</tr>
<tr>
<td>Sysplex</td>
<td>Enable sysplex support.</td>
<td>Setting up for sysplex</td>
</tr>
<tr>
<td>SSL/TLS</td>
<td>Enable SSL/TLS support.</td>
<td>Setting up for SSL/TLS</td>
</tr>
<tr>
<td>Encryption</td>
<td>Configure encryption.</td>
<td>Configuring for encryption</td>
</tr>
</tbody>
</table>

Setting up for WLM (workload management)

The LDAP server supports Workload Manager (WLM) to allow an installation to set performance goals for work within the LDAP server, based on the client IP address or the bound user’s distinguished name (DN) associated with LDAP requests. If a client application is spamming the LDAP server with numerous LDAP requests, a lower priority can be assigned to those requests. This support can prevent the LDAP server’s resources from being consumed by spamming LDAP client applications.

The wlmExcept configuration option can be used to specify the client’s IP address, the bound user’s distinguished name (DN), or both to route those requests to any configured WLM transaction name (TN). The wlmExcept configuration option can be specified multiple times within the LDAP server configuration file to allow multiple client IP addresses or bound users’ DNs to be associated with the same or different WLM transaction name. The order of the wlmExcept configuration options in the LDAP server configuration file determines the order the LDAP server uses to match incoming client requests and route them to the WLM transaction name. See page 108 for more information about the wlmExcept configuration option.

The WLMEXCEPT operator modify command can be used to change the routing of incoming client requests to new or different WLM transaction names while the server is running. The WLMEXCEPT operator modify command can be used to associate a search pattern in the cn=operations,cn=monitor
entry to a WLM transaction name. Each time the `WLMEXCEPT` operator modify command is issued, the new mappings are added before any of the configured `wlmeExcept` configuration options or previously issued `WLMEXCEPT` operator modify commands. See [LDAP server operator commands](#) for more information about the `WLMEXCEPT` operator modify command.

During LDAP server initialization, a WLM enclave with a `GENERAL` transaction name is automatically created. WLM enclaves are created by the LDAP server for each configured `wlmeExcept` configuration option. Based on how the WLM enclaves are configured, the LDAP server runs under the highest priority enclave in WLM.

It is strongly recommended that any WLM transaction names specified on `wlmeExcept` configuration options or on the `WLMEXCEPT` modify operator commands be mapped to a WLM service or report class. If a WLM transaction name is not mapped to a service or report class, a discretionary classification is used which causes those LDAP operations to receive a low work priority. This can cause performance problems in the LDAP server since these WLM transaction names are not properly classified. See [z/OS MVS Planning: Workload Management](#) for more information about configuring WLM.

See [Workload manager (WLM)](#) for more information about using LDAP with WLM.

### Copying the configuration files

The configuration files need to be copied from the directory in which they are installed, `/usr/lpp/ldap/etc`, to the directory where they are used, `/etc/ldap`. Do not modify these files in the installation directory because any service to the files will overwrite the modifications. Instead, modify them in `/etc/ldap`. The following commands copy the configuration files:

```bash
cp /usr/lpp/ldap/etc/ds.conf /etc/ldap/.
cp /usr/lpp/ldap/etc/ds.envvars /etc/ldap/.
```

### Creating a sample server with an LDBM backend

There is a set of example files shipped in `/usr/lpp/ldap/examples/sample_server` that can be used to understand how to configure and run the LDAP server using an LDBM backend. The `ds.README` provides step-by-step instructions for getting an LDAP server configured and started quickly. The following list shows the files shipped in that directory.

- `ds.README` (Installation information for sample server)
- `sample.ldif` (Sample directory entries for sample server)

### Creating the DB2 database and table spaces for TDBM or GDBM

When using TDBM or DB2-based GDBM, the LDAP server DB2 database must be created by running a SPUFI (SQL Processor Using File Input) script from DB2 Interactive (DB2I). DB2I is a DB2 facility that provides for the running of SQL statements, DB2 (operator) commands, and utility invocation. For details on how to use DB2I and SPUFI, see [DB2 Application Programming and SQL Guide](#). A sample DB2 SPUFI script to create the LDAP server DB2 database is provided. The same script is used for both TDBM and GDBM. To use it, do the following:

1. **Copy the SPUFI script over to your SPUFI input data set.**
   
   The SPUFI script for creating the database, table spaces, tables, and indexes can be found in `GLDHLQ.SGLDSAMP(DSTDBMDB)`. (`GLDHLQ` refers to the high-level qualifier that was used to install the LDAP server data sets.)

2. **Determine values for SPUFI script.**
   
   In order to create the DB2 database and table spaces for TDBM or GDBM, you must first decide on certain values within the SPUFI file, as shown in Table 5 (Table 4 lists variables that are used in more than one file or configuration step. Be sure to specify the same values where necessary.) The SPUFI script provides specific instructions and information to help you determine the values to use in the table. The `DSTDBMDB SPUFI file` shows an example of the file to edit and run in the SPUFI facility.
### Table 5. TDBM value definitions for DSTDBMDB

<table>
<thead>
<tr>
<th>Attribute script</th>
<th>Suggested value</th>
<th>Variable name in SPUFI script</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database name</td>
<td>GLDBDB*</td>
<td>-DB_NAME-</td>
</tr>
<tr>
<td>Database owner</td>
<td>GLDSRV*</td>
<td>-DB_USERID-</td>
</tr>
<tr>
<td>User ID running the LDAP server</td>
<td>GLDSRV</td>
<td>-LDAP_USERID-</td>
</tr>
<tr>
<td>CLI plan name</td>
<td>DSNACLI</td>
<td>-DB_PLAN-</td>
</tr>
<tr>
<td>Entry table space name</td>
<td>ENTRYTS</td>
<td>-ENTRYTS-</td>
</tr>
<tr>
<td>Buffer pool name for the LDAP entry table space</td>
<td>BP0</td>
<td>-ENTRYTS_BP0-</td>
</tr>
<tr>
<td>Long entry table space name</td>
<td>LENTRYTS</td>
<td>-ENTRYTS-</td>
</tr>
<tr>
<td>Buffer pool name for the LDAP long entry</td>
<td>BP0</td>
<td>-ENTRYTS_BP0-</td>
</tr>
<tr>
<td>Long attribute table space name</td>
<td>LATTRTS</td>
<td>-LATTRTS-</td>
</tr>
<tr>
<td>Buffer pool name for the LDAP long attribute</td>
<td>BP0</td>
<td>-LATTRTS_BP0-</td>
</tr>
<tr>
<td>Miscellaneous table space name</td>
<td>MISCTS</td>
<td>-MISCTS-</td>
</tr>
<tr>
<td>Buffer pool name for the LDAP miscellaneous attribute</td>
<td>BP0</td>
<td>-MISCTS_BP0-</td>
</tr>
<tr>
<td>Search table space name</td>
<td>SEARCHTS</td>
<td>-SEARCHTS-</td>
</tr>
<tr>
<td>Buffer pool name for the LDAP search table</td>
<td>BP0</td>
<td>-SEARCHTS_BP0-</td>
</tr>
<tr>
<td>Replica table space name</td>
<td>REPTS</td>
<td>-REPTS-</td>
</tr>
<tr>
<td>Buffer pool name for the LDAP replica attribute</td>
<td>BP0</td>
<td>-REPTS_BP0-</td>
</tr>
<tr>
<td>Descendants table space name</td>
<td>DESCTS</td>
<td>-DESCTS-</td>
</tr>
<tr>
<td>Buffer pool name for the LDAP descendants attribute</td>
<td>BP0</td>
<td>-DESCTS_BP0-</td>
</tr>
<tr>
<td>Storage group</td>
<td>SYSDEFLT</td>
<td>-SYSDEFLT-</td>
</tr>
<tr>
<td>Search column truncation size (VALUE in DIR_SEARCH)</td>
<td>32</td>
<td>-SEARCH_TRUNC_SIZE-</td>
</tr>
<tr>
<td>DN truncation size (DN_TRUNC in DIR_ENTRY)</td>
<td>32 for GDBM 64 for TDBM</td>
<td>-ENTRY_DN_TRUNC_SIZE-</td>
</tr>
<tr>
<td>Maximum size of a DN (DN in DIR_ENTRY)</td>
<td>512</td>
<td>-ENTRY_DN_SIZE-</td>
</tr>
</tbody>
</table>

**Note:** * This value must be unique for each database you are creating.

3. **Modify the script.**

   Use the values from Table 5 to modify the script. You must have a unique database name and owner for each database you are creating.

   It is recommended, but not required, that the GDBM backend table spaces be defined to DB2 with row level locking. You can do this by adding LOCKSIZE ROW to each CREATE TABLESPACE statement in the SPUFI script to create a GDBM database:

   ```sql
   CREATE TABLESPACE ttt IN yyy
   USING STOGROUP SYSDEFLT
   PRIQTY 14400
   SEQQTY 7200
   LOCKSIZE ROW
   BUFFERPOOL BP0;
   ```

   where *ttt* is a table space name used in the GDBM SPUFI script and *yyy* is the database name used in the script.
An existing GDBM database can be updated for row level locking by running the following statements using SPUFI (DB2 Interactive). Change the table space names if you used different names when creating the database.

```
ALTER TABLESPACE yyy.ENTRYTS LOCKSIZE ROW;
ALTER TABLESPACE yyy.ENTRYTS LOCKSIZE ROW;
ALTER TABLESPACE yyy.LATTRTS LOCKSIZE ROW;
ALTER TABLESPACE yyy.SECRTS LOCKSIZE ROW;
ALTER TABLESPACE yyy.DESCRTS LOCKSIZE ROW;
ALTER TABLESPACE yyy.MISCTS LOCKSIZE ROW;
ALTER TABLESPACE yyy.REPTS LOCKSIZE ROW;
```

where `yyy` is the database name used in the GDBM SPUFI script.

4. **Run the script from DB2 SPUFI.**

   Use the DB2 SPUFI (SQL Processor Using File Input) facility to create the database and table spaces. Be sure to run the script that was copied and modified in the previous steps under a user ID with DB2 **SYSADM** authority. When the script completes running, scan the output data set to ensure that it ran successfully.

5. **Grant appropriate DB2 resource authorizations.**

   In order to run the LDAP server, `ds2ldif`, and `ldif2ds`, certain minimum DB2 resource authorizations must be granted to the user ID or user IDs that will be running these programs. Following are the suggested minimums which should be granted to those user IDs, where `xxx` is the user ID running the LDAP server, `ds2ldif`, or `ldif2ds`, `yyy` is the database name identified in the SPUFI file and `zzz` is the CLI plan name as specified in your DB2 CLI initialization file. The grant for execute only has to be done once, for the LDAP server. The grant for dbadm has to be done for each TDBM or GDBM backend. Run the following statements using SPUFI (DB2 Interactive):

   ```
   grant execute on plan zzz to xxx;
   grant dbadm on database yyy to xxx;
   ```

   These privileges might be granted by any user ID with **SYSADM** authority.

   The LDAP server, `ds2ldif`, and `ldif2ds` require `SELECT` access to SYSIBM tables in DB2. If `SELECT` access to these tables is tightly controlled in your DB2 installation, it might be necessary to grant this access to the user ID under which the LDAP server, `ds2ldif`, or `ldif2ds` runs by performing the following operations (either using SPUFI or another means of issuing SQL commands). These grants are only done once, for the LDAP server.

   ```
   grant select on sysibm.syscolumns to xxx;
   grant select on sysibm.syscoldist to xxx;
   grant select on sysibm.systables to xxx;
   grant select on sysibm.systablepart to xxx;
   grant select on sysibm.syskeys to xxx;
   ```

   where `xxx` is the user ID under which the LDAP server runs. If this authority is not granted to the user ID under which the LDAP server runs, the LDAP server fails during start with an SQL -551 return code.

**Partitioning DB2 tables for TDBM**

**Note:** Partitioning is considered only for a TDBM database. Do not partition a GDBM database.

If you are creating a large TDBM directory, you should partition the "entry table space" and "search table space" to improve performance and ease maintainability of the database. The following information identifies the partitioning indexes and values to use when partitioning these table spaces.

<table>
<thead>
<tr>
<th>Table space</th>
<th>Table name</th>
<th>Partitioning index</th>
<th>Partitioning column</th>
<th>Value range of column</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search tablespace</td>
<td>DIR_SEARCH</td>
<td>DIR_SEARCHX2</td>
<td>EID</td>
<td>0-9999999999999999</td>
</tr>
<tr>
<td>Entry tablespace</td>
<td>DIR_ENTRY</td>
<td>DIR_ENTRYX0</td>
<td>EID</td>
<td>0-9999999999999999</td>
</tr>
</tbody>
</table>
The EID value generated by the TDBM backend is a 15 digit decimal number between 1 and 999999999999999. To determine the maximum value to assign to each partition, calculate the maximum value's first 4 digits and concatenate it with eleven 9s (999999999999999). The formula for calculating the first 4 digits is:

\[
\text{first 4 digits of partition_max_value} = \left(\frac{10000}{\text{number of partitions}}\right) \times \text{partition_number} - 1
\]

where partition_number starts with 1

You can also partition the following additional table spaces in TDBM using the EID range as the partitioning value.

### Table 7. TDBM table space partitioning using EID range

<table>
<thead>
<tr>
<th>Table space</th>
<th>Table name</th>
<th>Partitioning index</th>
<th>Partitioning column</th>
<th>Value range of column</th>
</tr>
</thead>
<tbody>
<tr>
<td>Descendants table space</td>
<td>DIR_DESC</td>
<td>DIR_DESCX1</td>
<td>DEID</td>
<td>0-999999999999999</td>
</tr>
<tr>
<td>Long attribute table space</td>
<td>DIR_LONGATTR</td>
<td>DIR_LONGATTRX1</td>
<td>EID</td>
<td>0-999999999999999</td>
</tr>
<tr>
<td>Long entry table space</td>
<td>DIR_LONGENTRY</td>
<td>DIR_LONGENTRYX1</td>
<td>EID</td>
<td>0-999999999999999</td>
</tr>
</tbody>
</table>

### Partitioning example

If you want to partition your directory into 10 partitions, the maximum value for each partition is:

<table>
<thead>
<tr>
<th>Partition number</th>
<th>Partition maximum value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>099999999999999</td>
</tr>
<tr>
<td>2</td>
<td>199999999999999</td>
</tr>
<tr>
<td>3</td>
<td>299999999999999</td>
</tr>
<tr>
<td>4</td>
<td>399999999999999</td>
</tr>
<tr>
<td>5</td>
<td>499999999999999</td>
</tr>
<tr>
<td>6</td>
<td>599999999999999</td>
</tr>
<tr>
<td>7</td>
<td>699999999999999</td>
</tr>
<tr>
<td>8</td>
<td>799999999999999</td>
</tr>
<tr>
<td>9</td>
<td>899999999999999</td>
</tr>
<tr>
<td>10</td>
<td>999999999999999</td>
</tr>
</tbody>
</table>

If you want to partition your directory into 4 partitions, the maximum value for each partition is:

<table>
<thead>
<tr>
<th>Partition number</th>
<th>Partition maximum value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>249999999999999</td>
</tr>
<tr>
<td>2</td>
<td>499999999999999</td>
</tr>
<tr>
<td>3</td>
<td>749999999999999</td>
</tr>
<tr>
<td>4</td>
<td>999999999999999</td>
</tr>
</tbody>
</table>
Setting up for TDBM

The LDAP server provides a backend to store directory information in a DB2 database. TDBM is a
general-purpose backend that can store any type of directory information.

In order to configure your LDAP server to run with the TDBM backend of the LDAP server:

- If you have not already done this, copy the configuration files from the `/usr/lpp/ldap/etc` directory to the
  `/etc/ldap` directory (see [Copying the configuration files]).
- You need to use the following lines in your `ds.conf` file:

  ```
  database tdbm GLDBTD31
dbuserid userid
  suffix "your_suffix"
  ```

  where `userid` is the TDBM database owner and `your_suffix` is any valid DN (distinguished name). Be
  sure to provide a meaningful value for the suffix.

Notes®:
1. Multiple TDBM backends can be configured in any given LDAP server, but, they must each use a
different DB2 database. The `dbuserid` value must be different than the ones used by any other TDBM
backends or DB2-based GDBM backend. Similarly, the database name used when creating the DB2
tables for the TDBM backend must be unique.
2. The attributes and object classes used by TDBM depend on your usage of TDBM. It is possible that
you need to add schema to the LDAP server schema. See [Setting up the schema for LDBM, TDBM,
and CDBM - new users] for more information about adding schema to the LDAP server.
3. A TDBM backend cannot be run in 64-bit mode. The LDAP server must be started in 31-bit mode.
4. If you specify the `dsnaoini` or `serverName` configuration option for TDBM, then you must also specify
the option for each TDBM backend and DB2-based GDBM backend and the value must be the same.

Copying a TDBM database

If you want to copy an existing TDBM database to a new one, you should use `ds2ldif` to unload the
existing TDBM database and `ldif2ds` to load the results into the new TDBM database.

If the new TDBM database is not on the same LDAP server as the existing one, the LDAP server schema
for the target LDAP server needs to contain all the attributes and object classes used by the TDBM entries
before they can be loaded into the new TDBM database. You can use `ds2ldif` or `ldapsearch` to unload the
LDAP server schema from the source LDAP server and then load the schema LDIF file into the target
LDAP server using `ldapmodify`.

See [Chapter 11, “Running and using the LDAP server utilities”] for more information about `ds2ldif` and
`ldif2ds`.

You can also use DB2 utilities to unload the existing database and load the new database.

Note: There are several table spaces, such as the miscellaneous table space (MISCTS) and the replica
table space (REPTS), which contain multiple tables. You must use the appropriate options on the
DB2 utilities when processing these table spaces.

Setting up for SDBM

The LDAP server can provide remote LDAP access to the user, group, connection, and general resource
profile information stored in RACF. It also supports setting RACF options that affect classes. See
[Chapter 16, “Accessing RACF information”] for details about how you can use this RACF information.

When creating change log records for changes to RACF data, SDBM is required.
In order to configure your LDAP server to run with the SDBM backend of the LDAP server:

- If you have not already done this, copy the configuration files from the /usr/lpp/ldap/etc directory to the /etc/ldap directory (see Copying the configuration files).
- You need to use the following lines in your ds.conf file:

  ```
  database sdbm GLDBSD31/GLDBSD64
  suffix "your_suffix"
  ```
  
  where `your_suffix` is any valid DN (distinguished name). Be sure to provide a meaningful value for the suffix. Note that it is no longer required that the `sysplex` attribute be present in the suffix. For example, a valid suffix line is:

  ```
  suffix "cn=RACFA,o=IBM,c=US"
  ```

**Notes:**

1. Only one SDBM backend can be defined in any given LDAP server.
2. The attributes and object classes used by SDBM are always in the LDAP server schema, with the exception of any attributes needed for RACF custom fields.
3. The `enableResources` configuration option must be specified in your ds.conf file if you intend to display or manage RACF resource profiles and class options. This configuration option is also required if you want to create change log entries for changes to RACF resource profiles. See "Configuration file options" on page 75 for more information.

---

**Setting up for LDBM**

The LDAP server provides a file-based backend to store directory information in a UNIX System Services file system. LDBM is a general-purpose backend that can store any type of directory information.

The amount of space needed to store an LDBM backend in a z/OS UNIX System Services file system is approximately four to six times the size of the expected input LDIF data. Generally, the space required to hold the LDBM backend data is two to three times the size of the expected input LDIF data. However, during the LDBM commit process each of the LDBM backend files is copied, therefore, resulting in occasionally needing twice the amount of file system space.

LDBM keeps its directory in storage in the LDAP address space while the LDAP server is running. See LDBM performance considerations for more information about LDBM storage usage.

When the LDAP server starts for the first time with a new LDBM backend configured, the server automatically creates the directories specified in the `databaseDirectory` server configuration option (or in /var/ldap/ldb if the configuration option is not specified). When the directories are created, the LDAP server’s userid is the owner of these directories. The permissions on these directories grant read, write, and execute access to the LDAP server’s userid. The group that the LDAP server’s userid belongs to is granted read access to the directories. As part of the LDBM backend initialization process, the server creates LDBM-x.db files (where x is a number such as 1, 2, 3, and so on) for each suffix in the LDBM backend section, along with an LDBM.ckpt file. These files are created with the LDAP server’s userid as the owner. The default permissions on these files grant read and write access to the LDAP server’s userid while the group to which the LDAP server’s userid belongs is granted read access.

If the default LDBM backend file or directory permissions or ownership are not sufficient for your needs, they can be changed manually by issuing `chmod` and `chown` commands. The LDAP server retains any manual changes in the file or directory permissions or ownership. For additional information about the `chmod` and `chown` commands, see z/OS UNIX System Services Command Reference.

In order to configure your LDAP server to run with the LDBM backend of the LDAP server:

- If you have not already done this, copy the configuration files from the /usr/lpp/ldap/etc directory to the /etc/ldap directory (see Copying the configuration files).
You need to use the following lines in your `ds.conf` file:

database ldbm GLDBLD31/GLDBLD64
suffix "your_suffix"

where `your_suffix` is any valid DN (distinguished name). Be sure to provide a meaningful value for the suffix.

**Notes:**

1. Multiple LDBM backends can be configured in an LDAP server, but each must use a different file
directory for storing its entries. If you are configuring multiple LDBM backends or do not want an
LDBM backend to store its entries in the default file directory, then add the `databaseDirectory`
option to the LDBM section of the configuration file. The `databaseDirectory` value must be different
than the ones used by any other LDBM backends, file-based GDBM backend, or CDBM backend.
The LDAP server must have read-write access to the file directory.

2. The attributes and object classes used by LDBM depend on your usage of LDBM. It is possible that
you need to add add schema to the LDAP server schema. See [Setting up the schema for LDBM, TDBM, and CDBM - new users](#) for more information about adding schema to the LDAP server.

### Copying an LDBM backend

If you want to copy an existing LDBM backend to a new one, you should use the `ds2ldif` utility to unload
the existing LDBM backend. Then use the `ldapadd` utility to load the new LDBM backend. If it is desired to
retain the existing `ibm-entryuuid` attribute values for the entries, the LDAP server must be put into
maintenance mode. See [Basic replication maintenance mode](#) for more information.

If the new LDBM backend is not on the same LDAP server as the existing one, the LDAP server schema
for the target LDAP server needs to contain all the attributes and object classes used by the LDBM entries
before they can be loaded into the new LDBM backend. You can use `ds2ldif` or `ldapsearch` to unload the
LDAP server schema from the source LDAP server and then load the schema LDIF file into the target
LDAP server using `ldapmodify`.

### Setting up for CDBM

The LDAP server provides the CDBM backend to store configuration information, for example, for the
advanced replication support. CDBM is file-based, storing its directory information in a UNIX System
Services file system.

The amount of space needed to store a CDBM backend in a z/OS UNIX System Services file system is
approximately four to six times the size of the expected input LDIF data. Generally, the space required to
hold the CDBM backend data is two to three times the size of the expected input LDIF data. However,
during the CDBM commit process each of the CDBM backend files is copied, therefore, resulting in
occasionally needing twice the amount of file system space.

CDBM keeps its directory in storage in the LDAP address space while the LDAP server is running.

When the LDAP server starts for the first time with the CDBM backend configured, the server
automatically creates the directories specified in the `databaseDirectory` server configuration option (or in
the directory specified by the `schemaPath` configuration option or in `/var/ldap/schema` if `schemaPath` is
not specified). When the directories are created, the LDAP server’s userid is the owner of these
directories. The permissions on these directories grant read, write, and execute access to the LDAP
server’s userid. The group that the LDAP server’s userid belongs to is granted read access to the
directories. As part of the CDBM backend initialization process, the server creates `LDBM-1.db` and
`LDBM-2.db` files for the `cn=configuration` and `cn=ibmpolicies` suffixes, along with an `LDBM.ckpt` file.
These files are created with the LDAP server’s userid as the owner. The default permissions on these files
grant read and write access to the LDAP server’s userid while the group to which the LDAP server’s
userid belongs is granted read access.
If the default CDBM backend file or directory permissions or ownership are not sufficient for your needs, they can be changed manually by issuing `chmod` and `chown` commands. The LDAP server retains any manual changes in the file or directory permissions or ownership. For additional information about the `chmod` and `chown` commands, see the [z/OS UNIX System Services Command Reference](#).

In order to configure your LDAP server to run with the CDBM backend of the LDAP server:

- If you have not already done this, copy the configuration files from the `/usr/lpp/ldap/etc` directory to the `/etc/ldap` directory (see Copying the configuration files).
- You need to use the following line to your `ds.conf` file:

  ```
  database cdbm GLDBCD31/GLDBCD64
  ```

Notes:

1. Only one CDBM backend can be configured in an LDAP server. If the `databaseDirectory` option is not specified in the CDBM backend section, the CDBM backend stores its entries in the directory specified by the `schemaPath` configuration option. If the `schemaPath` configuration option is not specified, the CDBM backend data is stored in the default `schemaPath` directory which is `/var/ldap/schema`. The `databaseDirectory` value must be different than the ones used by any other LDBM backends or file-based GDBM backend. The LDAP server must have read-write access to the file directory.

2. Depending on the types of entries you intend to add to the CDBM backend, you may need to add schema to the LDAP server schema for the attributes and object classes that are to be used. See Setting up the schema for LDBM, TDBM, and CDBM - new users for information about adding schema to the LDAP server.

### Setting up for GDBM

The LDAP server can provide a change log containing information about changes to:

- RACF users, groups, user-group connections, and general resource profiles
- TDBM, LDBM, and CDBM entries
- LDAP server schema entry

GDBM can be configured to store change log entries either in a z/OS UNIX System Services file system or in DB2.

Notes:

1. Only one GDBM backend can be defined in any given LDAP server.
2. The attributes and object classes used by GDBM are always in the LDAP server schema.
3. See Chapter 26, “Change logging” for additional configuration options that can be specified.
4. If you intend to create change log entries for changes to RACF data, you must also configure an SDBM backend and enable the LDAP Program Callable support. See Setting up for SDBM and Additional required configuration for more information.

### Configuring file-based GDBM

The amount of space needed to store a GDBM (file-based) backend in a z/OS UNIX System Services file system depends on how many change log entries are going to be stored and the size of the change log entries. The number of change log entries can be controlled using the `changeLogMaxEntries` and `changeLogMaxAge` options in the GDBM section of the LDAP server configuration file. The size of a change log entry is related to the size of the LDIF when adding or modifying a TDBM, LDBM, or CDBM entry, because this LDIF is inserted into the change log entry. Generally, the space required to hold the GDBM backend data is:

6 X (maximum number of GDBM entries) X (largest add or modify LDIF + 1000)

This includes the extra space needed to copy the database files during GDBM commit processing.
When the LDAP server starts for the first time with a new GDBM (file-based) backend configured, the server automatically creates the directories specified in the `databaseDirectory` server configuration option (or in `/var/ldap/gdbm` if the configuration option is not specified). When the directories are created, the LDAP server’s userid is the owner of these directories. The permissions on these directories grant read, write, and execute access to the LDAP server’s userid. The group that the LDAP server’s userid belongs to is granted read access to the directories. As part of the GDBM (file-based) backend initialization process, the server creates an `LDBM-1.db` file for the changelog backend, along with an `LDBM.ckpt` file. These files are created with the LDAP server’s userid as the owner. The default permissions on these files grant read and write access to the LDAP server’s userid while the group to which the LDAP server’s userid belongs is granted read access.

If the default GDBM (file-based) backend file or directory permissions or ownership are not sufficient for your needs, they can be changed manually by issuing `chmod` and `chown` commands. The LDAP server retains any manual changes in the file or directory permissions or ownership. For additional information about the `chmod` and `chown` commands, see `z/OS UNIX System Services Command Reference`.

In order to configure your LDAP server to run with the GDBM (file-based) backend of the LDAP server:

- If you have not already done this, copy the configuration files from the `/usr/lpp/ldap/etc` directory to the `/etc/ldap` directory (see Copying the configuration files).
- You need to use the following line in your `ds.conf` file:

  ```
  database gdbm GLDBG31/GLDBG64
  ```

The default file directory used by the file-based GDBM backend to store its entries is `/var/ldap/gdbm`. If you do not want GDBM to use that file directory, then add the `databaseDirectory` option to the GDBM section of the configuration file. The file directory must be different than the ones used by any LDBM backends or the CDBM backend. The LDAP server must have read-write access to the file directory.

### Configuring DB2-based GDBM

In order to configure your LDAP server to run with the GDBM (DB2-based) backend of the LDAP server:

- If you have not already done this, copy the configuration files from the `/usr/lpp/ldap/etc` directory to the `/etc/ldap` directory (see Copying the configuration files).
- You need to use the following lines in your `ds.conf` file:

  ```
  database gdbm GLDBG31
  dbuserid userid
  ```

  where `userid` is the GDBM database owner.

**Notes:**

1. A DB2-based GDBM backend cannot be run in 64-bit mode. The LDAP server must be started in 31-bit mode.
2. The `dbuserid` value must be different than the value used by any TDBM backends. If you specify the `dsnaoini` or `serverName` configuration option for GDBM, then you must also specify the option for each TDBM backend and the value must be the same.

### Setting up for Policy Director extended operations

The LDAP server provides the EXOP backend to support extended operations that retrieve Policy Directory data. See Chapter 20, “Using extended operations to access Policy Director data” for details about using this extended operations support. The EXOP backend is not needed when using any other extended operations, such as those used in advanced replication.

To configure your LDAP server to run with the EXOP backend:
• If you have not already done this, copy the configuration files from the `/usr/lpp/ldap/etc` directory to the `/etc/ldap` directory (see Copying the configuration files).

• You need to use the following line in the global section of the `ds.conf` file to enable Program Call support:

  ```
  listen ldap://:pc
  ```

• You need to also add the following line after the global section of the `ds.conf` file:

  ```
  database exop GLDXP031/GLDXP064
  ```

### Setting up for sysplex

An LDAP server running in a sysplex environment supports multiple instances of the same server within a cross-system coupling facility group. Each server instance runs on a separate system but shares the same schema and optionally backend databases. This provides improved workload management when the TCP/IP sysplex distributor is used to route incoming connections to available LDAP servers within the sysplex. Each server in the same group must be identical for the backends being shared to the other servers in the group, therefore, any server in the group can be used to satisfy a request. Authentication and authorization definitions are the same for all servers in the same group so that a client authenticated by one server in the group is assumed to be authenticated to all of the servers in the group.

In order to configure your LDAP server to run in a sysplex:

• If you have not already done this, copy the configuration files from the `/usr/lpp/ldap/etc` directory to the `/etc/ldap` directory (see Copying the configuration files). It is strongly recommended when the same configuration is wanted on all LDAP servers within the sysplex group that they physically share the same configuration file. This eliminates the possible incorrect configuration of servers that exist within the same cross system coupling facility (XCF) group.

• Add the following line to the global section of your `ds.conf` file:

  ```
  serverSysplexGroup group_name
  ```

  where `group_name` specifies the cross-system coupling facility (XCF) group. All of the LDAP servers sharing the same schema and backend databases must be a member of the same XCF group. LDAP sysplex support is activated when the `serverSysplexGroup` configuration option is specified.

• Each LDAP server in the XCF group must specify the same value for the `schemaPath` configuration option and must have read/write access to the specified directory or to `/var/ldap/schema` if the option is not specified. The schema directory used must exist within a shared z/OS UNIX System Services file system and must be accessible to all servers in the XCF group. See [z/OS UNIX System Services Planning](https://www.ibm.com/support/docview.wss?uid=swg27012605) for information about setting up a shared z/OS UNIX System Services file system.

• For each LDBM, TDBM, CDBM, and GDBM backend that is to be shared, add the following line to the backend section of your `ds.conf` file:

  ```
  multiserver on
  ```

  **Note:** The `multiserver` configuration option is not supported by the SDBM backend. Sysplex support for the SDBM backend is provided by RACF and not the LDAP server. See [z/OS Security Server RACF System Programmer's Guide](https://www.ibm.com/support/docview.wss?uid=swg27012677) for information about setting up a shared RACF database.

Each LDBM, TDBM, CDBM, and GDBM backend can either be shared within the XCF group or not. To share a backend, specify `multiserver on` in the backend section in the configuration file of each LDAP server. If `multiserver off` is specified or if the `multiserver` option is not specified, then the backend is not shared and changes to the backend are not reflected in the other servers on the sysplex, even if they contain the same suffix. If GDBM or CDBM backends are configured, then all LDBM, TDBM, CDBM, and GDBM backends must be shared or they all must be not shared. If GDBM and CDBM backends are not configured, then some LDBM and TDBM backends can be shared while others are not shared.

When an LDBM, TDBM, CDBM, or GDBM backend is shared, the `name` parameter must be specified on the `database` option for that backend and must be the same in the configuration file for each LDAP server.
The configuration file on each LDAP server that is sharing a backend must specify the same suffixes for that backend. Do not specify these suffixes for a different backend. If sharing an LDBM, CDBM, or file-based GDBM database, each LDAP server must have read/write access to the z/OS UNIX System Services directory containing the database files. When an LDBM, CDBM, or file-based GDBM backend is shared, the databaseDirectory configuration option must have the same value, exist within a shared z/OS UNIX System Services file system, and must be accessible to all servers in the XCF group. Each server must have read/write access to the specified directory or to /var/ldap/ldbm (LDBM), /var/ldap/schema (CDBM - defaults to the schemaPath configuration option setting), /var/ldap/gdbm (GDBM) if the databaseDirectory configuration option is not specified. See z/OS UNIX System Services Planning for information about setting up a shared z/OS UNIX System Services file system.

Modification requests for the LDAP server schema or for a shared backend can be directed to any LDAP server in the sysplex group. The modified schema or directory is seen by all servers within the group. Also, when performing a persistent search of a shared backend, only one persistent search request needs to be made to one of the LDAP servers in the sysplex. The sysplex support results in all the LDAP servers participating in the persistent search. The exception to this is if the target of the search is a TDBM backend when the serverCompatLevel configuration option is less than 4. In this case, an identical persistent search request must be issued to each LDAP server in the sysplex. See page 100 for more information about serverCompatLevel.

**Setting up for SSL/TLS**

The LDAP server contains the ability to protect LDAP access with Secure Sockets Layer (SSL) and Transport Layer Security (TLS). There are two types of connections that support secure communication:

- An SSL/TLS only secure connection. This connection requires that the first communication between the client and the server be the handshake that negotiates the secure communication. From that point on only secure communication can occur on the connection.

- A bimodal connection that supports secure and non-secure communication. The client is expected to begin communication in a non-secure mode. At some time during communication, the client might change to secure communication by sending a StartTLS extended operation after which the handshake to negotiate secure communication occurs followed by secure communication. The client might shutdown secure communication causing a StopTLS alert to be sent and the server will continue communication in a non-secure mode. At a later time, the client might restart secure communication by sending another StartTLS extended operation followed by the handshake.

Both types of connections require that SSL/TLS be configured for use by the LDAP server.

**Using SSL/TLS protected communications**

The Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols use public-key infrastructure (PKI) algorithms to establish and maintain an encrypted communications path between a client and server. In z/OS, the ability to set up and communicate over SSL/TLS protected communication links is provided by the LDAP server with a set of services provided in z/OS (the z/OS Cryptographic Services System SSL set of services).

In order for the LDAP client to communicate with an LDAP server over an SSL/TLS-protected TCP/IP socket connection, the LDAP server must transmit a certificate to the LDAP client and, optionally, the client can transmit its certificate to the LDAP server. The LDAP client and server must verify that the certificates they received are valid. After the LDAP client and server have determined the validity of the certificates provided to them, SSL/TLS-protected communication occurs between the LDAP client and server.

The LDAP client and server verify the certificates sent to them by using public-key digital signatures. The LDAP client and server take the certificates and compare the digital signature in the certificates with a signature that it computes based on having the public-key of the signer of the certificate. In order to do this, the LDAP client and server must have the public-key of the signer of the certificates. The LDAP client and server obtain this by reading a file that contains these public-keys. This file is called a key database.
A key database, RACF key ring, or PKCS #11 token contains the public-keys that are associated with signers of certificates. These public-keys are, in reality, contained in certificates themselves. Therefore, verifying one certificate requires the use of a different certificate, the signer’s certificate. In this fashion, a chain of certificates is established, with one certificate being verified by using another certificate and that certificate being verified by yet another certificate, and so on. A certificate, and its associated public key, can be defined as a root certificate. A root certificate is self-signed, meaning that the public-key contained in the certificate is used to sign the certificate. Using a root certificate implies that the user trusts the root certificate.

The key databases, RACF key rings, or PKCS #11 tokens used by the LDAP client and server must contain enough certificates in order to verify the certificates sent by the LDAP client and server during the start of the SSL/TLS connection. If either certificate is self-signed, then that certificate must be stored in the other’s key database. If the certificates are signed by some other certificate signer, then the signer’s certificate and any certificates that this certificate depends upon must be stored in the key databases. The key databases, RACF key rings, or PKCS #11 tokens used by the LDAP client and server must also contain the certificates that they will transmit to each other during the startup of the SSL/TLS-protected communications.

Creating and using key databases, key rings, or PKCS #11 tokens

The LDAP client and server use the System SSL functions provided in z/OS to set up SSL/TLS protected communications. The System SSL capability requires a key database, RACF key ring, or PKCS #11 token to be set up before SSL/TLS protected communications can begin.

The key database is a password protected file stored in the file system. This file is created and managed using a utility program provided with System SSL called gskkyman. Directions for using the gskkyman utility can be found in z/OS: System Secure Sockets Layer Programming. The key database file that is created must be accessible by the LDAP server.

The key ring is maintained by RACF. This object is created and managed using the RACF Digital Certificate command, RACDCERT. Directions for using the RACDCERT command can be found in z/OS Security Server RACF Command Language Reference.

The user ID under which the LDAP server runs must be authorized by RACF to use RACF key rings. To authorize the LDAP server, you can use the RACF commands in the following example:

RDEFINE FACILITY IRR.DIGTCERT.LIST UACC(NONE)
RDEFINE FACILITY IRR.DIGTCERT.LISTRING UACC(NONE)
PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(LDAPSrv) ACCESS(CONTROL)
PERMIT IRR.DIGTCERT.LIST CLASS(FACILITY) ID(LDAPSrv) ACCESS(CONTROL)

Remember to refresh RACF after doing the authorizations.

SETROPTS RACLIST(FACILITY) REFRESH

PKCS #11 tokens are stored and protected by ICSF. The gskkyman utility or the RACDCERT command can be used to create or modify PKCS #11 tokens. ICSF uses the CRYPTOZ SAF class to determine if the issuer of the gskkyman utility or the RACDCERT command is permitted to perform the operation against a z/OS PKCS #11 token. See z/OS Cryptographic Services System SSL Programming for more information about the gskkyman utility and z/OS Security Server RACF Command Language Reference for more information about the RACDCERT command.

The user ID associated with the LDAP server must be authorized by RACF to use the PKCS #11 token. To authorize the LDAP server, you can use the RACF commands in the following example (where NAME is the name of the PKCS #11 token).
Remember to refresh RACF after doing the authorizations.

SETROPTS RACLIST(CRYPTOZ) REFRESH

For testing purposes, the LDAP server can use a self-signed certificate. In this case, the certificate of the LDAP server must also be stored in the key database, RACF key ring, or PKCS #11 token of the LDAP client in order for SSL/TLS protected LDAP communications to work between the client and server.

If the SSL certificate that the LDAP server is going to use is not marked as the default certificate in the key database, RACF key ring, or the PKCS #11 token then the **sslCertificate** server configuration option must be specified in order to determine which SSL certificate to use.

### Obtaining a certificate

The LDAP server or client can obtain a certificate by contacting a certificate authority (CA) and requesting a certificate. Utilities to formulate a certificate request are provided by System SSL, **gskkyman**, and RACF, **RACDCERT**. This certificate request is typically passed to the CA by means of an electronic mail message or by an HTML form which is filled out using a web browser. When the CA verifies the information for the LDAP client or server, a certificate is returned to the requester, typically by an electronic mail message. The contents of the mail message are used to define the certificate in the key database, RACF key ring, or PKCS #11 token.

### Enabling SSL/TLS support

The following high-level steps are required to enable SSL/TLS support for LDAP. These steps assume you have already installed and configured the LDAP server and installed z/OS Cryptographic Services System SSL. The datasets containing the LDAP and SSL code must be APF authorized and available to the LDAP server.

1. Generate the LDAP server private key and server certificate and mark it as the default in the key database, RACF key ring, or PKCS #11 token or use its label on the **sslCertificate** option in the LDAP server configuration file.
2. Configure the LDAP server to listen for LDAP requests and configure the type of authentication wanted, server and optionally client authentication (see Setting up the security options for the LDAP server).
   - For a secure only socket, a **listen** configuration option or `-l` command-line parameter must be set up for the secure port.
   - For a bimodal socket, a **listen** configuration option or `-l` command-line parameter must be set up for the non-secure port.
3. Restart the LDAP server.

### Setting up the security options for the LDAP server

The following options for SSL/TLS can be set in the LDAP server configuration file. They are described in detail in Configuration file options.

- **listen**
- **sslAuth**
- **sslCertificate**
- **sslCipherSpecs**
- **sslKeyRingFile**
- **sslKeyRingFilePW**
- **sslKeyRingPWStashFile**
- **sslMapCertificate**
Notes:

1. The `replKeyRingFile` and `replKeyRingPW` options are no longer necessary or recognized by the LDAP server. These options should be removed from the configuration file.

2. The `security`, `port`, and `securePort` options have been deprecated by the `listen` option. For more information, see the `listen` option on page 86.

LDAP can be configured for SSL/TLS in two ways:

- For secure only communication, specify one or more `listen` options for secure communications in the following format:
  \l{daps://[IP_address | hostname][:portNumber]}

- For bimodal (non-secure/secure) communication, specify one or more `listen` options for non-secure communications in the following format:
  \l{ldap://[IP_address | hostname][:portNumber]}

See the `listen` option, on page 86 for more information.

`sslKeyRingFile` specifies the name of the key database, the RACF key ring, or the PKCS #11 token used by the LDAP server. This key database, RACF key ring, or PKCS #11 token is also used for SSL/TLS protected replication. Because the replicating server might be acting as both a replica server and an LDAP server, the replica server’s certificate (or CA certificate) must be contained in the replicating server’s key database file, RACF key ring, or PKCS #11 token. When using a PKCS #11 token, the `sslKeyRingFile` configuration option must be set such as this (where `NAME` is the name of the PKCS#11 token):

```
*TOKEN*/NAME
```

A key database requires a password. The password may be specified on the `sslKeyRingFilePW` option or the name of a password stash file may be specified on the `sslKeyRingPWStashFile` option in the LDAP server configuration file. Use of a stash file provides a method of specifying a password in a form that cannot be easily read by a human. The `gskkyman` utility provides a function to create the key database password stash file.

When a RACF key ring or PKCS #11 token is used instead of a key database, the `sslKeyRingFilePW` and `sslKeyRingPWStashFile` cannot be specified in the configuration file.

The LDAP server is configured to provide server and, optionally, client authentication. The `sslAuth` option controls this setting.

When using server authentication, by setting the `sslAuth` server configuration option to `serverAuth`, the LDAP server must have a digital certificate (based on the X.509 standard). This digital certificate is used to authenticate the LDAP server to the LDAP client application. The LDAP server supplies the client with the LDAP server’s X.509 certificate during the initial SSL handshake. If the client validates the server’s certificate, then a secure, encrypted communication channel is established between the LDAP server and the LDAP client application.

In addition, if the LDAP server is configured to use server and client authentication, by setting the `sslAuth` server configuration option to `serverClientAuth`, and the client sends a digital certificate on the initial SSL handshake, it must be validated by the LDAP server before the secure encrypted communication channel is established between them. The certificate is used to establish the bind identity of the client. This identity is also affected by the `sslMapCertificate` configuration option. See “Support of certificate bind” on page 63 for more information.

Note: If the LDAP server is configured for both server and client authentication but a client does not send a digital certificate, then the server acts as if configured for server authentication only. This compatibility with earlier versions of the LDAP server. In addition, System SSL can be configured to fail the SSL handshake if the client does not provide a certificate after the server has requested client authentication. System SSL provides an environment variable,
GSK_CLIENT_AUTH_NOCERT_A, which indicates that a client certificate must be passed to the server in order to prevent the SSL handshake from failing. See z/OS Cryptographic Services System SSL Programming for more information.

The `sslMapCertificate` option specifies whether the server maps a certificate used in a SASL EXTERNAL bind to the RACF user that is associated with the certificate.

The `sslCertificate` option indicates the label of the server certificate that is to be used. This option is needed if the default certificate has not been set in the key database, RACF key ring, or PKCS #11 token or if a certificate other than the default certificate wanted.

The `sslCipherSpecs` option specifies the cipher specifications that will be accepted from clients. If this option is not specified then all cipher specifications supported by the LDAP server will be used. Depending upon the level of System SSL support, the list of acceptable cipher specifications may be lowered because certain specifications might not be supported by System SSL for that level of the product.

The following table lists the LDAP cipher spec mask values, and the related decimal, hexadecimal, and keyword values. It also lists the System SSL cipher number.

<table>
<thead>
<tr>
<th>Cipher Type</th>
<th>Decimal Value</th>
<th>Hexadecimal Value</th>
<th>Decimal Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL</td>
<td>4294967295</td>
<td>xFFFFFFFF</td>
<td>4294967295</td>
<td>All cipher suites</td>
</tr>
<tr>
<td>ANY</td>
<td>4294967295</td>
<td>xFFFFFFFF</td>
<td>4294967295</td>
<td>All cipher suites</td>
</tr>
<tr>
<td>DES_SHA_EXPORT</td>
<td>512</td>
<td>x00000200 09</td>
<td>512</td>
<td>56-bit DES encryption with SHA-1 message authentication and RSA key exchange</td>
</tr>
<tr>
<td>DH_DSS_AES_128_SHA</td>
<td>1048576</td>
<td>x00100000 30</td>
<td>1048576</td>
<td>128-bit AES encryption with SHA-1 message authentication and fixed Diffie-Hellman key exchange signed with a DSS certificate</td>
</tr>
<tr>
<td>DH_DSS_AES_256_SHA</td>
<td>65536</td>
<td>x00010000 36</td>
<td>65536</td>
<td>256-bit AES encryption with SHA-1 message authentication and fixed Diffie-Hellman key exchange signed with a DSS certificate</td>
</tr>
<tr>
<td>DH_DSS_DES_SHA</td>
<td>128</td>
<td>x00000080 0C</td>
<td>128</td>
<td>56-bit DES encryption with SHA-1 message authentication and fixed Diffie-Hellman key exchange signed with a DSS certificate</td>
</tr>
<tr>
<td>DH_DSS_TRIPLE_DES_SHA</td>
<td>8</td>
<td>x00000008 0D</td>
<td>8</td>
<td>168-bit 3DES encryption with SHA-1 message authentication and fixed Diffie-Hellman key exchange signed with a DSS certificate</td>
</tr>
<tr>
<td>DH_RSA_AES_128_SHA</td>
<td>2097152</td>
<td>x00200000 31</td>
<td>2097152</td>
<td>128-bit AES encryption with SHA-1 message authentication and fixed Diffie-Hellman key exchange signed with an RSA certificate</td>
</tr>
<tr>
<td>Cipher</td>
<td>Decimal value</td>
<td>Hexadecimal value</td>
<td>SSL value</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------</td>
<td>-------------------</td>
<td>-----------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DH_RSA_AES_256_SHA</td>
<td>131072</td>
<td>x00020000</td>
<td>37</td>
<td>256-bit AES encryption with SHA-1 message authentication and fixed Diffie-Hellman key exchange signed with an RSA certificate</td>
</tr>
<tr>
<td>DH_RSA_DES_SHA</td>
<td>64</td>
<td>x00000040</td>
<td>0F</td>
<td>56-bit DES encryption with SHA-1 message authentication and fixed Diffie-Hellman key exchange signed with an RSA certificate</td>
</tr>
<tr>
<td>DH_RSA_TRIPLE_DES_SHA</td>
<td>4</td>
<td>x00000004</td>
<td>10</td>
<td>168-bit 3DES encryption with SHA-1 message authentication and fixed Diffie-Hellman key exchange signed with an RSA certificate</td>
</tr>
<tr>
<td>DHE_DSS_AES_128_SHA</td>
<td>4194304</td>
<td>x00400000</td>
<td>32</td>
<td>128-bit AES encryption with SHA-1 message authentication and ephemeral Diffie-Hellman key exchange signed with a DSS certificate</td>
</tr>
<tr>
<td>DHE_DSS_AES_256_SHA</td>
<td>262144</td>
<td>x00040000</td>
<td>38</td>
<td>256-bit AES encryption with SHA-1 message authentication and ephemeral Diffie-Hellman key exchange signed with a DSS certificate</td>
</tr>
<tr>
<td>DHE_DSS_DES_SHA</td>
<td>32</td>
<td>x00000020</td>
<td>12</td>
<td>56-bit DES encryption with SHA-1 message authentication and ephemeral Diffie-Hellman key exchange signed with a DSS certificate</td>
</tr>
<tr>
<td>DHE_DSS_TRIPLE_DES_SHA</td>
<td>2</td>
<td>x00000002</td>
<td>13</td>
<td>168-bit 3DES encryption with SHA-1 message authentication and ephemeral Diffie-Hellman key exchange signed with a DSS certificate</td>
</tr>
<tr>
<td>DHE_RSA_AES_128_SHA</td>
<td>8388608</td>
<td>x00800000</td>
<td>33</td>
<td>128-bit AES encryption with SHA-1 message authentication and ephemeral Diffie-Hellman key exchange signed with an RSA certificate</td>
</tr>
<tr>
<td>DHE_RSA_AES_256_SHA</td>
<td>524288</td>
<td>x00080000</td>
<td>39</td>
<td>256-bit AES encryption with SHA-1 message authentication and ephemeral Diffie-Hellman key exchange signed with an RSA certificate</td>
</tr>
<tr>
<td>DHE_RSA_DES_SHA</td>
<td>16</td>
<td>x00000010</td>
<td>15</td>
<td>56-bit DES encryption with SHA-1 message authentication and ephemeral Diffie-Hellman key exchange signed with an RSA certificate</td>
</tr>
</tbody>
</table>
Table 8. Supported ciphers (continued)

<table>
<thead>
<tr>
<th>Cipher</th>
<th>Decimal value</th>
<th>Hexadecimal value</th>
<th>SSL value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DHE_RSA_TRIPLE_DES_SHA</td>
<td>1</td>
<td>x00000001</td>
<td>16</td>
<td>168-bit 3DES encryption with SHA-1 message authentication and ephemeral Diffie-Hellman key exchange signed with an RSA certificate</td>
</tr>
<tr>
<td>RC2_MD5_EXPORT</td>
<td>4096</td>
<td>x00001000</td>
<td>06</td>
<td>40-bit RC2 encryption with MD5 message authentication and RSA key exchange</td>
</tr>
<tr>
<td>RC4_MD5_EXPORT</td>
<td>8192</td>
<td>x00002000</td>
<td>03</td>
<td>40-bit RC4 encryption with MD5 message authentication and RSA key exchange</td>
</tr>
<tr>
<td>RC4_MD5_US</td>
<td>2048</td>
<td>x00000800</td>
<td>04</td>
<td>128-bit RC4 encryption with MD5 message authentication and RSA key exchange</td>
</tr>
<tr>
<td>RC4_SHA_US</td>
<td>1024</td>
<td>x00000400</td>
<td>05</td>
<td>128-bit RC4 encryption with SHA-1 message authentication and RSA key exchange</td>
</tr>
<tr>
<td>RSA_AES_128_SHA</td>
<td>16384</td>
<td>x00004000</td>
<td>2F</td>
<td>128-bit AES encryption with SHA-1 message authentication and RSA key exchange</td>
</tr>
<tr>
<td>RSA_AES_256_SHA</td>
<td>32768</td>
<td>x00008000</td>
<td>35</td>
<td>256-bit AES encryption with SHA-1 message authentication and RSA key exchange</td>
</tr>
<tr>
<td>TRIPLE_DES_SHA_US</td>
<td>256</td>
<td>x00001000</td>
<td>0A</td>
<td>168-bit 3DES encryption with SHA-1 message authentication and RSA key exchange</td>
</tr>
</tbody>
</table>

Setting up an LDAP client

As with the LDAP server, the LDAP client that wants to use SSL/TLS protected communication needs access to a key database, RACF key ring, or PKCS #11 token. If the LDAP server you are going to contact is using a self-signed certificate (as is done frequently while testing SSL/TLS protected communications between an LDAP client and server), then the self-signed certificate of the LDAP server must be stored into the LDAP client’s key database, RACF key ring, or PKCS #11 token.

If the LDAP server you are going to contact is using a certificate which is signed by a certificate authority (CA), you must ensure that the certificate for the CA is contained in the key database, RACF key ring, or PKCS #11 token. Use whatever means is provided by the CA for obtaining the CA certificate. The certificate should be obtainable in a format that is acceptable to the gskkyman utility or RACDCERT command.

If the LDAP server is configured for server and client authentication and the client wants client authentication to occur, then the LDAP client must obtain its own certificate from a CA and store it in the client's own key database, RACF key ring, or PKCS #11 token and mark it as the default.

After the key database file, RACF key ring, or PKCS #11 token is created and contains the proper certificates, then the LDAP client is ready to perform SSL/TLS protected communications with an LDAP server. The LDAP operation utilities (for example, ldapsearch) can be used to communicate securely with the LDAP server using a secure only connection. The utilities are explained in IBM Tivoli Directory Server Client Programming for z/OS.
Using LDAP client APIs to access LDAP using SSL/TLS

The `ldap_ssl_client_init()` and `ldap_ssl_init()` APIs can be used to start a secure only connection to an LDAP server. A description of these APIs can be found in IBM Tivoli Directory Server Client Programming for z/OS.

Support of certificate bind

The SASL bind mechanism of EXTERNAL is supported by the LDAP server. This means that the authentication on the bind is performed using the data obtained during the SSL/TLS client authentication that was performed on the SSL/TLS handshake with the client.

To use SASL EXTERNAL bind, the following steps must occur:

- The LDAP server must be configured and started with `sslAuth` set to `serverClientAuth` so that the server can authenticate the client.
- The client connects to the LDAP server and performs the SSL/TLS handshake. The handshake sends the client certificate to the LDAP server.
- The client performs a SASL bind with the mechanism of EXTERNAL.

At this point, the LDAP server will consider the bind DN of the client for authorization purposes to be the client’s DN as transmitted in the client’s certificate on the handshake. The name specified in the BIND request must match the subject name in the client certificate or must be null.

The RACF RACDCERT command is used to associate a certificate with a RACF user ID. If `check`, `add`, or `replace` is specified on the `sslMapCertificate` configuration option, the LDAP server determines if the client certificate used during an EXTERNAL bind is associated with a RACF user ID, therefore, adds that RACF user ID to the bind information kept for this client.

After an EXTERNAL bind, the client can access LDBM, CDBM, TDBM, and GDBM backends based on the bind DN and the groups it belongs to. If the `sslMapCertificate` option is set to keep the associated RACF user ID in the bind information, the client can perform SDBM operations to access RACF data, under the context of the RACF user ID.

See z/OS Security Server RACF Security Administrator’s Guide for more information about associating a certificate with a RACF user ID.

Configuring for encryption

The LDAP server allows the `userPassword`, `secretKey`, `replicaCredentials`, `ibm-replicaKeyPwd`, and `ibm-slapdMasterPw` attribute values to be encrypted when stored in the directory. This prevents clear data from being accessed by users, including the system administrators. The administrator may configure the server to encrypt `userPassword` attribute values in either a one-way encryption format or a two-way symmetric encryption format. `secretKey`, `replicaCredentials`, `ibm-replicaKeyPwd`, and `ibm-slapdMasterPw` attribute values can only be encrypted in a two-way symmetric encryption format. Besides encryption, access to data stored in the directory can be protected by the directory access control mechanism.

One-way encryption formats

The one-way encryption formats include crypt, MD5, and SHA. During a simple bind, the bind password is encrypted using the appropriate algorithm and compared with the stored encrypted `userPassword` attribute value. Assuming that a client is authorized using directory access controls to see the `userPassword` attribute values on a search, the values are returned to the client in one of the following manners:
1. If the `pwSearchOutput` configuration option is set to `binary` and the `userPassword` attribute value is encrypted in MD5 or SHA, the LDAP server returns the encryption tag (either {MD5} or {SHA}) in UTF-8 followed by the encrypted binary hash.

2. If the `pwSearchOutput` configuration option is set to `base64` and the `userPassword` attribute value is encrypted in MD5 or SHA, the LDAP server returns the encryption tag (either {MD5} or {SHA}) in UTF-8 followed by the base64-encoded encrypted binary hash.

3. If the `userPassword` attribute value is encrypted in crypt, the LDAP server returns the encryption tag ((crypt)) in UTF-8 followed by the encrypted binary hash and sent over the wire in UTF-8.

The following are examples of retrieving the same SHA encrypted `userPassword` attribute value, using the `ldapsearch` client utility with the `-L` option specified. For additional information about the `ldapsearch` client utility, see "IBM Tivoli Directory Server Client Programming for z/OS".

When the `pwSearchOutput` configuration option is set to `binary`, the SHA encrypted `userPassword` value is displayed by `ldapsearch` as follows:

```
userpassword:: e1NIQX2pmT42RwaBaro+JXF4UMJsnNDynQ==
```

The `userPassword` attribute value that is returned above contains both the UTF-8 encryption tag, {SHA}, and the binary encrypted data, but they have been base64-encoded by the `ldapsearch` client utility to present the value in a printable format.

When the `pwSearchOutput` configuration option is set to `base64`, the SHA encrypted `userPassword` value is displayed by `ldapsearch` as follows:

```
userpassword: {SHA}qZk+NkcGgWq6PiVxeFDCbJzQ2J0=
```

The `userPassword` attribute value that is returned above is displayed as sent to the `ldapsearch` client utility because the encrypted binary data after the {SHA} encryption tag has already been base64-encoded by the LDAP server.

For applications requiring retrieval of clear text passwords or data, such as middle-tier authentication agents, the directory administrator must configure the LDAP server to perform either a two-way encryption or no encryption of user passwords or data.

**Two-way encryption formats**

The supported two-way encryption formats include DES and AES. Two-way encryption allows the values of the `userPassword`, `secretKey`, `replicaCredentials`, `ibm-replicaKeyPwd`, or `ibm-slapdMasterPw` attributes to be retrieved as part of an entry in the original clear format. Some applications, such as middle-tier authentication servers, require passwords to be retrieved in clear text while installation security policies might prohibit storing clear passwords in secondary permanent storage. This option satisfies both requirements. An encrypted password can be used for password matching on a simple bind and can be decrypted for return as clear text on a search request. During simple bind, the stored encrypted `userPassword` attribute values are decrypted and compared with the bind password. During a search, if the client is authorized using directory access controls to see the `userPassword`, `secretKey`, `ibm-replicaKeyPwd`, `ibm-slapdMasterPw`, or `replicaCredentials` attribute values, then they are decrypted and returned as clear text.

**Symmetric encryption keys**

The DES and AES encryption format use symmetric encryption keys. DES uses 56-bit (single length), 112-bit (double length) or 168-bit (triple length) keys while AES uses 256-bit keys. DES or AES keys are stored in the ICSF CKDS or in a sequential dataset referenced by the LDAPKEYS DD statement.

The ICSF KGUP utility is used to store an AES or DES key in the CKDS. See the Key Generator Utility Program in "z/OS Cryptographic Services ICSF Administrator’s Guide" for instructions on how to generate and store a key in the CKDS.
DES and AES keys can be stored in a sequential dataset referenced by the LDAPKEYS DD statement. The dataset consists of fixed-length or variable-length records with a maximum record length of 255. The records are assumed to be in the IBM-1047 code page. Comment records begin with ‘#’ or ‘*’ and blank records are ignored. Each record in the dataset defines a single key and has the following format:

key-label key-part-1 key-part-2 key-part-3 key-part-4

The fields are separated by one or more blanks. Each key part consists of 16 hexadecimal characters representing 8 bytes of the key. A DES key requires the key label and the one, two or three key parts while an AES key requires the key label and all four key parts. In a DES key, the low-order bit in each byte is a parity bit. The parity bit must be set so that there is an odd number of 1s in each byte, but the bit is not used for encryption. Therefore, DES uses 56-bits out of each 8-byte key part for encryption. An AES key does not use parity bits, therefore, the entire key (256 bits) is used for encryption.

The LDAP server will check the LDAPKEYS dataset first when looking for a key label. If the key label is not found, then the LDAP server will attempt to locate an AES or DES key in the ICSF CKDS.

### Configuring for user password encryption

The LDAP server allows prevention of unauthorized access to user passwords in the LDBM, TDBM, and CDBM backends. The `userPassword` attribute values can be encrypted when stored in the directory, which prevents clear text passwords from being accessed by any users, including the system administrators. Use of the terms “user password” and “password” pertain to the `userPassword` attribute. Use of the term “user entry” refers to an entry in LDBM, TDBM, or CDBM that contains a `userPassword` attribute.

The administrator may configure the server to encrypt `userPassword` attribute values in either a one-way encryption format or a two-way, symmetric, encryption format. The `pwEncryption` option in an LDBM, TDBM, or CDBM section of the LDAP server configuration file specifies the encryption method that is to be used to encrypt the `userPassword` attribute values in that LDBM, TDBM, or CDBM backend. For more information about the password encryption types, see the `pwEncryption` option on page 95.

After the server is configured and started, any user passwords for new user entries or modified passwords for existing user entries are encrypted before they are stored in either the LDBM, TDBM, or CDBM backend. The encrypted passwords are tagged with the encryption algorithm name so that passwords encrypted in different formats can coexist in the directory. If a tagged encrypted `userPassword` attribute value is present on an add or modify operation, the attribute value is added as it is, with no additional encryption performed on the value even if the `pwEncryption` configuration option is set to a different type of encryption.

If the `pwEncryption` configuration option in an LDBM, TDBM, or CDBM backend is changed, existing passwords remain unchanged and continue to be usable. In other words, existing user password values are not automatically converted to the new encryption method or key label.

The `db2pwdenc` utility is provided as a migration utility to encrypt all unencrypted, AES encrypted, or DES encrypted `userPassword` attribute values in the encryption method specified by the `pwEncryption` configuration option in the LDBM, TDBM, or CDBM backend. For example, the `db2pwdenc` utility allows an LDAP administrator to convert passwords from AES to DES, DES to AES, or AES to crypt. The `db2pwdenc` utility is similar to the LDAP operation utilities, such as `ldapsearch`, in that it acts such as a client to the LDAP server and has similar command-line parameters. For more information about the `db2pwdenc` utility, see “`db2pwdenc` utility” on page 185. For more information about the LDAP operation utilities, such as `ldapsearch`, see [IBM Tivoli Directory Server Client Programming for z/OS]. The `db2pwdenc` utility must be run by the LDAP server administrator using the `adminDN` and password configured on the server.

If the `pwEncryption` configuration option is changed from AES or DES encryption to another encryption method or to a different AES or DES key label, the LDAP server must have access to the original AES or DES key label so that decryption of existing password values still occurs on bind, search, and compare.
operations. If you want to remove the LDAP server’s access to the original AES or DES key label, it is necessary to migrate all existing AES or DES encrypted passwords to the new encryption method or new AES or DES key label by using the db2pwden utility. After the db2pwden utility has converted all of the AES or DES encrypted passwords to the new encryption method or new AES or DES key label, the LDAP server’s access to the original AES or DES key label can be removed.

A simple bind will succeed if the password provided in the bind request matches with any of the multiple values of the userPassword attribute. Note that depending on when userPassword values are stored in the directory, different attribute values can be encrypted using different encoding methods.

When ldif2ds is used to load a TDBM backend, all clear text userPassword attribute values in new entries are encrypted by the method specified on the pwEncryption configuration option in the LDAP server configuration file. If there is a tagged encrypted userPassword attribute value in an entry, the attribute value is added as it is, with no additional encryption performed on the value even if the pwEncryption configuration option is set to a different type of encryption.

For information about the unloading of userPassword attribute values in the ds2ldif utility, see ds2ldif utility.

Notes:
1. The z/OS LDAP server does not permit userPassword attributes in distinguished names.
2. Some important considerations for password encryption and replication are described in Password encryption and basic replication.
3. The crypt() algorithm, implemented across many platforms, accepts only the first eight characters of a password. As a result, any password supplied on an ldap_simple_bind() or ldap_compare() API that matches the first eight characters of a userPassword attribute value encrypted with the crypt algorithm in the directory will match.

Configuring for secret encryption

The LDAP server allows prevention of unauthorized access to data other than user passwords in the LDBM, TDBM, and CDBM backends. The secretKey, replicaCredentials, ibm-replicaKeyPwd, and ibm-slapdMasterPw attribute values can be encrypted when stored in the directory, which prevents clear text passwords and data from being accessed by any users, including the system administrators. Use of the term “secret encryption” refers to any entry in LDBM, TDBM, or CDBM that contains secretKey, replicaCredentials, ibm-replicaKeyPwd, or ibm-slapdMasterPw attributes.

The administrator may configure an LDBM, TDBM, or CDBM backend in the server to encrypt the secret encryption attribute values in DES or AES by specifying the secretEncryption option in the LDBM, TDBM, or CDBM backend section of the LDAP server configuration file. For more information about encrypting secret encryption attribute values, see the secretEncryption option on page 98.

After the server is configured and started, any secret encryption attribute values for new user entries or modified secret encryption attribute values for existing user entries are encrypted before they are stored in either the LDBM, TDBM, or CDBM backend. The encrypted secret encryption attribute values are tagged with the encryption algorithm name so that values encrypted in different formats can coexist in the directory.

If the secretEncryption configuration option in an LDBM, TDBM, or CDBM backend is changed, existing secret encryption attribute values remain unchanged and continue to be usable. In other words, existing secret encryption attribute values are not automatically converted to the new encryption method or key label.

If the secretEncryption configuration option is changed from AES to DES, DES to AES, or to a different AES or DES key label, the LDAP server must have access to the original AES or DES key label so that decryption of existing values still occurs on bind, search, and compare operations. If you want to remove
the LDAP server’s access to the original AES or DES key label, it is necessary to migrate all existing AES
or DES encrypted secret encryption values to the new AES or DES key label. This is accomplished by
using ldapsearch to retrieve all of the entries that have secret encryption values. For each entry that is
returned by ldapsearch, use ldapmodify to change the secret encryption to the same value returned on
ldapsearch. By modifying secret encryption values in this manner, the LDAP server will re-encrypt the
values using the new AES or DES key label that is specified on the secretEncryption configuration
option. After the conversion of all secret encryption values are completed, the LDAP server’s access to the
original AES or DES key label can be removed.

When ldif2ds is used to load a TDBM backend, all clear text secret encryption attribute values are
encrypted by the method specified on the secretEncryption option in the LDAP server configuration file.

For information about the unloading of secret encryption attribute values, see ds2ldif utility.

---

**Configuring for securityLabel option**

If you are configuring the securityLabel option, you must do one of the following:
- Ensure the LDAP server user ID has read access to the BPX.POE resource in the FACILITY class.
- Assign the LDAP server user ID a UID of 0 when the BPX.POE resource is not defined.

For other LDAP server considerations in a multilevel security environment, see [z/OS Planning for
Multilevel Security and the Common Criteria](#).
Chapter 8. Customizing the LDAP server configuration

The LDAP server configuration file, ds.conf, contains configuration options that are read once, when the LDAP server is started. Changes to this file are not put into effect until the LDAP server is restarted. The ds.conf file is also used by the ds2ldif and ldif2ds utilities. The following topics contain more information about the LDAP server configuration file and options that can be specified in it.

- Creating the ds.conf file
- Configuration file options
- Configuration considerations
- Determining operational mode
- Establishing the administrator DN and basic replication replica server DN and passwords
- Example configuration scenarios

The CDBM backend contains configuration entries whose attributes represent configuration options. These configuration entries are read when the LDAP server is started. The attribute values can be changed dynamically by an LDAP modify command while the LDAP server is running. Most changes take effect immediately, without restarting the LDAP server. The CDBM backend itself is configured in the LDAP server configuration file.

Creating the ds.conf file

This section discusses what is necessary for creating the ds.conf configuration file.

Locating ds.conf

All LDAP server runtime configuration is accomplished through the configuration file ds.conf, installed in the /usr/lpp/ldap/etc directory. If this is your first time installing the LDAP server, create a new copy of ds.conf with:

cp /usr/lpp/ldap/etc/ds.conf /etc/ldap/ds.conf

and edit /etc/ldap/ds.conf.

The default LDAP server configuration file is /etc/ldap/ds.conf. A different configuration file can be specified using the -f command-line parameter when starting the LDAP server, ds2ldif, and ldif2ds utilities.

The initial configuration contains default versions of some configuration settings. It does not contain a database suffix.

Configuration file format

The ds.conf file consists of the following sections:

- Global section
  Contains configuration options that apply to the LDAP server as a whole (including all backends).

- SDBM backend-specific section
  Contains configuration options that apply to the SDBM backend.

- TDBM backend-specific section
  Contains configuration options that apply to the TDBM backend. It is possible to have one or more of these sections depending on how many TDBM backends your installation uses.

- GDBM (DB2-based) backend-specific section
  Contains configuration options that apply to the DB2-based GDBM change log backend. The
configuration of either type of GDBM change log backend is mutually exclusive. The LDAP server will end when a file-based GDBM backend and a DB2-based GDBM backend are both configured in the same configuration file.

**LDBM backend-specific section**
Contains configuration options that apply to the LDBM backend. It is possible to have one or more of these sections depending on how many LDBM backends your installation uses.

**GDBM (file-based) backend-specific section**
Contains configuration options that apply to the file-based GDBM change log backend. The configuration of either type of GDBM change log backend is mutually exclusive. The LDAP server will end when a file-based GDBM backend and a DB2-based GDBM backend are both configured in the same configuration file.

**CDBM backend-specific section**
Contains configuration options that apply to the file-based CDBM backend.

**EXOP backend-specific section**
Contains only the `database` statement necessary for the EXOP backend.

Figure 6 shows the general format of `ds.conf`.

```
# Global options - these options apply to every database
<global configuration options>

# SDBM database definition and configuration options
database SDBM GLDBSD31/GLDBSD64
<configuration options specific to SDBM backend>

# TDBM database definition and configuration options
database TDBM GLDBTD31
<configuration options specific to TDBM backend>

# DB2-based GDBM database definition and configuration options
database GDBM GLDBGD31
<configuration options specific to DB2-based GDBM backend>

# LDBM database definition and configuration options
database LDBM GLDBLD31/GLDBLD64
<configuration options specific to LDBM backend>

# File-based GDBM database definition and configuration options
database GDBM GLDBGD31/GLDBGD64
<configuration options specific to file-based GDBM backend>

# CDBM database definition and configuration options
database CDBM GLDBCD31/GLDBCD64
<configuration options specific to CDBM backend>

# EXOP database definition and configuration options
database EXOP GLDXPD31/GLDXPD64
```

**Figure 6. General format of ds.conf**

Noted below are some rules for setting up `ds.conf`:

- The configuration file contains a global section containing options that apply to the entire LDAP server, followed by one or more database backend sections that contain options that apply to a specific backend. Each backend section begins with a `database` option and continues to the next `database` option. The global section starts at the beginning of the configuration file and ends at the first `database` option. The `sizelimit` and `timelimit` options can be either global or specific to a backend: they are
global if they appear in the global section and are specific to a backend if they appear in a backend section. See the descriptions of these options for more information.

- The configuration file must be in code page IBM-1047.
- The maximum length of a line in the configuration file is 1024 characters.
- Each configuration line consists of an option and a value separated by one or more blanks. Begin each configuration option in column one. The option is not case-sensitive. The value may or may not be case-sensitive depending upon the option. If an argument contains one or more blank spaces, the value should be enclosed in double quotation marks (for example, "value one"). If a value contains a double quotation mark or a backslash character (\), the double quotation mark or backslash character should be preceded by a backslash character (\).  
- A line that begins with a space or tab character in column one is considered a continuation of the previous line. Everything after the space or tab character is appended to the previous line. The maximum length of the initial line plus all continuation lines is 1024 characters.
- A line that begins with a pound sign (#) in column one is a comment line. Comment lines can be continued and are ignored.
- A pound sign (#) can be used to add a comment to the end of a configuration line. The pound sign (#) must be separated from the option value by at least one blank. Anything following the pound sign (#) is ignored, including any continuation lines. A pound sign (#) will not be treated as the start of a comment if it occurs within a quoted value.
- Blank lines can be used to separate configuration lines.
- Options that expect a value of on or off will also accept yes, y, no and n. The option value is not case-sensitive.
- For single-valued options that appear more than once, the last appearance in the ds.conf file is used.

Specifying a value for filename
For configuration file options that contain the filename value, the value can be specified in one of the following ways:

/pathname/filename
   Specifies the full path name of a file in the z/OS UNIX System Services file system.

filename
   Specifies a file name in the z/OS UNIX System Services file system that is relative to the current working directory of the LDAP server. The LDAP server sets the current working directory to the value of the HOME environment variable or to /etc/ldap if the HOME environment variable is not defined. This format is not recommended.

//dataset.name'
   Specifies the fully-qualified name of a file stored in a sequential dataset.

//dataset.name(member)
   Specifies the fully-qualified name of a file stored in a partitioned dataset.

//DD:DDNAME
   Specifies the DDNAME for a JCL DD statement that defines the file. The file may be in a UNIX System Services file system, a sequential dataset, or a member of a partitioned dataset.

Specifying a value for a distinguished name
- The value for the following configuration options is a distinguished name (DN): adminDN, krbLDAPAdmin, masterServerDN, peerServerDN, nativeAuthSubtree, and suffix. Also, the wlmExcept configuration option optionally accepts a distinguished name (DN). Special characters (as identified in RFC 2253) used in the DN must be properly escaped using two back slashes (\). Note that the double backslashes are only needed in the configuration file; in all other usages, the special characters are typically prefixed by a single backslash. See Chapter 16, "Accessing RACF information" for exceptions to this when...
For example, to use a RACF userid #1admin as the LDAP administrator, the adminDN configuration option may look similar to:

```
adminDN "racfid=\#1admin,profiletype=user,cn=myRacf"
```

With LDAP V3 support, UTF-8 characters can be used for textual attributes stored in the directory. It is preferred to allow any UTF-8 character to appear in distinguished names, and in particular, the adminDN distinguished name. Because the LDAP configuration files are defined to hold information in the IBM-1047 character set, a solution is required for the configuration files that allows you to use distinguished names containing UTF-8 characters but using only the IBM-1047 character set. To solve this problem, an escape mechanism has been introduced for purposes of entering a DN. This escape mechanism allows the entry of UTF-8 characters while keeping the input string value to within the IBM-1047 character set. The escape mechanism employed requires that you express UTF-8 characters that are not within the X'00' - X'7F' range (7-bit ASCII that is the single-byte form of UTF-8 characters) in the form of a set of four character representations. This representation has the form "&nmm" where 0<n<3 and 0<m<7. You might recognize nmm as being an octal value for a byte of information. Therefore, if you want to create the following distinguished name:

```
cn=Peter <U umlaut>nger, o=Widgets, c=DE
```

specify the DN as:

```
cn=Peter &nmm&nmmnger, o=Widgets, c=DE
```

Because the <U umlaut> is not within the 7-bit ASCII range, the value must be escaped to the octal representation of the UTF-8 multi-byte character. In the case of <U umlaut>, the Unicode code point is X'00DC'. Converted to UTF-8, this character is a multi-byte sequence: X'C39C'. (See UTF-8 support for conversion information.) Converted to the escaped form for input into the DN field, this character is represented as "&303&234" because X'C3' is octal 303 and X'9C' is octal 234. Therefore, the DN above is entered as:

```
cn=Peter &303&234nger, o=Widgets, c=DE
```

If there is a case where you need to enter a DN that contains the string "&nmm" where 0<n<3 and 0<m<7, then you must escape the ampersand by using its octal representation which is "&046".

There are several restrictions concerning the attributes used in a DN:

1. The default matching rule for an attribute used in a DN that is not contained in the LDAP server schema is caselgignoreMatch. This results in this part of the DN being normalized by using uppercase in the attribute value. If the attribute is later defined in the schema and the matching rule is not caselgignoreMatch, that part of the DN may now be normalized differently. Therefore, the normalized version of the DN used in an operation may not match the normalized version of the DN in the configuration file, resulting in the failure of the operation. In particular, if an attribute used in a suffix is not in the LDAP server schema and is later added to the schema with a different matching rule, then restart the LDAP server after the schema definition is added. Otherwise, operations using that suffix can fail.

2. Do not use an attribute with Integer syntax (1.3.6.1.4.1.1466.115.121.1.27) in a DN in the configuration file, especially in a suffix. Integer attribute values in a normalized DN have a special format that may cause problems within the LDAP server.

**Configuration file checklist**

The following table is provided to assist you in determining which configuration file options you will need to use in your ds.conf file. Depending on the section in the configuration file (Global, SDBM, TDBM, GDBM, LDBM, CDBM, or EXOP), certain topics (SSL, schema, replication, and so on) have options that are
Table 9. Configuration file options checklist

<table>
<thead>
<tr>
<th>Section/topic</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global</td>
<td>adminDN is required. adminPW, allowAnonymousBinds, altServer, armName, audit, commThreads, db2StartUpRetryInterval, db2StartUpRetryLimit, db2Terminate, digestRealm, dnCacheSize, idleConnectionTimeout, include, listen, logfile, maxConnections, operationsMonitor, operationsMonitorSize, pclIdleConnectionTimeout, pcThreads, plugin, pwSearchOutput, referral, schemaPath, schemaReplaceByValue, securityLabel, sendV3stringsoverV2as, serverCompatLevel, serverEtherAddr, sizeLimit, srvStartUpError, tcpTerminate, timeLimit, validateincomingV2strings, and wlmExcept are optional.</td>
</tr>
<tr>
<td>SSL/TLS</td>
<td>sslKeyRingFile is required if a listen option is initialized for secure socket communications or a listen option is initialized for non-secure socket communications that is intended to support switching to secure socket communications when the connection is established. sslAuth, sslCertificate, sslCipherSpecs, sslKeyRingFilePW, sslKeyRingPWStashFile, and sslMapCertificate are optional.</td>
</tr>
<tr>
<td>Sysplex</td>
<td>serverSysplexGroup is required.</td>
</tr>
<tr>
<td>Kerberos</td>
<td>supportKrb5 is required. krbKeytab, krbLDAPAdmin, and serverKrbPrinc are optional.</td>
</tr>
<tr>
<td>SDBM backend</td>
<td>database and suffix are required. enableResources, include, readOnly, sizeLimit and timeLimit are optional.</td>
</tr>
<tr>
<td>Kerberos</td>
<td>krbIdentityMap is required.</td>
</tr>
<tr>
<td>TDBM backend</td>
<td>database, dbuserid, and suffix are required. aclSourceCacheSize, attrOverflowCount, attrOverflowSize, changeLoggingParticipant, dnToEidCacheSize, dsnaoini, entryCacheSize, entryOwnerCacheSize, extendedGroupSearching, filterCacheBypassLimit, filterCacheSize, include, persistentSearch, readonly, serverName, sizeLimit, and timeLimit are optional.</td>
</tr>
<tr>
<td>Attribute encryption</td>
<td>pwCryptCompat, pwEncryption, and secretEncryption are optional.</td>
</tr>
<tr>
<td>Basic replication</td>
<td>Either peerServerDN or both masterServer and masterServerDN are required. peerServerPW and masterServerPW are optional.</td>
</tr>
<tr>
<td>Multi-server</td>
<td>multiserver is required.</td>
</tr>
<tr>
<td>Kerberos</td>
<td>krbIdentityMap is required.</td>
</tr>
<tr>
<td>Native authentication</td>
<td>useNativeAuth and nativeAuthSubtree are required. nativeUpdateAllowed is optional.</td>
</tr>
<tr>
<td>GDBM backend (DB2-based)</td>
<td>database and dbuserid are required. aclSourceCacheSize, attrOverflowSize, changeLogging, changeLoggingParticipant, changeLogMaxAge, changeLogMaxEntries, dnToEidCacheSize, dsnaoini, entryCacheSize, entryOwnerCacheSize, filterCacheBypassLimit, filterCacheSize, include, persistentSearch, readonly, serverName, sizeLimit, and timeLimit are optional.</td>
</tr>
<tr>
<td>Section/topic</td>
<td>Options</td>
</tr>
<tr>
<td>--------------</td>
<td>---------</td>
</tr>
<tr>
<td>Multi-server</td>
<td>multiserver is required.</td>
</tr>
<tr>
<td>LDBM backend</td>
<td>database and suffix are required. attrOverflowCount, changeLoggingParticipant, commitCheckpointEntries, commitCheckpointTOD, databaseDirectory, extendedGroupSearching, fileTerminate, filterCacheBypassLimit, filterCacheSize, include, persistentSearch, readOnly, sizeLimit, and timeLimit are optional.</td>
</tr>
<tr>
<td>Attribute encryption</td>
<td>pwCryptCompat, pwEncryption, and secretEncryption are optional.</td>
</tr>
<tr>
<td>Basic replication</td>
<td>Either peerServerDN or both masterServer and masterServerDN are required. peerServerPW and masterServerPW are optional.</td>
</tr>
<tr>
<td>Multi-server</td>
<td>multiserver is required.</td>
</tr>
<tr>
<td>Kerberos</td>
<td>krbIdentityMap is required.</td>
</tr>
<tr>
<td>Native authentication</td>
<td>useNativeAuth and nativeAuthSubtree are required. nativeUpdateAllowed is optional.</td>
</tr>
<tr>
<td>GDBM backend (file-based)</td>
<td>database is required. changeLogging, changeLoggingParticipant, changeLogMaxAge, changeLogMaxEntries, commitCheckpointEntries, commitCheckpointTOD, databaseDirectory, fileTerminate, filterCacheBypassLimit, filterCacheSize, include, persistentSearch, readOnly, sizeLimit, and timeLimit are optional.</td>
</tr>
<tr>
<td>Multi-server</td>
<td>multiserver is required.</td>
</tr>
<tr>
<td>CDBM backend</td>
<td>database is required. attrOverflowCount, changeLoggingParticipant, commitCheckpointEntries, commitCheckpointTOD, databaseDirectory, extendedGroupSearching, fileTerminate, filterCacheBypassLimit, filterCacheSize, include, persistentSearch, readOnly, sizeLimit, and timeLimit are optional.</td>
</tr>
<tr>
<td>Attribute encryption</td>
<td>pwCryptCompat, pwEncryption, and secretEncryption are optional.</td>
</tr>
<tr>
<td>Basic replication</td>
<td>Either peerServerDN or both masterServer and masterServerDN are required. peerServerPW and masterServerPW are optional.</td>
</tr>
<tr>
<td>Advanced replication</td>
<td>useAdvancedReplication is required.</td>
</tr>
<tr>
<td>Multi-server</td>
<td>multiserver is required.</td>
</tr>
<tr>
<td>Kerberos</td>
<td>krbIdentityMap is required.</td>
</tr>
<tr>
<td>Native authentication</td>
<td>useNativeAuth and nativeAuthSubtree are required. nativeUpdateAllowed is optional.</td>
</tr>
<tr>
<td>EXOP backend</td>
<td>database is required. include is optional.</td>
</tr>
</tbody>
</table>

Note: Be sure to specify adminDN. You can specify the adminPW here or in a directory entry. See Establishing the administrator DN and basic replication replica server DN and passwords for more details.
information. Note that the use of the **adminPW** option is strongly discouraged. Instead, an existing entry in the directory should be designated as the **adminDN**.

### Configuration file options

This section contains an alphabetic listing of the configuration file options. For each option, a table shows an X in the areas (Global, TDBM, LDBM, SDBM, GDBM, CDBM, and EXOP) of the configuration file where the option can be used.

**Note:** Some GDBM options can only be specified when GDBM is configured to be DB2-based and others can only be used when GDBM is file-based. See Configuration file checklist for a list of which options can be configured for each type of GDBM configuration.

<table>
<thead>
<tr>
<th><strong>aclSourceCacheSize num-entries</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Global</strong></td>
</tr>
<tr>
<td>---------</td>
</tr>
<tr>
<td>X</td>
</tr>
</tbody>
</table>

Specifies the maximum number of entries to store in the ACL Source cache. This cache holds information regarding ACL definitions within the database. Retrieval of information from this cache avoids database read operations when resolving access permissions.

The maximum size of this value is 2147483647. A value of 0 indicates that the cache is not used.

**Default =** 100

<table>
<thead>
<tr>
<th><strong>adminDN dn</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Global</strong></td>
</tr>
<tr>
<td>---------</td>
</tr>
<tr>
<td>X</td>
</tr>
</tbody>
</table>

The distinguished name (DN) of the administrator for this LDAP server. Typically, this DN has unrestricted access to all entries in the directory except for entries in backends that are read-only replicas. When the LDAP server is in maintenance mode, the LDAP administrator has unrestricted access to all entries in the directory. The name that is chosen should be descriptive of the person that will have knowledge of and administer the LDAP server. The format of the name must be in DN format that is described in Chapter 13, "Data model." It is recommended, though not necessary, that the DN have the same suffix as one of the **suffix** option values in the configuration file.

[Establishing the administrator DN and basic replication replica server DN and passwords](#) describes how to set up your administrator DN.

For information about specifying a value for a distinguished name for this option, see [Specifying a value for a distinguished name](#).

<table>
<thead>
<tr>
<th><strong>adminPW string</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Global</strong></td>
</tr>
<tr>
<td>---------</td>
</tr>
<tr>
<td>X</td>
</tr>
</tbody>
</table>

The password of the administrator (**adminDN**) for this server.

[Establishing the administrator DN and basic replication replica server DN and passwords](#) describes how to set up your administrator password.
**Note:** Use of the *adminPW* configuration option is strongly discouraged in production environments. Instead, specify your *adminDN* as the distinguished name of an existing entry in the directory information tree. This will eliminate passwords from the configuration file.

**allowAnonymousBinds (on | off)**

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Specifies whether an LDAP client can perform unauthenticated operations on the LDAP server. If *off*, clients must explicitly bind to the server with a distinguished name. If *on*, a client may access the server without binding with a distinguished name and will have access to data as a member of the *cn=anybody* group. See [Chapter 22, "Using access control"](z/OS MVS Setting Up a Sysplex) for more information about access control of directory data.

Default = *on*

**altServer *ldap_URL***

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Specifies an equivalent server to this LDAP server. It may or may not be a replica, but should contain the same naming contexts. There is no required format for the value, however, LDAP URL format is most commonly used and supported by LDAP clients. See page 86 for a description of LDAP URL format. The option may be specified multiple times to define more than one alternate server. The alternate servers are placed in the *altServer* attribute in the root DSE and can be queried by LDAP clients to determine other servers that may be contacted in case this server is not available at some later time.

In the following example, myldap.server.com is the host name and 3389 is the port number of the LDAP directory URL:

```
altsServer ldap://myldap.server.com:3389
```

In the following example,

```
5f1b:df00:ce3e:e200:20:800:2078:e3e3
```

is the IPv6 address and 389 is the port number of the LDAP URL:

```
altsServer ldap://[5f1b:df00:ce3e:e200:20:800:2078:e3e3]:389
```

**armName *name***

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Specifies the name that the LDAP server uses when registering with the Automatic Restart Management (ARM) service. The name is 1-7 characters and can consist of letters, numbers, and the special characters `$ # @ _`. Lowercase letters are converted to uppercase. The first character may not be a number. The system name is appended to form the element name. The *armName* configuration option must be specified if you is running multiple instances of the LDAP server on the same system and ARM processing is enabled. See [z/OS MVS Setting Up a Sysplex](z/OS V1R11.0 IBM Tivoli Directory Server Administration and Use for z/OS) for more information about automatic restart manager.

For example, for system DCESEC4, specifying:

```
armaName LDAP1
```
will result in the element name LDAP1_DCESEC4.

The LDAP server registers with ARM using the element name formed from the `armName` configuration option, an element type of 'SYSLDAP', an element bind of 'CURJOB', and a termination type of 'ELEMTERM'. See the description of the IXCARM macro in [z/OS MVS Programming: Sysplex Services Reference](https://publib.boulder.ibm.com/infocenter/pseries/v2r2/index.jsp?topic=/com.ibm.zos.r22.doc_colls/aix_libref.htm) for more information about these parameters and how to override them using the current ARM policy.

Default = GLDSRVR

## attrOverflowCount `count`

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For TDBM, specifies the number of attribute values required to store the attribute values in a long attribute value table. The choice of this value allows large multi-valued attributes such as group membership lists to be stored in a separate table with its own index.

For LDBM and CDBM, specifies the number of attribute values required to store the attribute values in an internal indexed table, providing quicker access to the values of large multi-valued attributes such as group membership lists.

The value must be either 0 or in the range 64 to 2147483647. A value of 0 disables attribute overflow based on the attribute value count.

Default = 512

## attrOverflowSize `num-of-bytes`

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Specifies, in bytes, the minimum size of an attribute value required to store the value in a long attribute value table. The choice of this value allows large attribute values (such as JPEG and GIF files) to be stored in a separate DB2 table in a separate DB2 table space. The maximum size of this value is 2147483647. A value of 0 disables attribute overflow based on attribute size.

Default = 255

## audit {on | off | all,operations | error,operations | none,operations}

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Turns LDAP auditing on or off and specifies which operations are to be audited and the associated audit level. When auditing is on, an LDAP SMF type 83 subtype 3 audit record is generated for an operation if the operation is specified on an audit option and the operation result matches the audit level.

This option can be specified multiple times, once to turn auditing on or off and once or more times for each audit level to specify the operations to audit for that level. Multiple operations can be specified for a level by either putting a + between them on the audit option or by specifying multiple audit options with the same level.

Operations can be audited all the time or only when they fail. The following audit levels are supported:

- **all** An LDAP audit record is generated for the specified operations.
error
An LDAP audit record is generated for the specified operations when they fail.

none
An LDAP audit record is not generated for the specified operations.

The supported values for operations can be one or more of: add, bind, compare, connect, delete, disconnect, exop, modify, modifydn, search, unbind.

If an operation is specified in more than one level, the last level is used for the operation. If an operation is not specified in any level, the level defaults to none for that operation.

The LDAP server AUDIT operator modify command can be used to change the audit settings and to turn audit on or off while the LDAP server is running. See LDAP server operator commands for more information.

Default = off

For example, the following audit options turn auditing on for modify, search, and bind failures and for all add operations. The other operations are not audited.

audit error,modify+search+bind
audit all,add
audit on

changeLogging {on | off}

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Turns change logging on or off.

When change logging is on, all change logging operations are allowed. When change logging is off, change log entries can be searched, modified, and deleted, but no new change log entries can be created and no automatic trimming of the change log is performed.

Default = on

changeLoggingParticipant {on | off}

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Allows/disallows change logging for changes made to entries in this backend. When specified in GDBM, changeLoggingParticipant controls the logging of modifications to the LDAP server schema entry.

Note: This option does not turn on or off change logging. That is done by the changeLogging option.

Default = on

changeLogMaxAge nnn

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

Specifies the maximum age in seconds of an entry in the change log. Change log entries are deleted when they have been in the change log longer than this value, except if changeLogging off is specified. The value must be between 0 and 2147483647. A value of 0 indicates that there is no maximum.
changeLogMaxEntries \textit{nnn}

<table>
<thead>
<tr>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

Specifies the maximum number of entries that the change log can contain. If the number of change log entries exceeds this value and \textit{changeLogging off} is not specified, change log entries with the lowest change numbers are deleted. If the change log is DB2-based, change log entries are deleted until the number of remaining entries is 95% of the maximum. If the change log is file-based, change log entries are deleted until the number of remaining entries is the maximum. The value must be between 0 and 2147483647. A value of 0 indicates that there is no maximum.

Default = 0

commitCheckpointEntries \textit{nnn}

<table>
<thead>
<tr>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

Specifies the maximum number of entries in the checkpoint file. An entry is added to the LDBM, CDBM, or file-based GDBM checkpoint file each time a directory entry is added, changed, deleted, or renamed. When the maximum number is reached, the entries in the checkpoint file are merged into the database file and the entries are removed from the checkpoint file. The value must be between 0 and 2147483647. A value of 0 indicates there is no maximum.

Default = 10000

commitCheckpointTOD \textit{hh:mm}

<table>
<thead>
<tr>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

Specifies a time of day at which the checkpoint file is merged into the database file. An entry is added to the LDBM, CDBM, or file-based GDBM checkpoint file each time a directory entry is added, changed, deleted, or renamed. Every day at the specified time, the entries in the checkpoint file are merged into the database file and the entries are removed from the checkpoint file. The value must be between 00:00 and 23:59. Specify a value outside this range to disable time of day checkpoint processing.

Default = 00:00

commThreads \textit{num-threads}

<table>
<thead>
<tr>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

Specifies the number of threads to be initialized for the communication thread pool. This thread pool handles the connections between the LDAP server and its clients. It is recommended that \textit{commThreads} be set to approximately two times the number of processors that are running in your LPAR. However, this is a general rule depending upon the activity that your LDAP server experiences.

Default = 10

The \textit{commThreads} option deprecates the \textit{maxThreads} and \textit{waitingThreads} options, that are no longer evaluated by the LDAP server.
### database dbtype dlibpath [name]

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

Marks the beginning of a new database section. All global options must appear before the first database section. All options after the `database` option pertain to this backend until another `database` option is encountered.

- **For `dbtype`:**
  - Specify `tdbm` (DB2-based), `ldbm` (file-based), `sdbm` (RACF-based), `gdbm` (DB2-based or file-based), `cdbm` (file-based), or `exop` (extended operations).
  
  **Note:** The server compatibility level must be at least 5 when the CDBM backend is configured. See page 100 for more information about the `serverCompatLevel` configuration option.

- **For `dlibpath`:**
  - This is the file name of the shared library (DLL) containing the backend database code.
  
  Unless you have changed the names of the LDAP DLLs, specify `GLDBTD31` when `dbtype` is `tdbm`, `GLDBLD31/GLDBLD64` when `dbtype` is `ldbm`, `GLDBSD31/GLDBSD64` when `dbtype` is `sdbm`, `GLDBGD31/GLDBGD64` when `dbtype` is `gdbm`, `GLDBCD31/GLDBCD64` when `dbtype` is `cdbm`, and `GLDXPD31/GLDXPD64` when `dbtype` is `exop`.
  
  **Note:** The `GLDBGD64` DLL provides 64-bit support only for the file-based version of GDBM.
  
  It cannot be used when GDBM is configured to be DB2-based.

- **For `name`:**
  - This value is a name that is used to identify this backend. You cannot specify `schema`, `rootDSE`, or `Monitor` as the name. A name is generated if no name is specified for a backend. However, a name must be specified if the `multiserver on` option is specified for this backend and the name must not be longer than 8 characters. In addition, when multi-server mode is active, the same name must be specified for each instance of the backend within the cross-system group.

### databaseDirectory name

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Specifies the name of the directory containing the data files used by the backend to store directory data. A fully-qualified directory path must be specified. A unique directory must be specified for each backend. In addition, when multi-server mode is active, the same directory path must be specified for each instance of the backend within the cross-system group.

- **LDBM Default = /var/ldap/ldbm**
- **GDBM Default = /var/ldap/gdbm**
- **CDBM Default = /var/ldap/schema** if `schemaPath` not specified, else `schemaPath` option setting

### dbuserid userid

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Specifies a z/OS user ID that is the owner of the DB2 tables. When specified in a GDBM backend section, this option indicates that the GDBM backend is DB2-based and not file-based.
Note: The dbuserid value must be unique within the configuration file. Multiple backends on an LDAP server cannot share a database.

**db2StartUpRetryInterval num-seconds**

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Specifies the number of seconds the LDAP server should wait before each DB2 connection retry attempt as a consequence of the initial DB2 connection failure.

During LDAP initialization, an initial attempt at establishing a DB2 connection is made if at least one DB2–based backend is defined. If the connection attempt is unsuccessful and the LDAP server is set up to wait for DB2, the LDAP server retries the connection for a specified number of times, waiting for **db2StartUpRetryInterval** seconds before each retry attempt. While waiting for a connection to DB2, the LDAP server does not receive requests. The value must be between 1 and 999.

**Note: db2StartUpRetryInterval** is ignored if no DB2-based backend (TDBM and DB2-based GDBM) is defined or if the **db2StartUpRetryLimit** configuration option has a zero value or is not specified.

Default = 45

**db2StartUpRetryLimit num-retries**

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Specifies a limit of the number of DB2 connection retries the LDAP server attempts as a result of the initial DB2 connection failure.

During LDAP initialization, an initial attempt at establishing a DB2 connection is made if at least one DB2–based backend is defined. If the connection attempt is unsuccessful and **db2StartUpRetryLimit** has a non-zero value, the LDAP server retries the connection for the specified **db2StartUpRetryLimit** times, waiting for the specified **db2StartUpRetryInterval** number of seconds before each retry attempt. When the number of retry attempts equals **db2StartUpRetryLimit** and a connection to DB2 still cannot be established, all backends that require DB2 fail to configure. While waiting for a connection to DB2, the LDAP server does not receive requests. The value must be between 0 and 99. A value of 0 indicates that no DB2 connection retries are to be attempted.

**Note: db2StartUpRetryLimit** is ignored if no DB2-based backend (TDBM and DB2-based GDBM) is defined.

Default = 0

**db2Terminate {terminate | recover | restore}**

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Specifies how the LDAP server will react to a termination of DB2 after the server successfully starts.

If set to **terminate**, the LDAP server will shutdown.

If set to **recover** or **restore**, the LDAP server will disconnect from DB2 but remain running to allow access to non-DB2 backends (for example, SDBM, LDBM, CDBM, and file-based GDBM).
DB2 is once again active, the LDAP server will reconnect to DB2. There is no access allowed to
DB2-based backends (TDBM and DB2-based GDBM) during the time when DB2 is down. Client
requests to those backend are rejected with LDAP_UNAVAILABLE return code and a reason code
message that includes "DB2 Unavailable".

**Note:** `db2Terminate` is ignored and no DB2 monitoring is done if no DB2-based backend (TDBM
and DB2-based GDBM) is configured.

If using a sysplex distributor, this configuration option should be set to `terminate`. This will allow
client requests to be routed to other LDAP servers in the sysplex who can connect to their
databases.

Default = `recover`

digestRealm `hostname`

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Specifies a realm name to be used when doing DIGEST-MD5 or CRAM-MD5 SASL authentication
binds to the LDAP server. The `digestRealm` is used to help calculate a hash for DIGEST-MD5
and CRAM-MD5 authentication binds. It is suggested that the `hostname` be a DNS-host name and
not an IP address.

Default = fully qualified hostname of the LDAP server if a DNS (Domain Name Server) is active on
the system. Otherwise, the default is the name of the host processor.

dnCacheSize `num-entries`

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Specifies the maximum number of entries to store in the Distinguished Name normalization cache.
This cache holds information related to the mapping of Distinguished Names between their raw
form and their canonical form. Retrieval of information from this cache reduces processing
required to locate entries in the database.

The maximum size of this value is 2147483647. A value of 0 indicates that the cache is not used.

Default = 1000

dnToEidCacheSize `num-entries`

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Specifies the maximum number of entries to store in the Distinguished Name to Entry Identifier
mapping cache. This cache holds information related to the mapping of Distinguished Names in
their canonical form and their Entry Identifier within the database. Retrieval of information from this
cache avoids database read operations when locating entries within the database.

The maximum size of this value is 2147483647. A value of 0 indicates that the cache is not used.

Default = 1000

dsnahini `dsname`

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Specifies the name of the CLI Initialization file or sequential data set (or PDS member) you created in step 4 in Getting DB2 installed and set up for CLI and ODBC. This must be either a fully-qualified dataset name, a DD name or a path name. A dataset name is not enclosed in single quotes or prefixed with ‘/’, a DD name starts with ‘//’, and a path name starts with ‘/’ or ‘./’.

There are three ways to specify the CLI initialization file and the search order is as follows:

1. The DSNAOINI DD statement in the JCL for the LDAP server started task
2. The DSNAOINI environment variable
3. The `dsnaoini` configuration option. If the `dsnaoini` configuration option is specified for a backend, the option must also be specified, with the same value, for all the TDBM and DB2-based GDBM backends in the configuration file.

Running the LDAP server using data sets gives more information about this process. See DB2 ODBC Guide and Reference for details on ways to specify the CLI initialization file. In order for the TDBM or GDBM backend to run, the initialization file must be specified in one of the ways indicated.

**enableResources (on | off)**

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

Specifies whether the SDBM backend supports operations on RACF resources and classes. If **on**, SDBM accepts operations for the setropts, class, and resource profile entries. LDAP also accepts requests for creating a change log entry for a change to a RACF resource profile. If **off**, an SDBM search from the suffix does not return these entries and operations (including a change log request) involving these entries are rejected.

Default = **off**

**entryCacheSize num-entries**

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Specifies the maximum number of entries to store in the Entry cache. This cache holds information contained within individual entries in the database. Retrieval of information from this cache avoids database read operations when processing entries within the database.

The maximum size of this value is 2147483647. A value of 0 indicates that the cache is not used.

Default = **5000**

**entryOwnerCacheSize num-entries**

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

Specifies the maximum number of entries to store in the Entry Owner cache. This cache holds information regarding ACL definitions within the database. Retrieval of information from this cache avoids database read operations when resolving access permissions.

The maximum size of this value is 2147483647. A value of 0 indicates that the cache is not used.

Default = **100**

**extendedGroupSearching (on | off)**
Specifies whether a backend participates in extended group membership searching on a client bind request. If this option is on, group memberships are gathered from this backend during LDAP directory bind processing in addition to the backend in which the bind DN resides. If this option is off, group memberships are not gathered from this backend unless the bind DN resides in this backend.

See “Associating DNs and access groups with a bound user” on page 352 for information about group gathering after a successful bind.

The server control authenticateOnly is supported by the LDAP server so that a client can override both extendedGroupSearching and group membership gathering from the backend where the DN resides. See Appendix C, “Supported server controls” for more information.

This option applies only to the backend in which it is defined.

Default = off

fileTerminate {terminate | recover}

Specifies whether the LDAP server ends when file system errors occur. If terminate, the LDAP server ends when a file system error is detected. If recover, the LDAP server continues processing, but the backend experiencing the file system error is set to read-only mode. No updates can be made to the directory controlled by this backend. When the problem is corrected, the backend can be reset to read-write mode using the LDAP server BACKEND operator modify command. See “LDAP server operator commands” on page 153 for information about the LDAP server BACKEND modify command.

Default = recover

filterCacheBypassLimit num-entries

Specifies the maximum number of returned entries allowed in the result set of any individual search that is stored in the Search Filter cache. Search filters that match more than this number of entries will not be added to the Search Filter cache. This option is useful for maintaining the effectiveness of the Search Filter cache and Entry cache. It can be used to prevent a few search requests with large result sets from dominating the contents of the Entry cache.

The value must be in the range of 1 to 250. This option is ignored when the filter cache is not in use.

Default = 100.

filterCacheSize num-filters

Specifies the maximum number of filters to store in the Search Filter cache. This cache holds information related to the mapping of search request inputs and the result set. Retrieval of
information from this cache avoids database read operations when processing search requests. Individual search requests which return more entries than specified in the filterCacheBypassLimit option are not placed in the cache.

The maximum size of this value is 2147483647. A value of 0 indicates that the cache is not used.

TDBM Default = 500  
LDBM Default = 5000  
CDBM Default = 5000  
GDBM Default = 0

idleConnectionTimeout *num-seconds*

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Specifies the amount of time in seconds that the LDAP server will wait on an idle connection before dropping a particular client connection. Idle connections are detected by the LDAP server’s network monitor task, that wakes up every 30 seconds to check for them. Therefore, it is possible for an idle connection to remain connected slightly longer than the idleConnectionTimeout value.

The value must be either 0 or between 30 and 2147483647. A value of 0 indicates that an idle connection will remain indefinitely.

Default = 0 (indefinitely)  
Recommended value = 1800 (30 minutes)

include *filename [systemName]*

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Specifies the path and file name of a file to be included as a part of the LDAP server configuration. See Specifying a value for filename for information about specifying filename.

Note that the LDAP server will not detect loop conditions in a set of included files. Configuration may encounter errors or fail if the same file is processed more than once. While nested include files are supported, including the same file in such a way as to form a loop condition is not supported.

If the optional MVS™ system name is specified, the include file is processed only on the specified system. This allows the LDAP server configuration files to be shared by multiple servers where each server runs on a different z/OS system. System-specific configuration information is then be placed in an include file that is processed only on the system that it applies.

krbIdentityMap [on | off]

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Specifies if this backend will participate in Kerberos identity mapping. If it will participate, then the server will attempt to map the Kerberos identity that performed the bind to DNs that exist in this backend. The mapped DNs will then be used for access control.

Default = off

krbKeytab {krbKeytab | none}
Specifies the Kerberos key table that is used by the LDAP server. The key table is used to obtain the encryption key for the Kerberos principal associated with the LDAP server. A key table must be provided if Kerberos authentication is used and the Kerberos KDC is not running on the same system as the LDAP server. However, a key table is not necessary if the Kerberos KDC is running on the same system as the LDAP server, the user ID associated with the LDAP server has a RACF KERB segment containing the server principal name, and the user ID associated with the LDAP server has read permission to the IRR.RUSERMAP facility class. In this case, the `krbKeytab` option should either be omitted or should be set to `none`. Following is an example:

```
krbKeytab /home/users/u1/keytab
```

Default = no value

**krbLDAPAdmin**  
`kerberosIdentityDN`

Specifies the Kerberos identity that represents the LDAP administrator. This option allows the administrator to bind through Kerberos and still maintain administrative authority. The value for this option must be specified as a DN with the attribute type of `ibm-kn`. The `ibm-kn` attribute type is case-sensitive and must match the actual Kerberos identity. Following is an example:

```
krbLDAPAdmin ibm-kn=LDAPAdmin@myrealm.com
```

For information about specifying a value for a distinguished name for this option, see Specifying a value for a distinguished name.

**listen** `ldap_URL`

Specifies, in LDAP URL format, the IP address (or host name) and the port number where the LDAP server will listen to incoming client requests. This option may be specified more than once in the configuration file.

Note that the `listen` value may be established in the configuration file, or it may be established using the `-l` command-line parameter when starting the LDAP server (see Setting up and running the LDAP server).

Default = The server listens on all available IPv4 addresses (`INADDR_ANY`), using port 389. This is equivalent to `ldap://:389`.

The format of `ldap_URL` for the `listen` option to listen on a TCP/IP socket interface is the following. This format is also used for other configuration options whose value is in LDAP URL format, such as `altServer`, `masterServer`, and `referral`.

```
{ldap:// | ldaps://}[IP_address | hostname][:portNumber]
```

The format of `ldap_URL` for the `listen` option to listen on the Program Call interface is the following:

```
ldap://:pc
```

where:

`ldap://` Specifies that the server listen on nonsecure addresses or ports. Note that if SSL/TLS is
configured for the server, then once a connection is established, the client may switch to secure communication using the Start TLS extended operation.

\texttt{ldaps://}

Specifies that the server listen on secure addresses or ports. When a connection is established to the server, the client must begin the SSL/TLS handshake protocol. The \texttt{sslKeyRingFile} option must also be specified when using this format.

\textbf{IP\_address}

Specifies either the IPv4 or IPv6 address.

\textbf{hostname}

Specifies the host name. If the host name is used for the \texttt{listen} option, all of the IPv4 or IPv6 addresses associated with the \texttt{hostname} are obtained from the DNS (Domain Name Server) and the LDAP server listens on each of these IP addresses.

\textbf{portNumber}

Specifies the port number. The \texttt{portNumber} is optional. If the port number is not specified for an \texttt{ldap://}, then the default of 389 is used for nonsecure connections. If the port number is not specified for an \texttt{ldaps://}, then the default of 636 is used for secure connections.

Range = 1 - 65536

If the \texttt{serverSysplexGroup} option is present in the configuration file, the port number specified for this server instance must be the same as the port number specified for all other members of the sysplex group for dynamic workload balancing to function properly.

It is advisable to reserve the port number or numbers chosen here in your TCP/IP profile data set. Also, be aware that port numbers below 1024 may require additional specifications. Consult \textit{z/OS Communications Server: IP Configuration Reference} for more information.

\textbf{pc}

Specifies that the LDAP server should listen for program call (PC) calls from Policy Director or RACF change logging using the z/OS Security Authorization Facility (SAF) interface. Only one LDAP server on a system can listen for PC calls.

Note that when the \texttt{listen} option is initialized to listen for PC calls on the LDAP server, the \texttt{listen} parameter must not include an IP address or a host name and you cannot specify \texttt{ldaps}.

Following are some examples of how you can specify \texttt{ldap\_URL}.

- If you specify:
  \texttt{ldap://}

  the LDAP server binds and listens on all available IPv4 addresses (INADDR\_ANY) on the system on the nonsecure default port of 389.

- If you specify:
  \texttt{ldap://us.endicott.ibm.com:489}

  the LDAP server binds and listens on all of the IPv4 and IPv6 addresses associated with host name \texttt{us.endicott.ibm.com} on the nonsecure port of 489 for incoming client requests.

- If you specify:
  \texttt{ldap://9.130.77.27}

  the LDAP server binds and listens on IPv4 address 9.130.77.27 on the default nonsecure port of 389 for incoming client requests.

- If you specify:
  \texttt{ldaps://us.endicott.ibm.com}
the LDAP server binds and listens for incoming client requests on the IPv4 and IPv6 addresses associated with host name us.endicott.ibm.com on the default secure port of 636.

- If you specify:
  ldaps://9.130.77.27:736

the LDAP server binds and listens on IPv4 address 9.130.77.27 on the secure port of 736.

- If you specify:
  ldap://:489

the LDAP server binds and listens on all available IPv4 addresses (INADDR_ANY) on the system on the nonsecure port of 489 for incoming client requests

- If you specify:
  ldaps://:777

the LDAP server binds and listens on all available IPv4 addresses (INADDR_ANY) on the system on the secure port of 777 for incoming client requests.

- If you specify:
  ldaps://[5f1b:df00:ce3e:e200:20:800:2078:e3e3]:389

the LDAP server binds and listens on the IPv6 address 5f1b:df00:ce3e:e200:20:800:2078:e3e3 on the system on the nonsecure port of 389 for incoming client requests.

- If you specify:
  ldaps://[::ffff:9.130.77.75]:777

the LDAP directory server binds and listens on the IPv4 mapped IPv6 address of ::ffff:9.130.77.75 on the system on the secure port of 777 for incoming client requests.

- If you specify:
  ldap://[:]

the LDAP server binds and listens on all available IPv4 (INADDR_ANY) and IPv6 (in6addr_any) addresses on the nonsecure default port of 389.

- If you specify:
  ldap://:pc

the LDAP server binds and listens for PC calls from Policy Director and from RACF change logging using the SAF interface into the server.

**Note:** The listen parameter deprecates the security, port, and securePort options in the configuration file. If there is a listen option specified in the configuration file along with either security, port, or securePort, the listen option takes precedence over what has been specified for security, port, or securePort. If using an earlier version of the configuration file with security, port, or securePort, the LDAP server is configured to listen on the port numbers specified for securePort, port, or both depending upon the security setting. However, it is highly recommended that the LDAP server be configured using the listen option.

### logfile filename

<table>
<thead>
<tr>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Specifies where to place the activity log records when activity logging is enabled. See Activity logging for more information.
See [Specifying a value for filename](#) for information about specifying the filename.

Default = /etc/ldap/gldlog.output

**masterServer ldap_URL**

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

Specifies for this backend the location of this replica's master server for basic replication. There is no required format for the value, however the z/OS LDAP client can only follow a `masterServer` value if it is in LDAP URL format. See page 86 for a description of LDAP URL format. The presence of this option indicates that this LDAP server is a basic replication read-only replica for this backend and receives updates from a master LDAP server. Any other update requests for this backend received directly by the this LDAP server is redirected to the master server. You must also specify the `masterServerDN` option in this section of the configuration file. The master server must contain all of the suffixes defined for this backend.

The `masterServer` option can be specified multiple times if there are multiple master servers. In this case, the LDAP client will attempt to contact each server in the list until it is able to establish a connection with one of the servers.

The `masterServer` option indicates basic replication is configured for this backend section. Therefore, the `masterServer` configuration option cannot be specified if the `useAdvancedReplication` configuration option is set to `on` in the CDBM backend database section.

In the following example, `myldap.server.com` is the host name and 3389 is the port number of the LDAP URL:

```
masterServer ldap://myldap.server.com:3389
```

In the following example, the IPv6 address of `5f1b:df00:ce3e:e200:20:800:2078:e3e3` is the IP address and 389 is the port number of the LDAP URL.

```
masterServer ldap://[5f1b:df00:ce3e:e200:20:800:2078:e3e3]:389
```

**masterServerDN dn**

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

Specifies the distinguished name (DN) always allowed to make changes to this basic replication read-only replica backend. The value must be in DN format that is described in Chapter 13, "Data model." The presence of this option indicates that this LDAP server is a read-only replica for this backend and receives updates from a master LDAP server using the specified DN. The specified DN is a special entry that should only be used when replicating to this read-only replica backend. The DN has unrestricted update, compare, and search access for all entries in the backend on this server, even if the LDAP server is in maintenance mode. When in maintenance mode, only this DN and the LDAP administrator can access and update the entries in this backend. All other update operations for this backend received by the replica server are redirected to the master server. Care must be taken when updating this backend to ensure the replica server remains synchronized with the master server.

You must also specify the `masterServer` option in this section of the configuration file. You cannot specify the `peerServerDN` option.

The `masterServerDN` option indicates basic replication is configured for this backend section. Therefore, the `masterServerDN` configuration option cannot be specified if the `useAdvancedReplication` configuration option is set to `on` in the CDBM backend database section.
It is recommended, though not required, that the DN have the same suffix as one of the suffix option values in the configuration file. Establishing the administrator DN and basic replication replica server DN and passwords describes how to set up your master server DN.

For information about specifying a value for a distinguished name for this option, see Specifying a value for a distinguished name.

```
masterServerPW string
```

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Specifies the password for the masterServerDN that is allowed to make updates for this backend. This option is only applicable for a basic replication read-only LDAP server. See Establishing the administrator DN and basic replication replica server DN and passwords for additional information about the master server password.

Note: Use of the masterServerPW configuration option is strongly discouraged in production environments. Instead, specify your masterServerDN as the distinguished name of an existing entry in the directory information tree, including a userPassword attribute. This will eliminate passwords from the configuration file.

The masterServerPW option indicates basic replication is configured for this backend section. Therefore, the masterServerPW configuration option cannot be specified if the useAdvancedReplication configuration option is set to on in the CDBM backend database section.

```
maxConnections num-connections
```

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Specifies the maximum number of concurrently connected clients that the LDAP server allows.

Range = 30 to 65535
Default = operating system maximum

The LDAP server limits the number of client connections by restricting the number of file and socket descriptors used by the LDAP server. Some of the descriptors are used by the LDAP server for its own file descriptors and passive socket descriptors. The value specified for this option should take into account that the server uses approximately 10 descriptors for internal functions and will use more depending upon the number of additional sockets used as passive sockets for connection attempts by clients.

The maximum number of client connections is further restricted by:

- The maximum number of files a single process can have concurrently active.

  The MAXFILEPROC statement for BPXPRMxx and the FILEPROCMax option on the RACF altuser command are used to set the limit. Only processes with superuser authority can adjust the limit beyond the limit specified by MAXFILEPROC and FILEPROCMax. Attempts to exceed this limit by non-superuser processes may be audited by the security manager.

- The maximum number of sockets allowed by the TCP/IP socket file system.

  The MAXSOCKETS option on the NETWORK statement for BPXPRMxx sets this limit.

Setting these limits too high can affect system performance by using too many resources and deprive other functions of their share of the same resources.

```
multiserver {on | off}
```
Indicates the operating mode that the LDAP server will run for this backend. Specifying on indicates the server runs in multi-server mode for this backend (see Determining operational mode). In multi-server mode, the LDAP server shares directory data with other instances of the LDAP server running within the sysplex. The serverSyplexGroup configuration option must also be specified when running in multi-server mode. Specifying off indicates the server runs in single-server mode for this backend.

Default = off

You can configure a backend to operate in single-server mode while another backend operates in multi-server mode except when GDBM or CDBM is configured. When CDBM or GDBM is configured, all TDBM, LDBM, GDBM, and CDBM backends must be configured to use the same operating mode.

nativeAuthSubtree {all | dn}

Specifies the distinguished name of a subtree where all of its entries are eligible to participate in native authentication. This option can appear multiple times to specify all subtrees that use native authentication. If this option is omitted or is set to all, then the entire directory is subject to native authentication. This option is ignored if useNativeAuth selected or all is not specified.

For information about specifying a value for a distinguished name for this option, see Specifying a value for a distinguished name.

Default = all

nativeUpdateAllowed {on | off}

Enables native password or password phrase changes in the Security Server to occur through a modify request to the TDBM, LDBM, or CDBM backend if the useNativeAuth selected or all option is specified. This option does not affect the ability to change a native password or password phrase during a bind operation.

Default = off

operationsMonitor {ip | ipAny | all}

Specifies the search patterns monitored by the LDAP server. Operations monitor supports two types of search patterns, that are searchStats and searchIPStats. A searchStats pattern consists of the search parameters (search base, scope, filter, and attributes to be returned) and status (SUCCESS or FAILURE). A searchIPStats pattern consists of the same elements as in the searchStats pattern, but also includes the client IP address. If operations monitor is enabled, LDAP monitors search statistics for the types of search patterns that are configured. See “Operations monitor” on page 489 for more information about operations monitor.
If set to ip, then only searchIPStats patterns are monitored. This option setting is useful in determining if there are any specific clients spamming the LDAP server.

If set to ipAny, then only searchStats patterns are monitored. This option is useful for evaluating the performance of search patterns.

If set to all, the operations monitor will monitor both searchStats and searchIPStats patterns. Therefore, each search is included in two search patterns, one matching the searchStats pattern and one matching the searchIPStats pattern.

Default = ipAny

operationsMonitorSize num-entries

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Specifies the maximum number of search patterns for which the operations monitor gathers statistics. The value must be between 0 and 2147483647. A value of 0 indicates that the operations monitor is turned off. When the operations monitor is turned off, the cn=operations,cn=monitor entry is not returned on a cn=monitor search.

Default = 1000

pcIdleConnectionTimeout num-seconds

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Specifies the amount of time in seconds that an idle connection will remain valid over the LDAP PC (program call) callable interface. After the specified time, the PC connection is considered no longer in use and any resources associated with the connection are released. Idle connections are detected when the LDAP server receives a new PC connection or a request on an existing PC connection.

The value must be either 0 or between 30 and 2147483647. A value of 0 indicates that an idle connection will remain indefinitely.

Default = 0 (indefinitely)

Recommended value = 0

pcThreads num-threads

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Specifies the number of threads to be initialized to handle incoming program call (PC) calls using the z/OS SAF interface into the LDAP server. No threads are used if the program call interface is not active. The value must be in the range of 2 to 2147483647.

Default = 10

peerServerDN dn

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Specifies the distinguished name (DN) always allowed to make changes to this basic replication peer replica backend. The value must be in DN format that is described in Chapter 13, “Data”
The presence of this option indicates that this LDAP server is a peer replica for this backend, and can receive updates from another peer LDAP server using the specified DN and processing updates received from clients. The specified DN is a special entry that should only be used when replicating to this peer replica backend. The DN has unrestricted update, compare, and search access for all entries in the backend on this server, even if the LDAP server is in maintenance mode. When in maintenance mode, only this DN and the LDAP administrator can access and update the entries in this backend.

Update operations for this backend received from you bound as peerServerDN (or as adminDN when in maintenance mode) are performed on the local database and are not sent to any peer and read-only replica servers. When not in maintenance mode, all other update operations for this backend are performed on the local database and are sent to the other peer and read-only replica servers. Update operations from a peer or a master are never replicated. It does not matter if you are in maintenance mode or not. Updates made by the adminDN are replicated unless the server is in maintenance mode.

You cannot also specify the masterServerDN option in this section of the configuration file.

The peerServerDN option indicates basic peer-to-peer replication is configured for this backend section. Therefore, the peerServerDN configuration option cannot be specified if the useAdvancedReplication configuration option is set to on in the CDBM backend database section.

It is recommended, though not required, that the DN have the same suffix as one of the suffix option values in the configuration file. Establishing the administrator DN and basic replication replica server DN and passwords describes how to set up your peer replica DN.

For information about specifying a value for a distinguished name for this option, see Specifying a value for a distinguished name.

peerServerPW string

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

Specifies the password for the peerServerDN that is allowed to make updates for this backend. This option is only applicable for a basic replication peer replica LDAP server. See Establishing the administrator DN and basic replication replica server DN and passwords for additional information about the peer server password.

Note: Use of the peerServerPW configuration option is strongly discouraged in production environments. Instead, specify your peerServerDN as the distinguished name of an existing entry in the directory information tree, including a userPassword attribute. This will eliminate passwords from the configuration file.

The peerServerPW option indicates basic peer-to-peer replication is configured for this backend section. Therefore, the peerServerPW configuration option cannot be specified if the useAdvancedReplication configuration option is set to on in the CDBM backend database section.

persistentSearch {on | off}

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

Allows or disallows persistent search for changes made to entries in a backend. When off is specified, persistent search requests for this backend are rejected. See PersistentSearch for more information about persistent search.
Default = off

```plaintext
plugin pluginType pluginName pluginInit [pluginParameters]
```

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Defines a plug-in extension to the LDAP server. Writing an LDAP server plug-in and using the SLAPI service routines are described in IBM Tivoli Directory Server Plug-in Reference for z/OS. A sample plug-in and its makefile are included in /usr/lpp/ldap/examples. Building and using the sample plug-in are described in the IBM Tivoli Directory Server Plug-in Reference for z/OS.

- For `pluginType`:
  - Specify `preOperation`, `clientOperation` or `postOperation`. A `preOperation` plug-in is called by the LDAP server before a client request is processed. A `clientOperation` plug-in is called to process a client request. A `postOperation` plug-in is called after a client request is processed. A `clientOperation` plug-in is called when a client request matches a distinguished name suffix or extended operation object identifier registered for the plug-in.

- For `pluginName`:
  - Specify the name of the shared library (DLL) containing the plug-in code. A plug-in that supports both 31-bit and 64-bit addressing modes should specify both file names separated by a slash, "/", such as plugin31/plugin64. A plug-in that supports only 31-bit addressing mode should only specify one file name, such as plugin31.

- For `pluginInit`:
  - Specify the name of the plug-in initialization routine. This plug-in routine is called by the LDAP server to allow the plug-in to initialize. The plug-in initialization routine will register supported message types, distinguished name suffixes and extended operation object identifiers supported by the plug-in.

- For `pluginParameters`:
  - Optionally, specify plug-in parameters. The plug-in can retrieve these parameters using the `slapi_pblock_get()` routine.

```plaintext
port num-port
```

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** The `port` option has been deprecated by the `listen` option. See page 88 for information about the `listen` option.

Specifies the TCP/IP port used by the LDAP server for non-SSL communications. The value must be in the range of 1 to 65535.

Default = 389

If the `serverSysplexGroup` option is present in the configuration file, the port number specified for this server instance must be the same as the port number specified for all other members of the sysplex group for dynamic workload balancing to function properly.

The port number may be established in the configuration file, or it may be established using the `-p` command-line parameter when starting the LDAP server (see Setting up and running the LDAP server).

It is advisable to reserve the port number chosen here in your TCP/IP profile data set. Also, be aware that port numbers below 1024 may require additional specifications. See z/OS Communications Server: IP Configuration Guide for further information.
pwCryptCompat {on | off}

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Specifies whether to use an EBCDIC version or a UTF-8 version of the crypt() algorithm to encrypt passwords when `pwEncryption crypt` is contained in this section of the configuration file. If `on`, the EBCDIC version of the crypt() algorithm is used. This is what the z/OS Integrated Security Services LDAP server used. If `off`, the UTF-8 version is used. Note that ASCII is a subset of UTF-8. When sharing LDAP directory data between z/OS and an ASCII-based platform, specify `pwCryptCompat off` to ensure that the encrypted value is the same on both platforms.

Default = `on`

pwEncryption {none | crypt | MD5 | SHA | DES: `keylabel` | AES: `keylabel`}

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Specifies what encryption method to use when storing the `userPassword` attribute values in the backend of the directory.

- **none** Specifies no encryption. The `userPassword` attribute values are stored in clear text format. The stored values are prefixed with the tag (none). The original value, without the tag, is returned for a search request.
- **crypt** Specifies that `userPassword` attribute values are encoded by the crypt() algorithm before they are stored in the directory. The stored values are prefixed with the tag (crypt). There are two version of the crypt() algorithm: an EBCDIC-based version and a UTF-8-based version. See the `pwCryptCompat` option and the notes below for information about selecting which version to use. The original password value cannot be retrieved in clear text format. The tag and the encrypted value are returned for a search request.
- **MD5** Specifies that `userPassword` attribute values are encoded by the MD5 encrypt algorithm before they are stored in the directory. The stored values are prefixed with the tag (MD5). The original password value cannot be retrieved in clear text format. The tag and the encrypted value are returned for a search request.
- **SHA** Specifies that `userPassword` attribute values are encoded by the SHA encrypt algorithm before they are stored in the directory. The stored values are prefixed with the tag (SHA). The original password value cannot be retrieved in clear text format. The tag and the encrypted value are returned for a search request.
- **DES: `keylabel`** Specifies that `userPassword` attributes values are encrypted by the DES algorithm before they are stored in the directory. The stored values are prefixed with the tag '{DES: `keylabel`}'. The original password value, without the tag, is returned for a search request. The key label must refer to either a valid data-encrypting key generated by the KGUP utility and stored in the ICSF CKDS or to an entry in the dataset referenced by the LDAPKEYS DD statement. See [Symmetric encryption keys](#) for more information.

- **AES: `keylabel`** Specifies that `userPassword` attribute values are encrypted by the AES algorithm using the specified key label before they are stored in the directory. The stored values are prefixed with the tag (AES: `keylabel`). The original password value without the tag is returned for a search request. The key label must refer to either a valid data-encrypting
key generated by the KGUP utility and stored in the ICSF CKDS or to an entry in the dataset referenced by the LDAPKEYS DD statement. See Symmetric encryption keys for more information.

Notes:
1. When a password is stored in a TDBM, LDBM, or CDBM backend, it is prefixed with the appropriate encryption tag so that when a clear text password is sent on an LDAP API simple bind it can be encrypted in that same method for password verification.
2. The crypt algorithm, implemented across many platforms, accepts only the first eight characters of a password. As a result, any password supplied on a bind or compare operation that matches the first eight characters of a userPassword attribute value encrypted with the crypt algorithm in the directory will match.
3. When the pwCryptCompat option is set to on, the values encrypted using the crypt algorithm are not portable to other X/Open-conformant systems if the userPassword values are unloaded using the ds2ldif utility with the -t command-line parameter and loaded by another platform’s load utility. If the pwCryptCompat option is set to off, the values encrypted using the crypt algorithm are portable to other X/Open-conformant systems if the userPassword values are unloaded using the ds2ldif utility with the -t command-line parameter. The output LDIF file from ds2ldif can then be loaded using another platform’s load utility.
4. If a tagged encrypted userPassword attribute value is included in an add or modify operation, the attribute value is added as it is with no additional encryption performed on the value even if the pwEncryption option is set to a different type of encryption.

Default = none

pwSearchOutput {binary | base64}

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Specifies the format of MD5 and SHA encrypted userPassword attribute values when retrieved on a search operation.

If set to binary and a userPassword attribute value is encrypted in MD5 or SHA, the LDAP server returns the encryption tag (either {MD5} or {SHA}) in UTF-8 followed by the encrypted binary hash.

If set to base64 and a userPassword attribute value is encrypted in MD5 or SHA, the LDAP server returns the encryption tag (either {MD5} or {SHA}) in UTF-8 followed by the base64-encoded encrypted binary hash.

For an example of using this option, see One-way encryption formats.

Default = binary

readOnly {on | off}

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Specifies the ability to modify the database. The LDAP server BACKEND operator modify command can be used to change the backend database to read-write or read-only mode while the LDAP server is running. Any attempt to use the LDAP server to modify the database will fail if readOnly is turned on.

Notes:
1. For GDBM, change log entries are not created and are not trimmed (deleted) by the LDAP server when readOnly is on.
2. When running in multi-server mode, the **readOnly** configuration option should be the same for all LDAP servers in the cross-system group because any LDAP server can potentially handle update requests.

3. For SDBM, **readonly on** does not prevent changing a RACF password during a bind operation, using the `currentValue/newValue` format. However, it does prevent changing the password by using a modify operation of the **racfpassword** attribute.

4. Also, when LDBM, TDBM, or CDBM is using native authentication, the RACF password can be changed during bind even though **readonly on** is specified. The RACF password cannot be changed by using the LDBM, TDBM, or CDBM native authentication modify of the **userpassword** attribute.

Default = off

**referral** *ldap_URL*

<table>
<thead>
<tr>
<th>Schema</th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Specifies the referral to pass back when the target of a client request is not included in any suffix within the LDAP server. It is also known as the default referral. The **referral** option can appear multiple times and should list equivalent servers. There is no required format for the value, however the z/OS LDAP client can only follow a referral value if it is in LDAP URL format. See page [86](#) for a description of LDAP URL format.

A default referral is not returned to the client if the client request includes the **manageDsaiT** control. See [manageDsaiT](#) for more information about this control.

In the following example, `myldap.server.com` is the host name and 3389 is the port number of the LDAP directory URL:

```
referral ldap://myldap.server.com:3389
```

In the following example, the IPv6 address `5f1b:df00:ce3e:e200:20:800:2078:e3e3` is the IP address and 389 is the port number of the LDAP URL:

```
referral ldap://[5f1b:df00:ce3e:e200:20:800:2078:e3e3]:389
```

**schemaPath** *name*

<table>
<thead>
<tr>
<th>Schema</th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Specifies the name of the file directory containing the LDAP schema database and checkpoint files. A fully-qualified directory path must be specified. When multi-server mode is active, the same schema path must be specified for each LDAP server within the cross-system group. The schema database file is automatically created during LDAP server initialization if it does not already exist. The LDAP server must have write access to the schema directory. This configuration option also determines the directory used by CDBM to store its data if the CDBM backend is configured and the **databaseDirectory** configuration option is not specified in the CDBM backend configuration section.

Default = `/var/ldap/schema`

**schemaReplaceByValue** {on | off}

<table>
<thead>
<tr>
<th>Schema</th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Determines the behavior of modify operations with replace values of the schema entry. When **schemaReplaceByValue off** is specified, a modify operation with replace values for an attribute in
the schema entry behaves like a typical modify operation: all the values currently in the attribute are replaced by the values specified in the modify operation. When `schemaReplaceByValue` is specified, individual values in an attribute in the schema entry can be replaced without removing all the other values currently in the attribute. Except in several specific cases, the values of the attribute that are in the initial LDAP server schema cannot be changed or removed. See [Updating the schema](#) for more information about modifying the schema.

The `schemaReplaceByValue` configuration option can be overridden on a specific modify operation by including the `schemaReplaceByValueControl` control in the modify request.

Default = `on`

```plaintext
secretEncryption {none | DES: keylabel | AES: keylabel}
```

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Specifies the encryption method to use when storing the `secretKey`, `replicaCredentials`, `ibm-replicaKeyPwd`, and `ibm-slapdMasterPwd` attribute values in this backend. Applications may use the `secretKey` attribute type to store sensitive data that needs to be encrypted in the directory and to retrieve the data in clear text format. This encryption method is used to protect the `replicaCredentials` attribute values in this backend when basic replication is enabled. This encryption method also protects the `ibm-replicaKeyPwd` and `ibm-slapdMasterPwd` attribute values in this backend when advanced replication is enabled.

- **none** Specifies no encryption. The `secretKey`, `replicaCredentials`, `ibm-replicaKeyPwd`, and `ibm-slapdMasterPwd` attribute value is stored in clear text format. The stored value is prefixed with the tag (none). This is the default if the `secretEncryption` option is not specified. The attribute value without the tag is returned for a search request.

- **DES: keylabel**
  The `secretKey`, `replicaCredentials`, `ibm-replicaKeyPwd`, and `ibm-slapdMasterPwd` attribute value is encrypted by the DES algorithm before it is stored in the directory. The stored value is prefixed with the tag (DES:keylabel). The original value without the tag is returned for a search request. The key label must refer to either a valid data-encrypting key generated by the KGUP utility and stored in the ICSF CKDS or to an entry in the dataset referenced by the LDAPKEYS DD statement. See [Symmetric encryption keys](#) for more information.

- **AES: keylabel**
  The `secretKey`, `replicaCredentials`, `ibm-replicaKeyPwd`, and `ibm-slapdMasterPwd` attribute value is encrypted by the AES algorithm before it is stored in the directory. The stored value is prefixed with the tag (AES:keylabel). The original value without the tag is returned for a search request. The key label must refer to either a valid data-encrypting key generated by the KGUP utility and stored in the ICSF CKDS or to an entry in the dataset referenced by the LDAPKEYS DD statement. See [Symmetric encryption keys](#) for more information.

Default = `none`

```plaintext
securePort num-port
```

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** The `securePort` option has been deprecated by the `listen` option. See page 86 for information about the `listen` option.
Specifies the TCP/IP port used by the LDAP server for SSL communications. The value must be in the range of 1 to 65535.

Default = 636

If the serverSysplexGroup option is present in the configuration file, the secure port number specified for this server instance must be the same as the secure port number specified for all other members of the sysplex group for dynamic workload balancing to function properly.

The secure port number may be established in the configuration file, or it may be established using the -s command-line parameter when starting the LDAP server (see Setting up and running the LDAP server).

It is advisable to reserve the port number chosen here in your TCP/IP profile data set. Also, be aware that port numbers below 1024 may require additional specifications. See z/OS Communications Server: IP Configuration Guide for further information.

<table>
<thead>
<tr>
<th>security (ssl</th>
<th>sslonly</th>
<th>none</th>
<th>nossl)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Global</td>
<td>TDBM</td>
<td>LDBM</td>
</tr>
<tr>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** The security option has been deprecated by the listen option. See page 86 for information about the listen option.

Specifies what type of communications is accepted by the LDAP server. The ssl setting indicates that the server will listen on the secure port and the non-secure port. The sslonly setting means that the server will listen only on the secure port. The none or nossl settings indicate that the server will listen only on the non-secure port. The sslKeyRingFile option must also be specified when the ssl or sslonly settings are used.

Default = none

<table>
<thead>
<tr>
<th>securityLabel (on</th>
<th>off)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Global</td>
</tr>
<tr>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

Determines if the security label processing is activated with bound LDAP clients. When on, the security labels associated with the LDAP client and LDAP server are verified during the authentication process. Security labels are recorded in all LDAP audit records. When off, no security label processing is done.

Default = off

Use this option when configuring the LDAP server in a multilevel security environment. For more information about configuring a z/OS system for multilevel security and how to configure an LDAP server in that environment, see z/OS Planning for Multilevel Security and the Common Criteria.

| sendV3stringsoverV2as {UTF-8 | ISO8859-1} |
|----------|----------|------|------|------|------|
|          | Global   | TDBM | LDBM  | SDBM  | GDBM  | CDBM  | EXOP |
|          | X        |      |       |       |       |       |      |

Specifies the output data format to use when sending UTF-8 information over the LDAP Version 2 protocol.

Default = UTF-8
serverCompatLevel {3 | 4 | 5}

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Specifies the server compatibility level. This value can be used to limit the functions supported by the server so that the server can be compatible with older versions of LDAP servers when they are sharing directory data in a sysplex group. To produce consistent results, all the LDAP servers in the same sysplex group name must support the same functions. If fallback is required to a lower server compatibility level than is currently being used, it is necessary to remove all exploitation of function that is available at the current compatibility level but not at the lower level. The server may not start at the lower level until this is done.

The serverCompatLevel values are:

- **3** - This value limits the sharing of data in a sysplex to TDBM backends, DB2-based GDBM backends, and schema. Basic replication is supported from (but not into) the sysplex. Dynamic and nested groups are supported, as is schema replace by value. Specify this value when a z/OS Integrated Security Services (ISS) LDAP server is running in the sysplex.
- **4** - This value enables cross-system coupling facility (XCF) messaging support for TDBM and DB2-based GDBM backends in the sysplex group and supports basic replication from and into the sysplex.

**Note:** When the Schema, LDBM, and file-based GDBM backends are shared in a sysplex, XCF messaging is used to communicate between the LDAP servers in the same sysplex group no matter the serverCompatLevel setting.

Advanced replication and the CDBM backend are not supported. Specify this value when the sysplex group contains a z/OS IBM TDS server running on z/OS V1R10 or earlier and there are no ISS LDAP servers in the sysplex.

- **5** - This value enables advanced replication and allows the CDBM backend to be configured. Schema and all backends can be shared in the sysplex. Specify this value when the sysplex group only contains a z/OS IBM TDS LDAP servers running on z/OS V1R11 or later. Specify this value when the sysplex group only contains z/OS IBM TDS servers running on z/OS V1R11 or later.

Default = **5** if not running in a sysplex (the serverSysplexGroup configuration option is not specified)

Default = **4** if running in a sysplex (the serverSysplexGroup configuration option is specified)

serverEtherAddr mac_address

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Specifies the Media Access Control (MAC) address used for entry UUID generation. This value must be unique for all LDAP servers in your enterprise. You must specify the MAC address if multiple LDAP servers will run on a (hardware) system. This applies if your LDAP servers are on different LPARs and also if two LDAP servers are on the same LPAR. You do not need to specify this field if this is the only LDAP server that will run on this (hardware) system.

The MAC address consists of 12 hexadecimal digits. The suggested form of the mac_address is:

```
4xaaaaaaaaaaaaaa
```

Where:
Is a one-character LDAP directory number. If more than one LDAP server is operating on a processor, specify a different \textit{x} value for each server. If more than 16 LDAP servers are wanted, then use a serial number and model number from a processor that is not running an LDAP server. If another processor is not available, then set the \textit{x}, \textit{mmmm}, and \textit{ssssss} values from the MAC address on an old Ethernet card that is no longer being used or not used to run an LDAP server.

\textit{mmmm} is the four-digit model number of the processor.

\textit{ssssss} is the six-digit serial number of the processor.

It is not necessary to follow this convention if you will specify the \texttt{serverEtherAddr} option for all LDAP servers in your enterprise. In this case, you can specify any combination of 12 hexadecimal digits if each LDAP server has a unique value.

Following is an example:

\texttt{serverEtherAddr 4A123401234D}

Default = The LDAP server uses the hardware model and serial numbers to generate a MAC address.

\texttt{serverKrbPrinc kerberosIdentity}

<table>
<thead>
<tr>
<th>\textbf{Global}</th>
<th>\textbf{TDBM}</th>
<th>\textbf{LDBM}</th>
<th>\textbf{SDBM}</th>
<th>\textbf{GDBM}</th>
<th>\textbf{CDBM}</th>
<th>\textbf{EXOP}</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{X}</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Specifies the Kerberos principal name assigned to the LDAP server that was created in \texttt{Defining the Kerberos identity}. This value becomes the server name in Kerberos service tickets. The principal name must consist of characters that can be represented in the ISO8859-1 code page. The format for \texttt{kerberosIdentity} is:

\texttt{ldap_prefix/primary-dns-name@krbRealmName}

Where

\texttt{ldap_prefix}

Is \texttt{ldap} or LDAP. Use \texttt{ldap} to assure interoperability with all LDAP clients. LDAP is accepted, but this value is not usable with many non-z/OS LDAP clients.

\texttt{primary-dns-name}

Is the canonical host name returned by the DNS name service.

\texttt{krbRealmName}

Is the Kerberos defined realm that the LDAP server operates. For additional information about setting up a Kerberos realm on z/OS, see \texttt{z/OS Integrated Security Services Network Authentication Service Administration}.

Following are examples:

\texttt{serverKrbPrinc LDAP/myhost.realm.com@myrealm.com}

\texttt{serverKrbPrinc ldap/myhost.myrealm.com@myrealm.com}

Default = \texttt{ldap/primary-dns-hostname@default-krbRealmName}

\texttt{serverName string}

<table>
<thead>
<tr>
<th>\textbf{Global}</th>
<th>\textbf{TDBM}</th>
<th>\textbf{LDBM}</th>
<th>\textbf{SDBM}</th>
<th>\textbf{GDBM}</th>
<th>\textbf{CDBM}</th>
<th>\textbf{EXOP}</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{X}</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Specifies the name of the DB2 server location that manages the tables for the LDAP server. This value must match the name of one of the DATA SOURCE stanzas that must be specified in the ODBC initialization data set that is specified by the \texttt{dsnaoini} option in the configuration file. See

Chapter 8. Customizing the LDAP server configuration 101
For a description of the DSNAOINI ODBC initialization data set contents. Using the example DSNAOINI file in Figure 2 the value of string for serverName is LOC1.

If the serverName configuration option is specified for a backend, the option must also be specified, with the same value, for all the TDBM and DB2-based GDBM backends in the configuration file.

Default = The default data source is used. This is the DB2 subsystem specified by the MVSDEFAULTSSID record in the DSNAOINI file.

serverSysplexGroup name

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>serverSysplexGroup</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Specifies that this LDAP server is participating in data sharing in a sysplex and indicates the name of the cross-system coupling facility (XCF) group. All LDAP servers in the sysplex that specify the same group name share the LDAP server schema and the directories of backends that specify the multiserver on option. The group name is 1-8 characters and consists of letters (A-Z), numbers (0-9), and special characters (@, #, and $). The special characters must be in the IBM-1047 code page.

sizeLimit num-limit

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>sizeLimit</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Specifies the maximum number of entries to return from a search operation. The maximum number can be modified on a specific search request as described below.

Range = 0 - 2147483647

0 = no limit

Default = 500

This option applies to all backends, except EXOP, unless specifically overridden in a backend definition. Specifying this before a database line in the configuration sets the option for all backends, except EXOP. Specifying it after a database line sets the option just for the backend defined by the database line.

A limit on the number of entries returned can also be specified by the client on a search request. Note that the following behavior is used when referring to the sizeLimit parameter.

When accessing the LDAP server using the TDBM, LDBM, CDBM, or GDBM backend:

- If the client has not bound as the adminDN, then the limit is the smaller of the limit passed by the client and the limit read by the server from the sizeLimit option in ds.conf (which defaults to 500). If the client does not specify a limit, then the server limit is used.
- If the client has bound as the adminDN, then the limit is the value passed by the client. If the client does not specify a limit, then the number of entries returned is unlimited. The limit from the ds.conf file is ignored when the client has bound as the adminDN.

When accessing the LDAP directory support for RACF (the SDBM backend):

- The limit is the smaller of the limit passed by the client and the limit read by the server from the sizeLimit option in ds.conf (which defaults to 500). If the client does not specify a limit, then the server limit is used. It does not matter how the client has bound.
- The number of entries returned may be further restricted by limits imposed by RACF. See Chapter 16, “Accessing RACF information” for more information.
There are additional considerations for size limit when performing a subtree search from the root DSE (a NULL-based search). See Root DSE search with subtree scope (Null-based subtree search) for more information.

**srvStartUpError {terminate | ignore}**

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Specifies whether the LDAP server stops if a backend or plug-in fails to initialize after the configuration file is read. If **terminate**, the server ends when any backend or plug-in fails to initialize. If **ignore**, the LDAP server continues processing if the schema successfully initializes.

The option also applies to failures when initializing the LDAP PC callable support interface if that has been configured and initializing WLM support. Note that a configuration error that occurs before backend or plug-in initialization begins always causes the server to end.

Default = **terminate**

**sslAuth {serverAuth | serverClientAuth}**

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Specifies the SSL/TLS authentication method. The **serverAuth** method allows the LDAP client to validate the LDAP server on the initial contact between the client and the server.

The **serverClientAuth** method allows the LDAP client to validate the LDAP server. In addition, the LDAP server validates the LDAP client if the client sends its digital certificate on the initial contact between the client and the server.

**Note:** In order for clients to perform SASL EXTERNAL binds to the LDAP server, it is necessary to configure the server with **sslAuth serverClientAuth**.

See Setting up for SSL/TLS for more SSL/TLS information.

Default = **serverAuth**

**sslCertificate {certificateLabel | none}**

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Specifies the label of the certificate that is used for LDAP server authentication. If using a key database file, the certificate is created and managed using the **gskkyman** utility. If using a RACF key ring, the certificate is created and managed using the **RACDCERT** command. If using a PKCS #11 token, the certificate can be created and managed by using either the **gskkyman** utility or the **RACDCERT** command. See z/OS Cryptographic Services System SSL Programming for details on using the **gskkyman** utility or z/OS Security Server RACF Command Language Reference for details on using the **RACDCERT** command. See Setting up for SSL/TLS for more SSL/TLS information.

Default = **none**

If the value is **none** (by default or by specification), the default certificate, marked in the key database file, the RACF key ring, or the PKCS #11 token, is used for server authentication.

**sslCipherSpecs {string | ANY}**
Specifies the SSL Version 3 and TLS Version 1 cipher specifications that the LDAP server accepts from clients. The cipher specification is a blank delimited string that represents an ORed bitmask indicating the SSL/TLS cipher specifications that is accepted from clients. Clients that support any of the specified cipher specifications is able to establish an SSL/TLS connection with the server. Table 8 shows a list of the cipher spec mask values and the related decimal, hexadecimal, and keyword values. See [z/OS Cryptographic Services System SSL Programming](https://www.ibm.com/support/docview.wss?uid=swg27037930) for a description of supported cipher specifications.

The cipher specification may be specified as follows:

- A decimal value (for example, 256)
- A hexadecimal value (for example, x100)
- A keyword (for example, TRIPLE_DES_SHA_US)
- A construct of those values using plus and minus signs to indicate inclusion or exclusion of a value. For example,
  - 256+512 is the same as specifying 768, or x100+x200, or TRIPLE_DES_SHA_US+DES_SHA_EXPORT
  - 52992 is the same as specifying ALL+RC2_MD5_EXPORT+RC4_MD5_EXPORT

Depending upon the level of System SSL support installed, some ciphers may not be supported. System SSL will ignore the unsupported ciphers. You should consult the System SSL documentation to determine the specific ciphers that your installation supports.

See [Setting up for SSL/TLS](https://www.ibm.com/support/docview.wss?uid=swg27037930) for more SSL/TLS information.

Default = ANY

**sslKeyRingFile name**

Specifies the path and file name of the SSL/TLS key database file, the name of the RACF key ring, or the name of the PKCS #11 token to be used by the LDAP server. SSL/TLS connections will not be available if this option is not specified.

When using a key database file, the file path and name specified here must match the path and name of the key database file that was created using the **gskkyman** utility (see [z/OS Cryptographic Services System SSL Programming](https://www.ibm.com/support/docview.wss?uid=swg27037930)). Also, see [Setting up for SSL/TLS](https://www.ibm.com/support/docview.wss?uid=swg27037930) for more SSL/TLS information.

The LDAP server supports the use of a RACF key ring. Specify the RACF key ring name for the **sslKeyRingFile** and comment out the **sslKeyRingFilePW** and **sslKeyRingPWStashFile** configuration options to use this support.

The LDAP server also supports the use of a PKCS #11 token. Specify the PKCS #11 token on the **sslKeyRingFile** configuration option in the following format (where NAME is the name of the PKCS #11 token): +TOKEN+/NAME. Ensure that the **sslKeyRingFilePW** and **sslKeyRingPWStashFile** configuration options are commented out to use this support.

See "Creating and using key databases, key rings, or PKCS #11 tokens" on page 57 for more information.

**sslKeyRingFilePW string**
Specifies the password protecting access to the SSL/TLS key database file. The password string must match the password to the key database file that was created using the `gskkyman` utility (see `z/OS Cryptographic Services System SSL Programming`). Also, see Setting up for SSL/TLS for more SSL/TLS information.

**Note:** Use of the `sslKeyRingFilePW` configuration option is strongly discouraged. As an alternative, use either a RACF key ring, a PKCS #11 token, or specify the `sslKeyRingPWStashFile` configuration option.

Comment out the `sslKeyRingFilePW` and `sslKeyRingPWStashFile` configuration options if you are using a RACF key ring or PKCS #11 token.

### sslKeyRingPWStashFile name

Specifies a file system file name where the password for the server’s key database file is stashed. Use the full path name of the stash file in the file system for `name`.

If this option is present, then the password from this stash file overrides the `sslKeyRingFilePW` configuration option, if present. Use the `gskkyman` utility with the `-s` option to create a key database password stash file. See Setting up for SSL/TLS for more SSL/TLS information.

Comment out the `sslKeyRingFilePW` and `sslKeyRingPWStashFile` configuration options if you are using a RACF key ring or PKCS #11 token.

### sslMapCertificate {off | check | add | replace} {fail | ignore}

Specifies the server maps a certificate used in a `SASL EXTERNAL` bind to the RACF user that is associated with the certificate.

When `check`, `add`, or `replace` is specified for the first value, RACF is searched for the user ID associated with the certificate used during a SASL certificate bind. The `sslKeyRingFile` configuration option must be specified to indicate which key database, RACF key ring, or PKCS #11 token to use to do this. If there is no RACF user ID associated with the certificate and `fail` is specified for the second value, then the `SASL EXTERNAL` bind fails. If there is no associated RACF user ID and `ignore` is specified for the second value, the bind continues without mapping the certificate to a RACF user.

If an associated RACF user ID is found and `add` or `replace` is specified for the first value, a distinguished name (DN) is created based on the user ID and the SDBM suffix. For `add`, this mapped DN is added to the list of DNs associated with the bind DN that was created from the subject's name in the certificate. For `replace`, this mapped DN replaces the bind DN that was created from the subject's name in the certificate. The mapped DN is used when gathering the groups in which the bound user exists and when checking authorization for LDAP operations, including SDBM operations. SDBM must be configured when `add` or `replace` is specified.

When `off` is specified for the first value, RACF is not searched for the user ID associated with the certificate and no certificate mapping is performed. In this case, it does not matter what the second value is (`fail` or `ignore`).
suffix dn_suffix

<table>
<thead>
<tr>
<th></th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Denotes the distinguished name of the root of a subtree in the namespace managed by this backend within the LDAP server. This option may be specified more than once to indicate all the roots of the subtrees within this backend except for the SDBM backend. The SDBM backend must have only one suffix. Note that a suffix cannot be specified for the GDBM, CDBM, and EXOP backends. When the GDBM backend is configured, the cn=changelog suffix is reserved. When the CDBM backend is configured, the cn=configuration and cn=ibmpolicies suffixes are reserved.

Identical and overlapping suffixes cannot be specified in the LDAP server configuration file, even if the suffixes are within different backends. These suffixes create confusion and can result in unexpected results. An example of overlapping suffixes is:

```plaintext
suffix ou=Server Group, o=IBM
suffix o=IBM
```

See [Specifying a value for a distinguished name](#) for information about specifying special characters and restrictions on attributes in the suffix.

Domain Component naming as specified by RFC 2247 is also supported in the LDAP server. For example, the domain name `ibm.com` could be specified as the following suffix in the configuration file:

```plaintext
suffix "dc=ibm,dc=com"
```

supportKrb5 {on | off}

<table>
<thead>
<tr>
<th></th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Specifies if the LDAP server participates in Kerberos GSS API Authentication. If it participates, then Kerberos GSS API binds are accepted and information is stored in the server’s root DSE.

Default = off

tcpTerminate {terminate | recover}

<table>
<thead>
<tr>
<th></th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Specifies whether the LDAP server ends when network interfaces are not active. The LDAP server periodically polls the network interfaces it is using to determine when they go down and come back up. If an interface fails but the LDAP server still has at least one active interface, the server continues processing and reestablishes a failed interface when it detects that it has become active. If all interfaces fail and `tcpTerminate terminate` is specified, the LDAP server ends. If `tcpTerminate recover` is specified, then the LDAP server remains active and attempts to reestablish network interfaces when it detects they have become active. All client operations targeted to the LDAP server fail until a network interface can be reconnected. The frequency of polling can be set using the `LDAP_NETWORK_POLL` environment variable. See Environment variables used by the LDAP server for more information.

The `tcpTerminate` option is also used to determine whether the LDAP server ends if SSL or Kerberos initialization fails during server initialization. If `terminate` is specified, the LDAP server ends. If `recover` is specified, the LDAP server continues initialization, but the failed interface (SSL or Kerberos) cannot be used until the error is fixed and the LDAP server is restarted.
Default = recover
timeLimit num-seconds

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

Specifies the maximum number of seconds (in real time) the LDAP server will spend answering a search request. This maximum number can be modified on a specific search request as described below. If a request cannot be processed within this time, a result indicating an exceeded time limit is returned.

Range = 0 - 2147483647
0 = no limit
Default = 3600

This option applies to all backends, except EXOP, unless specifically overridden in a backend definition. Specifying this before a database line in the configuration sets the option for all backends, except EXOP. Specifying it after a database line sets the option just for the backend defined by the database line.

A limit on the amount of time can also be specified by the client on a search request. Note that the following behavior is used on a search operation when referring to the timeLimit parameter.

When accessing the LDAP server using the TDBM, LDBM, CDBM, or GDBM backend:
- If the client has not bound as the adminDN, then the limit is the smaller of the limit passed by the client and the limit read by the server from the timeLimit option in ds.conf (which defaults to 3600). If the client does not specify a limit, then the server limit is used.
- If the client has bound as the adminDN, then the limit is the value passed by the client. If the client does not specify a limit, then there is no time limit. The limit from the ds.conf file is ignored when the client has bound as the adminDN.

When accessing the z/OS LDAP server support for RACF (the SDBM backend):
- The limit is the smaller of the limit passed by the client and the limit read by the server from the timeLimit option in ds.conf (which defaults to 3600). If the client does not specify a limit, then the server limit is used. It does not matter how the client has bound.

There are additional considerations for time limit when performing a subtree search from the root DSE (a NULL-based search). See Root DSE search with subtree scope (Null-based subtree search) for more information.

useAdvancedReplication {on | off}

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Specifies if the LDAP server supports advanced replication. If advanced replication is active, then the masterServer, masterServerDN, masterServerPW, peerServer, peerServerDN, and peerServerPW configuration options cannot be specified in any LDBM, TDBM, or CDBM backends.

Note:
- The LDAP server will not start when useAdvancedReplication on is specified and entries with an objectclass of replicaObject are present in a TDBM, LDBM, or CDBM backend. If entries with an objectclass of replicaObject are attempted to be added or modified in this configuration, the add or modify request is rejected.
The LDAP server will not start when `useAdvancedReplication off` is specified and entries with an objectclass of `ibm-replicationAgreement`, `ibm-replicationContext`, `ibm-replicationGroup`, or `ibm-replicationSubEntry` are present in a TDBM, LDBM, or CDBM backend. If entries with these objectclass values are attempted to be added or modified in this configuration, the add or modify request is rejected.

See Chapter 24, “Advanced replication” for additional information about advanced replication.

The server compatibility level must be at least 5 when `useAdvancedReplication on` is specified. See the `serverCompatLevel` configuration option on page 100 for more information about the server compatibility level.

Default = off

`useNativeAuth {selected | all | off}`

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

Enables native authentication in the backend. If the value is:
- `selected`, only entries with the `ibm-nativeId` attribute that are within the native subtrees (see `nativeAuthSubtree` option at 31) use native authentication.
- `all`, all entries within native subtrees use native authentication. These entries can contain the `ibm-nativeId` or `uid` attribute to specify the RACF ID.
- `off`, no entries participate in native authentication.

Default = off

`validateincomingV2strings {on | off }`

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Specifies whether the incoming strings are validated. If set to on, this setting limits the format of incoming string data sent over the LDAP Version 2 protocol to the IA5 character set (X'00'-X'7F' or "7-bit ASCII"). With this setting, textual data received on operations outside of the IA5 character set causes the operations to fail with `LDAP_PROTOCOL_ERROR`.

Default = on

Note that while supported, it is not recommended to run with this data filtering disabled.

`wlmExcept name [IP_address] [dn]`

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>TDBM</th>
<th>LDBM</th>
<th>SDBM</th>
<th>GDBM</th>
<th>CDBM</th>
<th>EXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Specifies the Workload Manager (WLM) transaction name used for client requests originating from an IP address or a bound user’s distinguished name (DN). The `wlmExcept` configuration option can be specified multiple times to allow the routing of different LDAP client requests to the same or different WLM transaction names. The order of the `wlmExcept` configuration options in the LDAP server configuration file determines the order the LDAP server uses to match incoming client requests and route them to the WLM transaction name. During LDAP server initialization, an WLM enclave is created for each unique name. See Workload manager (WLM) for more information about configuring the LDAP server to use WLM.

`name` Specifies the WLM transaction name used for this enclave. The name must be 1-8 characters long and can consist of letters, numbers, and the special characters ‘$’, ‘#’, or
'@'. The WLM transaction name must be configured in WLM. Multiple wlmExcept configuration options with the same name use the same enclave.

**IP_address**
Specifies the client's IPv4 or IPv6 address to be associated with this WLM enclave.

dn
Specifies the bind user's distinguished name to be associated with this WLM enclave. For information about specifying a value for a distinguished name for this option, see Specifying a value for a distinguished name.

**Notes:**
1. If both the IP_address and dn values are not specified with the wlmExcept configuration option, a WLM enclave is created with the transaction value name. However, the enclave is not associated with any client requests until a WLMEXCEPT modify command is issued.
2. If both IP_address and dn are specified, only incoming client requests originating from that IP_address and bound as the dn are routed to the WLM transaction name specified.

Default = GENERAL

By default, the WLM transaction name, GENERAL, is used by the LDAP server for client requests originating from IP addresses or bind distinguished names not specified on wlmExcept configuration options. WLM transaction name GENERAL needs to be configured in WLM. See z/OS MVS Planning: Workload Management for more information about configuring WLM.

**Deprecated options**
The listen parameter deprecates the security, port, and securePort options in the configuration file. If a listen option is specified in the configuration file with either security, port, or securePort, the listen will take precedence over what has been specified for the deprecated security, port, and securePort options. If using an earlier version of the configuration file that contains the security, port, or securePort options, the LDAP server is configured to listen on the port numbers specified for securePort, port, or both, depending upon the security setting. However, it is highly recommended that the LDAP server be configured using the listen option. See the description of the listen option at 86 for more information.

**Ignored options**
The replKeyRingFile and replKeyRingPW options are no longer necessary or evaluated by the LDAP server. These options should be removed from the configuration file. Use the sslKeyRingFile configuration option to specify the key database file, RACF key ring, or PKCS #11 token. The sslKeyRingPWStashFile configuration option is used to specify the password stash file for the key database file while the sslKeyRingFilePW configuration option is used to specify the password of the key database file.

The maxThreads and waitingThreads options are no longer necessary or evaluated by the LDAP server. These options should also be removed from the configuration file. Use the commThreads option to set the number of threads initialized at server start-up for communicating with the clients. See the description of the commThreads option at 79 for more information.

The databasename and verifySchema options are no longer necessary or evaluated by the LDAP server. These options should be removed from the configuration file.

The sysplexGroupName and sysplexServerName options are no longer necessary or evaluated by the LDAP server. These options should be removed from the configuration file. Use the serverSysplexGroup option to identify the cross-system coupling facility (XCF) group.
CDBM backend configuration entries

When the CDBM backend is configured in the LDAP server configuration file, configuration related entries are stored in the `cn=configuration` suffix. These entries contain attributes that represent configuration options. The attribute values can be changed dynamically by an LDAP modify command while the LDAP server is running. Unless otherwise noted below, the changes take effect immediately, without needing to restart the server. By default, the CDBM backend only allows the LDAP administrator to modify the configuration entries, but access can be changed by modifying the ACL on the configuration entries.

When the LDAP server starts, the configuration entries that do not already exist are created with each attribute assigned to its default value. If an attribute value is deleted, the default value is used. The deleting and renaming of advanced replication configuration entries is only supported when `useAdvancedReplication off` is specified in the CDBM backend.

This section discusses the entries that exist under the `cn=configuration` suffix and the attribute values in these entries that affect the configuration of the LDAP server.

**cn=configuration**

This is a container entry that is used to define the server ID of this server if advanced replication is configured. See Chapter 24, “Advanced replication” for more information about advanced replication.

<table>
<thead>
<tr>
<th>Attribute description and default</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>cn</strong></td>
</tr>
<tr>
<td>Specifies the common name of the configuration entry. This attribute is never interpreted by the server and therefore does not affect advanced replication configuration.</td>
</tr>
<tr>
<td><strong>Default:</strong> Configuration</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>ibm-slapdServerID</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Specifies a short descriptive name of this server in an advanced replication environment. This server name is used when configuring the relationships among LDAP servers in an advanced replication environment and therefore a unique <code>ibm-slapdServerID</code> value should be chosen for each server in the replication topology. This value is displayed as the <code>ibm-serverID</code> attribute value in the root DSE entry.</td>
</tr>
<tr>
<td><strong>Default:</strong> A randomly generated attribute value similar to an <code>ibm-entryUUID</code> attribute value that is created when the CDBM backend is first initialized.</td>
</tr>
</tbody>
</table>

**cn=Replication,cn=configuration**

This entry is used to configure many aspects of advanced replication such as the maximum number of pending or failed replication changes displayed for a replication agreement. See Chapter 24, “Advanced replication” for more information about advanced replication.

<table>
<thead>
<tr>
<th>Attribute description and default</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>cn</strong></td>
</tr>
<tr>
<td>Specifies the common name of the configuration entry. This attribute does not affect advanced replication configuration.</td>
</tr>
<tr>
<td><strong>Default:</strong> Replication</td>
</tr>
<tr>
<td>Attribute description and default</td>
</tr>
<tr>
<td>-----------------------------------</td>
</tr>
<tr>
<td><strong>ibm-replicationOnHold</strong></td>
</tr>
<tr>
<td>A boolean (true or false) used to indicate whether replication is suspended from all replication agreements in the server. If set to true, replication from all replication agreements is suspended and updates are queued. If set to false, replication updates are handled normally by each replication agreement.</td>
</tr>
<tr>
<td><strong>Default:</strong> false</td>
</tr>
</tbody>
</table>

| **ibm-slapdMaxPendingChangesDisplayed** |
| Specifies the maximum number of pending replication changes and the maximum number of failed replication changes that are displayed when searching a replication agreement on a supplier server. Increase this value if more pending and failed changes must be displayed for each replication agreement. The pending replication changes are stored in the replication agreement entry in the `ibm-replicationPendingChanges` multi-valued operational attribute. The failed replication changes are stored in the `ibm-replicationFailedChanges` multi-valued operational attribute. See [Table 74](#) for more information about these operational attributes. |
| The value must be between 0 and the maximum integer size. A value of 0 indicates that no pending changes are displayed for each replication agreement. |
| **Default:** 200                   |

| **ibm-slapdReplConflictMaxEntrySize** |
| Specifies the maximum length (in bytes) for all attribute values in an entry for replication conflict resolution to occur. If a replication conflict occurs on the consumer server and the total attribute value length for all values in an entry is less than or equal to this number, the entry is resent to the consumer server to automatically correct the replication conflict. Otherwise, the entry is not resent to the consumer server. This value applies to each replication agreement in the server. |
| Increase this value when large entries are modified so that out of sync conditions between a supplier and consumer server can be resolved automatically with conflict resolution. If automatic replication conflict resolution support is not desired, set this value to a small number. |
| The value must be between 0 and the maximum integer size. A value of 0 indicates that all entries are resent to the consumer server regardless of the size of the entry. |
| **Default:** 0                     |

| **ibm-slapdReplContextCacheSize** |
| Specifies the maximum size of each advanced replication context cache, in bytes. An advanced replication context cache is used to store pending replication updates for each replication context in the server. This cache reduces the number of queries to the backends to find the same information. Increase the size of the cache when replicating more and larger entries, such as large group entries. |
| This value must be between 0 and the maximum integer size. A value of 0 indicates that there are no replication context caches in the server. |
| **Default:** 100000                 |
Table 11. cn=Replication,cn=configuration entry attribute descriptions (continued)

<table>
<thead>
<tr>
<th>Attribute description and default</th>
</tr>
</thead>
<tbody>
<tr>
<td>ibm-slapdReplMaxErrors</td>
</tr>
<tr>
<td>Specifies the maximum number of advanced replication failures that are logged for each backend in the server. If there are multiple replication agreement entries in a backend, each agreement shares the maximum number of replication failures allowed for the backend. The failed replication changes are stored in the replication agreement entry in the ibm-replicationFailedChanges multi-valued operational attribute. When the number of replication failures exceeds this value, advanced replication for this agreement stalls. See Monitoring and diagnosing advanced replication problems for more information about recovering from out of sync and stall conditions.</td>
</tr>
<tr>
<td>This value must be between -1 and the maximum integer size. A value of 0 indicates that advanced replication failures are not logged for any replication agreements, therefore, replication stalls at the first failed replication update. A value of -1 indicates that an unlimited number of advanced replication failures are logged for all replication agreements.</td>
</tr>
<tr>
<td>Default: 0</td>
</tr>
</tbody>
</table>

| ibm-slapdReplRestrictedAccess     |
| A boolean (true or false) used to control access to replication topology entries (replication contexts, groups, subentries, and agreements). This attribute provides a way to limit access to the replication topology entries in the LDAP server. If set to true, only the LDAP administrator and the master server DN have access to replication topology entries. If set to false, other users with the proper authority can access the replication topology entries. |
| Default: false                    |
| Note: This is a user-modifiable operational attribute. This attribute must be specifically requested on search requests in order to be returned. |

**cn=Log Management,cn=Configuration**

This is a container entry that does not contain any attribute values that affect the configuration of the LDAP server.

**cn=Replication,cn=Log Management,cn=Configuration**

This entry is used by advanced replication to specify the location of the lost and found log file. See Chapter 24, “Advanced replication” for more information about advanced replication.

Table 12. cn=Replication,cn=Log Management,cn=Configuration entry attribute descriptions

<table>
<thead>
<tr>
<th>Attribute description and default</th>
</tr>
</thead>
<tbody>
<tr>
<td>cn</td>
</tr>
<tr>
<td>Specifies the common name of the configuration entry. This attribute does not affect advanced replication configuration.</td>
</tr>
<tr>
<td>Default: Replication</td>
</tr>
</tbody>
</table>

| ibm-slapdLog                       |
| Specifies the z/OS Unix System Services filename and directory location for the lost and found log file. The lost and found log file is created by the consumer server the first time a replication conflict occurs. Any entries that are deleted because of a replication conflict are stored in LDIF format in this file. The directory path that is specified in this attribute value must exist before the file is created, otherwise replication conflicts are not written to the lost and found log file. |
| This value cannot be deleted when advanced replication is configured. If this value is modified, the original value is still used until the LDAP server is restarted. |
| Default: /var/ldap/logs/lostandfound.log |
### Configuration considerations

The following table shows all of the different options you have and the decisions you must make for your LDAP server configuration. It also shows where you can find the associated reference information to help you make these decisions.

Table 13. Configuration considerations

<table>
<thead>
<tr>
<th>Dependency</th>
<th>More information</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Operational mode</strong></td>
<td>Determining operational mode</td>
</tr>
<tr>
<td>You need to determine the type of operational mode your LDAP server will run in. For example, single-server mode or multi-server mode.</td>
<td></td>
</tr>
<tr>
<td><strong>TDBM backend</strong></td>
<td>Setting up for TDBM</td>
</tr>
<tr>
<td>You can use a TDBM backend database based on DB2.</td>
<td></td>
</tr>
<tr>
<td><strong>SDBM backend</strong></td>
<td>Setting up for SDBM</td>
</tr>
<tr>
<td>You can use a SDBM backend database based on RACF.</td>
<td></td>
</tr>
<tr>
<td><strong>LDBM backend</strong></td>
<td>Setting up for LDBM</td>
</tr>
<tr>
<td>You can use a LDBM backend database based on a UNIX System Services file system.</td>
<td></td>
</tr>
<tr>
<td><strong>GDBM backend</strong></td>
<td>Configuring file-based GDBM</td>
</tr>
<tr>
<td>You can use a GDBM backend database based on DB2 or based on a UNIX System Services file system to log changes to entries within the LDAP server and to RACF user, group, connection, and general resource profiles.</td>
<td></td>
</tr>
<tr>
<td><strong>CDBM backend</strong></td>
<td>Setting up for CDBM</td>
</tr>
<tr>
<td>You can use a CDBM backend database based on a UNIX System Services file system for configuration information.</td>
<td></td>
</tr>
<tr>
<td><strong>EXOP backend</strong></td>
<td>Chapter 20, &quot;Using extended operations to access Policy Director data&quot;</td>
</tr>
<tr>
<td>You can use an EXOP backend to retrieve Policy Director data.</td>
<td></td>
</tr>
<tr>
<td><strong>Plug-in extension</strong></td>
<td>IBM Tivoli Directory Server Plug-in Reference for z/OS</td>
</tr>
<tr>
<td>You can use a plug-in to extend the capabilities of the LDAP server.</td>
<td></td>
</tr>
<tr>
<td><strong>SSL/TLS</strong></td>
<td>Setting up for SSL/TLS</td>
</tr>
<tr>
<td>If you want to protect LDAP access with Secure Socket Layer (SSL) or Transport Layer Security (TLS), your LDAP server can be configured to provide server and, optionally, client authentication.</td>
<td></td>
</tr>
<tr>
<td><strong>Encryption</strong></td>
<td>Configuring for encryption</td>
</tr>
<tr>
<td>Your LDAP server can prevent unauthorized access to ibm-slapdMasterPw, ibm-replicaKeyPwd, userPassword, secretKey, and replicaCredential values in a TDBM, LDBM, or CDBM backend database.</td>
<td></td>
</tr>
<tr>
<td><strong>Kerberos authentication</strong></td>
<td>Chapter 17, &quot;Kerberos authentication&quot;</td>
</tr>
<tr>
<td>You can enable GSS API Kerberos binds and configure identity mapping.</td>
<td></td>
</tr>
<tr>
<td><strong>Native authentication</strong></td>
<td>Chapter 18, &quot;Native authentication&quot;</td>
</tr>
<tr>
<td>You can enable and configure your directory to perform authentication using the Security Server.</td>
<td></td>
</tr>
<tr>
<td>Dependency</td>
<td>More information</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>-----------------------------------------</td>
</tr>
<tr>
<td>CRAM-MD5 and DIGEST-MD5 authentication</td>
<td>Chapter 19, “CRAM-MD5 and DIGEST-MD5 authentication”</td>
</tr>
<tr>
<td>Administrator DN and replica server DN and passwords</td>
<td>Establishing the administrator DN and basic replication replica server DN and passwords</td>
</tr>
<tr>
<td>Basic replication</td>
<td>See Chapter 23, “Basic replication” or Establishing the administrator DN and basic replication replica server DN and passwords for more information.</td>
</tr>
<tr>
<td>Advanced replication</td>
<td>See Chapter 24, “Advanced replication” for more information.</td>
</tr>
<tr>
<td>Referrals</td>
<td>See Chapter 27, “Referrals” for more information.</td>
</tr>
</tbody>
</table>

“Example configuration scenarios” on page 120 has a variety of examples showing different LDAP server configurations.

**Determining operational mode**

When the software has been installed, you are ready to configure it for use at your site. The LDAP server may be configured to run in one of several operational modes when a TDBM, LDBM, CDBM, or GDBM backend is configured.

- **Single-server mode**
  
  In this operational mode, only a single instance of the LDAP server may use a given TDBM, LDBM, CDBM, or GDBM database to store directory data. This server may perform basic or advanced replication (see Chapter 23, “Basic replication” or Chapter 24, “Advanced replication”) of TDBM, LDBM, or CDBM database changes to other servers (on the same host system or on another host system). See Operating in single-server mode for more information.

- **Multiple single-server mode LDAP servers**
  
  In this operational mode, two or more LDAP servers, each in single-server mode, can be run on the same system with different TDBM, LDBM, CDBM, or GDBM backends. This server may perform basic or advanced replication (see Chapter 23, “Basic replication” or Chapter 24, “Advanced replication”) of TDBM, LDBM, or CDBM database changes to other servers (on the same host system or on another host system). However, each server must have its own separate set of replica servers. See Operating in multiple single-server mode for more information.

- **Multi-server mode**
  
  In this operational mode, multiple concurrent instances of the LDAP server use the same TDBM, LDBM, CDBM, or GDBM database to store directory data. The LDAP servers may run on the same host system or on different host systems. In both cases, XCF messaging (Parallel sysplex support) is used to communicate between the LDAP servers. A parallel sysplex is a collection of z/OS systems that cooperate, using certain hardware and software products, to process work. A parallel sysplex enables...
high-performance, multisystem data sharing across multiple Central Processor Complexes and z/OS images, and dynamic workload balancing across constituent systems in the sysplex. For additional information, see [z/OS Parallel Sysplex Overview].

These servers may perform basic or advanced replication of TDBM, LDBM, or CDBM database changes to other servers (on the same host system or on another host system). However, each server must have the same set of replica servers. See Chapter 23, “Basic replication” or Chapter 24, “Advanced replication” for more information.

Multi-server mode is intended for use in an environment where high transactional volume is common, or where maximum availability is required. This mode provides benefits of improved availability, fault tolerance, improved resource utilization, and improved performance. These benefits are achieved by enabling concurrent running of multiple servers that are functionally equivalent and that provide access to the same LDAP directory data.

See [Operating in multi-server mode] for more information.

The following operational mode can be used in conjunction with any of the modes described above.

- **Program call (PC) callable support mode**
  The program call (PC) callable support in LDAP directory provides a program call interface to the LDAP directory extended operations backend (EXOP) and to the change log backend (GDBM). This interface is only available using the z/OS SAF interfaces designed to allow Policy Director access to LDAP data and to allow RACF to log changes to RACF data in the LDAP change log.

  See [Operating in PC callable support mode] for more information.

In any of these modes, all combinations of TDBM (one or more), LDBM (one or more), SDBM, GDBM, CDBM, and EXOP backends are supported. The GDBM backend requires the SDBM backend to create change log entries for changes to RACF data.

**Notes:**

1. A single LDAP server instance can have one SDBM backend, one GDBM backend, and one CDBM backend, but it can have multiple TDBM and LDBM backend instances.
2. If multiple single-server mode LDAP servers are being used on the same system, only one of the LDAP servers can be configured for PC callable support.
3. If multi-server mode is being used and RACF data is accessed from both servers, then the RACF database should also be shared across the systems where the LDAP servers run to ensure consistency of SDBM operations. See [z/OS Security Server RACF System Programmer’s Guide] for information about setting up a shared RACF database.

**Operating in single-server mode**

For the LDAP server to operate in single-server mode, the server configuration file may contain any of the previously documented options except the `serverSysplexGroup` option (the presence that causes the LDAP server to operate in multi-server mode). If the `multiserver` option is present, its value must be set to off.

**Restrictions**

- If one LDAP server instance using a given DB2-based backend to store directory information is operating in single-server mode, it must be the only instance of the LDAP server using that DB2-based backend. Configuring more than one LDAP server instance to use the same DB2 database may yield unpredictable results if one or more of those server instances is configured in single-server mode. If you want to access the same DB2-based backend from more than one server instance, all server instances using the same DB2-based backend must be configured to operate in multi-server mode.

**Operating in multiple single-server mode**

For the LDAP server to operate in multiple single-server mode, there must be two or more LDAP servers running on the same system, each running in single-server mode. The server configuration file may
contain any of the previously documented options except the serverSysplexGroup option (the presence that causes the LDAP server to operate in multi-server mode). If the multiserver option is present, its value must be set to off.

Restrictions

The LDAP servers cannot share TDBM, LDBM, CDBM, or GDBM backends and cannot share the LDAP server schema. This means that each server must have unique values for the databaseDirectory configuration option (for an LDBM, file-based GDBM, or CDBM backend), the dbuserid configuration option (for a TDBM or DB2-based GDBM backend), and the schemaPath configuration option (for the schema).

Setting up multiple LDAP servers with DB2-based backends

In order to set up two or more LDAP servers on the same system with different DB2-based backends, do the following:

• Follow the steps outlined in Creating the DB2 database and table spaces for TDBM or GDBM and be sure to:
  1. Modify the SPUFI file that creates tables and indexes. Change -DB_USERID- (database owner) and -DB_NAME- (database name) to a different value and submit the SPUFI. A separate set of DB2 tables is created.
  2. Update the dbuserid configuration option with the -DB_USERID- value from step 1.

Operating in multi-server mode

For the LDAP server to operate in multi-server mode, the global section of the server configuration file must contain the serverSysplexGroup option. This option specifies the name of the cross-system coupling facility (XCF) group and activates the LDAP sysplex support. All the LDAP servers in the sysplex that specify the same group name share the LDAP server schema and the directories of backends that specify multiserver on in their backend section of the server configuration file. Each LDAP server in the XCF group must specify the same value for the schemaPath option and must have read/write access to the specified directory or to /var/ldap/schema if the option is not specified. The schema directory that is used must reside within a shared z/OS UNIX System Services file system and must be accessible to all servers in the XCF group. An LDBM, CDBM, or file-based GDBM backend is shared when multiserver on is specified in the backend section. When an LDBM, CDBM, or file-based GDBM backend is shared, the databaseDirectory configuration option must have the same value, reside within a shared z/OS UNIX System Services file system, and must be accessible to all servers in the XCF group. Each server must have read/write access to the specified directory or to /var/ldap/ldbm (LDBM default), /var/ldap/schema (CDBM default if schemaPath is not specified), /var/ldap/gdbm (GDBM default) if the databaseDirectory configuration option is not specified. See z/OS UNIX System Services Planning for information about setting up a shared z/OS UNIX System Services file system.

The port specified in the listen, port, and securePort options must be the same on each LDAP server in the sysplex group for dynamic workload balancing to function properly in the sysplex.

When you are using referrals, multiple default referrals defined for other servers can be set up to point to each of the multiple server instances. Similarly, any referral objects defined in other servers that point to the multiple server instances can have multi-valued ref attributes set up, each of which is an LDAP directory URL pointing to the corresponding server instances.

When LDAP servers in a sysplex group are started, the first server to start becomes the sysplex owner of all the shared backends, including the schema. The LDAP server DISPLAY XCF operator modify command can be used to determine the active servers in the sysplex group and shows which one is the owner. See Displaying performance information and server settings for more information.

The sysplex owner is responsible for making all updates to the schema and to any shared LDBM or CDBM backends. If changes to the schema, LDBM, or CDBM entry are directed to another server in the sysplex group, that server uses XCF to forward the change to the sysplex owner. The sysplex owner
makes the change, both in memory and in the database checkpoint file, and then uses XCF to broadcast the change to the other servers. The other servers update their directory in memory.

For a shared TDBM backend, any server in the sysplex group can update a TDBM entry in the database in DB2. That server then notifies all the other servers (including the sysplex owner) by way of XCF that a change has taken place. These servers refresh their various caches so that out-of-date contents are not used. This sysplex support for caches is only available when the serverCompatLevel configuration option is set to 4 or higher. If the serverCompatLevel configuration option is not at this level or higher, the servers in the sysplex do not support caching in the shared TDBM backend.

All replication is handled by the sysplex owner. If a shared backend is replicated, the sysplex owner creates the necessary replication information for a change to an entry in that backend and sends it to the replica or consumer servers.

If the GDBM backend is shared, GDBM is contacted to create a change log record on the server that makes the change. GDBM processing depends on how GDBM is configured. If GDBM is file-based, it handles the change log request similar to LDBM: the change log request is forwarded to the sysplex owner, who creates the change log record in GDBM and notifies the other servers. When GDBM is DB2-based, it handles the request such as TDBM: the DB2 database is updated locally and the other servers are notified.

If the LDAP server that is the sysplex owner ends, another LDAP server in the sysplex group becomes the owner.

**Dependencies**
When sharing a DB2-based backend on multiple systems in a sysplex, the DB2 subsystems that each server instance will attach (see Getting DB2 installed and set up for CLI and ODBC) must be configured on each of the images as members of the same DB2 data sharing group. See DB2 Data Sharing: Planning and Administration for information about planning, installing, and enabling DB2 data sharing, and z/OS MVS Setting Up a Sysplex for information about planning and installing a Parallel Sysplex.

**Restrictions**
Each LDAP server in the XCF group must specify the same value for the schemaPath configuration option and must have read/write access to the specified directory or to /var/ldap/schema if the option is not specified. The schema directory that is used must reside within a shared z/OS UNIX System Services file system and must be accessible to all servers in the XCF group. See z/OS UNIX System Services Planning for information about setting up a shared z/OS UNIX System Services file system.

You can configure one backend to be shared (multiserver on) while another backend is not shared (multiserver off) except when GDBM or CDBM is configured. When GDBM or CDBM is configured, all TDBM, LDBM, CDBM, and GDBM backends must be configured to be shared or all must be not shared.

**Note:** The following restrictions apply to the backend section of the configuration file on each LDAP server that is sharing the backend:

- The name parameter must be specified on the database option and must have the same value on each server.
- The values of the suffix option must be the same on each server.
- For a DB2-based backend, the dbuserid option must have the same value.
- For a file-based backend, the databaseDirectory configuration option must have the same value, reside within a shared z/OS UNIX System Services file system, and must be accessible to all servers in the XCF group. Each server must have read/write access to the specified directory or to /var/ldap/ldbm (LDBM default), /var/ldap/schema (CDBM default if the schemaPath configuration option is also not specified), /var/ldap/gdbm (GDBM default) if the databaseDirectory configuration option is not specified. See z/OS UNIX System Services Planning for information about setting up a shared z/OS UNIX System Services file system.
The SDBM backend does not support the multiserver configuration option. RACF provides the sysplex sharing support for the RACF database. See z/OS Security Server RACF System Programmer's Guide for information about setting up a shared RACF database.

Operating in PC callable support mode

The program call (PC) callable support in the LDAP server provides a program call interface to the LDAP server extended operations backend (EXOP) and to the change log backend (GDBM). This interface is only available using the z/OS SAF interfaces designed to allow Policy Director access to LDAP server data and to allow RACF to log changes to RACF data in the LDAP server change log. The PC callable support is initialized in an LDAP server when the appropriate listen option is included in the configuration file or specified when starting the server. An LDAP server can be dedicated to running just the PC callable support or it can run the PC callable support in addition to its typical socket interfaces.

Running the PC callable support has two interactions with the system:
- The address space of the LDAP server is made non-swappable during initialization of the PC callable support. As a result, resources used by that address space can significantly affect system performance.
- Because the PC callable support connects its PC table to a system index, the address space identifier of the LDAP server address space is not re-usable until the next IPL. If the system is configured with a low limit on the number of address spaces, it is possible to run out of address space identifiers, preventing new address spaces from being started. This problem is more likely to occur if the LDAP server running PC callable support is frequently brought down and re-started.

When using the PC callable support for Policy Director access to LDAP server data, consider configuring a separate LDAP server to run only the PC callable support. Because the server is not also running the backend controlling the data, fewer resources is made non-swappable and the server will less likely need to be re-started. The disadvantage is that an extended operation request will require the LDAP server to communicate with another LDAP server for the data needed to satisfy the request, that can be slower than accessing that data on the same LDAP server. In general,
- If the data used in the extended operations is not on this system, then configure a separate LDAP server for the PC callable support.
- If the data is on this system, then try both configurations (PC callable support in a separate LDAP server and PC callable support in the same LDAP server as the data) to determine the impact on performance.

When using the PC callable support for RACF change logging, the LDAP server should also provide typical socket interfaces to allow usage of the change log entries.

At most, one LDAP server in a system can activate PC callable support. If an LDAP server tries to initialize PC callable support after another LDAP server has already tried (successfully or unsuccessfully) to initialize PC callable support, the initialization fails. The first LDAP server that tries to initialize the PC callable support locks the access to the PC callable support until that LDAP server has been shut down. If you are running the LDAP server in a sysplex, configure one LDAP server on each system in the sysplex to run the PC callable support. Each system should share the GDBM, CDBM, and RACF databases to ensure that they return the same results.

Establishing the administrator DN and basic replication replica server DN and passwords

There are several ways that the administrator distinguished name (DN) and password or the replica server DN and password can be configured. One of these ways must be used, because an administrator DN and password are required for the LDAP server and some other LDAP directory programs to operate. The administrator DN must be specified in the global section of the configuration file using the adminDN option (see page 75). The administrator DN password can optionally (this is not recommended) be placed in the configuration file using the adminPW option (see page 75) or can be held in the namespace managed by
this instance of the LDAP server. If a replica is being established for a backend, the masterServerDN or peerServerDN option must be specified in the backend section of the configuration file. The masterServerPW or peerServerPW option can optionally be specified. (This is also not recommended.) All of the options described below are applicable for adminDN and the first three options described below are applicable for masterServerDN and peerServerDN.

- **Administrator DN and password in configuration file**

  The simplest but least secure method is to select an administrator DN that is outside of the scope of suffixes managed by this server (see the suffix option on page [106]). In other words, choose an administrator DN such that it does not fall within the portion or portions of the namespace managed by this server. Selection of this type of administrator DN requires that the password be placed in the configuration file using the adminPW option (see page [75]).

  For example, you might choose a simple DN, such as "cn=Admin" for the administrator DN and a simple password such as secret. The configuration file options may be:

  ```
  adminDN "cn=Admin"
  adminPW secret
  ```

  **Note:** Do not use the example above without changing the password value, and the actual distinguished name.

  When a program or user binds using this administrator DN, the LDAP server verifies that the password supplied on the request matches the value provided in the configuration file for the adminPW option.

  **Note:** When first configuring a TDBM, LDBM, or CDBM backend, it may be necessary to use this approach until the schema supporting the directory entries is loaded. When the schema is loaded and the entry representing the administrator is added, the adminDN can be changed to the entry DN (see the next list item regarding “Administrator DN and password as a TDBM, LDBM, or CDBM entry”). The server must be restarted to pick up the new adminDN.

- **Administrator DN and password as a TDBM, LDBM, or CDBM entry**

  In this method, the administrator DN is established as an entry managed by a TDBM, LDBM, or CDBM backend. The userPassword attribute is used to hold the password for the administrator DN in this case. Alternatively, the password or password phrase can be in a RACF user associated with the administrator DN by using native authentication. See Chapter 18, “Native authentication” for more information about using native authentication.

  For example, if the TDBM or LDBM backend is managing the portion of the namespace "o=Your Company", one administrator DN that could be selected is "cn=LDAP Admin,o=Your Company".

  The configuration file includes the following options:

  ```
  adminDN "cn=LDAP Admin,o=Your Company"
  ```

  ```
  database tdbm GLDBTD31
  ```

  ```
  suffix "o=Your Company"
  ```

  The LDIF-format entry to be added to the database through ldapadd or ldif2ds might be:

  ```
  dn: cn=LDAP Admin,o=Your Company
  objectClass: person
  cn: LDAP Admin
  description: Administrator DN for o=Your Company server
  sn: Administrator
  uid: admin
  userPassword: secret
  ```

  **Note:** Do not use the example above without changing the password value, and the actual distinguished name.

  If this entry is contained in a file system file called admin.ldif, it can be loaded using ldapadd:

  ```
  ldapadd -h ldaphost -p ldapport -D binddn -w passwd -f admin.ldif
  ```
Note: The ldapadd example above assumes that the LDAP server is running and that the suffix entry (entry with the name "o=Your Company") already exists. Furthermore, the binddn is assumed to exist and have sufficient authority to add the entry. When initially setting up the LDAP server, one way to satisfy the assumption is to first configure the LDAP server using the adminDN and adminPW configuration options. Then start the LDAP server, load the suffix entry and the administrator DN entry, using the adminDN and adminPW configuration values for binddn and passwd respectively. After the add operations complete, stop the LDAP server, change the adminDN configuration option value to the name of the entry just added and remove the adminPW configuration option. Then restart the LDAP server.

For a TDBM backend, you can use the load utility, ldif2ds, to load the administrator DN entry.

When a program or user binds using this administrator DN, the LDAP server verifies that the password supplied on the request matches the value of the userPassword attribute stored in the entry in DB2.

CRAM-MD5 and DIGEST-MD5 authentication binds with the adminDN are supported if the entry exists in a TDBM, LDBM, or CDBM backend. The adminDN entry must contain a uid attribute value that is used as the user name by a client application when attempting a CRAM-MD5 or DIGEST-MD5 authentication bind. For more information about CRAM-MD5 and DIGEST-MD5 authentication, see Chapter 19, "CRAM-MD5 and DIGEST-MD5 authentication."

• Administrator DN and password in RACF

This method requires that the LDAP server be configured to use the RACF support provided in the SDBM backend. The administrator DN can be established as a RACF-style DN based upon a RACF user ID. (See RACF-style distinguished names for more information.) In this case, the password for the administrator DN is the RACF user ID’s password or password phrase, and is stored and verified by RACF.

For example, if you configure the LDAP server with RACF support where the portion of the namespace held by RACF is "sysplex=Sysplex1,o=Your Company", and the RACF user ID that is used for the administrator is gladmin, the configuration file includes these options:

```plaintext
adminDN "racfid=gladmin,profiletype=user,sysplex=Sysplex1,o=Your Company"
...  
database sdbm GLDBSD31/GLDBSD64
suffix "sysplex=Sysplex1,o=Your Company"
```

When a program or user binds using this administrator DN, the LDAP server makes a request to RACF to verify that the password supplied on the request matches the RACF password or password phrase for RACF user ID gladmin.

Note that the RACF user ID specified must have an OMVS segment defined and an OMVS UID present.

• krbLDAPAdmin in the configuration file

You may want to configure the administrator to be able to bind to the server through Kerberos. In this case you need to create a Kerberos identity for the user and also add this value to the configuration file. You cannot use the krbLDAPAdmin option to provide a Kerberos identity for the masterServerDN or peerServerDN option.

For example, if the RACF user ID associated with the LDAP administrator is LDAPADM and this identity is configured in Kerberos to be ldapadm@realm1.com, then your configuration file will have the following:

```plaintext
krbLDAPAdmin ibm-krn=ldapadm@realm1.com
```

This allows the administrator to bind to the server through Kerberos Authentication rather than by performing a simple bind.

### Example configuration scenarios

This section shows scenarios of LDAP server configurations. Only some of the options that can be specified for each section of the LDAP server configuration file are shown. See Table 9 for a complete list of the options that are available for each section.
Configuring a TDBM backend with SSL/TLS only and password encryption

The configuration example in this section uses the TDBM backend and shows the configuration file checklist followed by the corresponding sample configuration file.

Table 14. Sample checklist (using TDBM, SSL/TLS, and password encryption)

<table>
<thead>
<tr>
<th>Section</th>
<th>Check</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global</td>
<td>✓</td>
</tr>
<tr>
<td>SSL/TLS</td>
<td>✓</td>
</tr>
<tr>
<td>Sysplex</td>
<td></td>
</tr>
<tr>
<td>Kerberos</td>
<td></td>
</tr>
<tr>
<td>SDBM backend</td>
<td></td>
</tr>
<tr>
<td>Kerberos</td>
<td></td>
</tr>
<tr>
<td>GDBM backend</td>
<td></td>
</tr>
<tr>
<td>Multiserver</td>
<td></td>
</tr>
<tr>
<td>TDBM backend</td>
<td>✓</td>
</tr>
<tr>
<td>Password encryption</td>
<td>✓</td>
</tr>
<tr>
<td>Basic replication</td>
<td></td>
</tr>
<tr>
<td>Multi-server</td>
<td></td>
</tr>
<tr>
<td>Kerberos</td>
<td></td>
</tr>
<tr>
<td>Native authentication</td>
<td></td>
</tr>
<tr>
<td>LDBM backend</td>
<td></td>
</tr>
<tr>
<td>Password encryption</td>
<td></td>
</tr>
<tr>
<td>Basic replication</td>
<td></td>
</tr>
<tr>
<td>Multi-server</td>
<td></td>
</tr>
<tr>
<td>Kerberos</td>
<td></td>
</tr>
<tr>
<td>Native authentication</td>
<td></td>
</tr>
<tr>
<td>CDBM backend</td>
<td></td>
</tr>
<tr>
<td>Password encryption</td>
<td></td>
</tr>
<tr>
<td>Basic replication</td>
<td></td>
</tr>
<tr>
<td>Advanced replication</td>
<td></td>
</tr>
<tr>
<td>Multi-server</td>
<td></td>
</tr>
<tr>
<td>Kerberos</td>
<td></td>
</tr>
<tr>
<td>Native authentication</td>
<td></td>
</tr>
<tr>
<td>EXOP backend</td>
<td></td>
</tr>
</tbody>
</table>

Sample ds.conf for TDBM, SSL/TLS, and password encryption:

```
# Filename ds.conf

# Global section
sizelimit 500
timelimit 3600
adminDn "cn=LDAP Administrator,o=Your Company"

listen ldaps://:636
sslAuth serverClientAuth
```
sslCertificate none
sslCipherSpecs 15104
sslKeyRingFile /u01/ldapsrv/ldapsrv.kdb
sslKeyRingPWStashFile /u01/ldapsrv/ldapsrv.sth

# TDBM backend section
database tdbm GLDBTD31 LocalDirectory
suffix "o=Your Company"
servername LOC1
dbuserid GLDSRV
attrOverflowSize 500
pwEncryption MD5

## Configuring SDBM and GDBM (DB2-based) backends

The configuration example in this section uses SDBM and GDBM backends and shows the configuration file checklist followed by the corresponding sample configuration file. In this example, the GDBM backend is based on DB2.

### Table 15. Sample checklist (using SDBM and GDBM)

<table>
<thead>
<tr>
<th>Section</th>
<th>Check</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global</td>
<td>✔</td>
</tr>
<tr>
<td>SSL/TLS</td>
<td></td>
</tr>
<tr>
<td>Sysplex</td>
<td></td>
</tr>
<tr>
<td>Kerberos</td>
<td></td>
</tr>
<tr>
<td>SDBM backend</td>
<td>✔</td>
</tr>
<tr>
<td>Kerberos</td>
<td></td>
</tr>
<tr>
<td>GDBM backend</td>
<td>✔</td>
</tr>
<tr>
<td>Multiserver</td>
<td></td>
</tr>
<tr>
<td>TDBM backend</td>
<td></td>
</tr>
<tr>
<td>Password encryption</td>
<td></td>
</tr>
</tbody>
</table>
Sample `ds.conf` for SDBM and GDBM:

```plaintext
# Global section
sizelimit 500
timelimit 3600
adminDn "racfid=ldadmin,profiletype=user,cn=myRACF"
listen ldap://:pc
listen ldap://:389

# SDBM backend section
database sdbm GLDBSD31/GLDBSD64
suffix "cn=myRACF"

# GDBM backend section
database gdbm GLDBGD31/GLDBGD64
servername LOC1
dbuserid GLDSRV
attrOverflowSize 500
```

**Configuring SDBM and TDBM backends**

The configuration example in this section uses both SDBM and TDBM backends and shows the configuration file checklist followed by the corresponding sample configuration file.

**Table 16. Sample checklist (using SDBM and TDBM)**

<table>
<thead>
<tr>
<th>Section</th>
<th>Check</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global</td>
<td>✓</td>
</tr>
<tr>
<td>SSL/TLS</td>
<td></td>
</tr>
<tr>
<td>Sysplex</td>
<td></td>
</tr>
<tr>
<td>Kerberos</td>
<td></td>
</tr>
<tr>
<td>SDBM backend</td>
<td>✓</td>
</tr>
<tr>
<td>Kerberos</td>
<td></td>
</tr>
<tr>
<td>GDBM backend</td>
<td></td>
</tr>
<tr>
<td>Multiserver</td>
<td></td>
</tr>
<tr>
<td>TDBM backend</td>
<td>✓</td>
</tr>
<tr>
<td>Password encryption</td>
<td></td>
</tr>
<tr>
<td>Basic replication</td>
<td></td>
</tr>
<tr>
<td>Multi-server</td>
<td></td>
</tr>
<tr>
<td>Kerberos</td>
<td></td>
</tr>
<tr>
<td>Native authentication</td>
<td></td>
</tr>
<tr>
<td>LDBM backend</td>
<td></td>
</tr>
<tr>
<td>Password encryption</td>
<td></td>
</tr>
<tr>
<td>Basic replication</td>
<td></td>
</tr>
<tr>
<td>Multi-server</td>
<td></td>
</tr>
<tr>
<td>Kerberos</td>
<td></td>
</tr>
<tr>
<td>Native authentication</td>
<td></td>
</tr>
<tr>
<td>CDBM backend</td>
<td></td>
</tr>
<tr>
<td>Password encryption</td>
<td></td>
</tr>
</tbody>
</table>
Table 16. Sample checklist (using SDBM and TDBM) (continued)

<table>
<thead>
<tr>
<th>Section</th>
<th>Check</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic replication</td>
<td></td>
</tr>
<tr>
<td>Advanced replication</td>
<td></td>
</tr>
<tr>
<td>Multi-server</td>
<td></td>
</tr>
<tr>
<td>Kerberos</td>
<td></td>
</tr>
<tr>
<td>Native authentication</td>
<td></td>
</tr>
</tbody>
</table>

**EXOP backend**

Sample ds.conf for SDBM and TDBM:

```bash
# Filename ds.conf

# Global section
sizelimit 500
timelimit 3600
adminDn "racfid=1dadmin,profiletype=user,cn=myRACF"
listen ldap://:389

# SDBM backend section
database sdbm GLDBSD31/GLDBSD64
suffix "cn=myRACF"

# TDBM backend section
database tdbm GLDBTD31
suffix "o=Your Company"
servername LOC1
dbuserid GLDSRV
attrOverflowSize 500
```

**Configuring LDBM with native authentication and GDBM (file-based) backends**

The configuration example in this section uses the LDBM and GDBM backends and shows the configuration file checklist followed by the corresponding sample configuration file. The GDBM backend is based on the UNIX System Services file system.

Table 17. Sample checklist (using LDBM and GDBM)

<table>
<thead>
<tr>
<th>Section</th>
<th>Check</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global</td>
<td></td>
</tr>
<tr>
<td>SSL/TLS</td>
<td></td>
</tr>
<tr>
<td>Sysplex</td>
<td></td>
</tr>
<tr>
<td>Kerberos</td>
<td></td>
</tr>
</tbody>
</table>

**SDBM backend**

Kerberos

**GDBM backend**

<table>
<thead>
<tr>
<th>Check</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Section</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiplex</td>
<td></td>
</tr>
<tr>
<td>Kerberos</td>
<td></td>
</tr>
</tbody>
</table>

**TDBM backend**

Password encryption

<table>
<thead>
<tr>
<th>Check</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Section</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic replication</td>
<td></td>
</tr>
<tr>
<td>Multi-server</td>
<td></td>
</tr>
<tr>
<td>Kerberos</td>
<td></td>
</tr>
<tr>
<td>Section</td>
<td>Check</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>Native authentication</td>
<td></td>
</tr>
<tr>
<td><strong>LDBM backend</strong></td>
<td>✔</td>
</tr>
<tr>
<td>Password encryption</td>
<td></td>
</tr>
<tr>
<td>Basic replication</td>
<td></td>
</tr>
<tr>
<td>Multi-server</td>
<td></td>
</tr>
<tr>
<td>Kerberos</td>
<td></td>
</tr>
<tr>
<td>Native authentication</td>
<td></td>
</tr>
<tr>
<td><strong>CDBM backend</strong></td>
<td></td>
</tr>
<tr>
<td>Password encryption</td>
<td></td>
</tr>
<tr>
<td>Basic replication</td>
<td></td>
</tr>
<tr>
<td>Advanced replication</td>
<td></td>
</tr>
<tr>
<td>Multi-server</td>
<td></td>
</tr>
<tr>
<td>Kerberos</td>
<td></td>
</tr>
<tr>
<td>Native authentication</td>
<td></td>
</tr>
<tr>
<td><strong>EXOP backend</strong></td>
<td></td>
</tr>
</tbody>
</table>

Sample `ds.conf` for LDBM and GDBM:

```plaintext
# Filename ds.conf

# Global section
sizelimit 500
timelimit 3600
adminDn "cn=LDAP Administrator,o=My Company
listen ldap://:389

# GDBM backend section
database gdbm GLDBGD31/GLDBGD64

# LDBM backend section
database ldbm GLDBLD31/GLDBLD64
suffix "o=My Company"
usenativauth all
nativeauthsubtree all
```
### Configuring LDBM and CDBM backends with advanced replication

The configuration example in this section uses both CDBM and LDBM backends and shows the configuration file checklist followed by the corresponding sample configuration file.

#### Table 18. Sample checklist (using CDBM and LDBM)

<table>
<thead>
<tr>
<th>Section</th>
<th>Check</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global</td>
<td></td>
</tr>
<tr>
<td>SSL/TLS</td>
<td></td>
</tr>
<tr>
<td>Sysplex</td>
<td></td>
</tr>
<tr>
<td>Kerberos</td>
<td></td>
</tr>
<tr>
<td>SDBM backend</td>
<td></td>
</tr>
<tr>
<td>Kerberos</td>
<td></td>
</tr>
<tr>
<td>GDBM backend</td>
<td></td>
</tr>
<tr>
<td>Multiserver</td>
<td></td>
</tr>
<tr>
<td>TDBM backend</td>
<td></td>
</tr>
<tr>
<td>Password encryption</td>
<td></td>
</tr>
<tr>
<td>Basic replication</td>
<td></td>
</tr>
<tr>
<td>Multi-server</td>
<td></td>
</tr>
<tr>
<td>Kerberos</td>
<td></td>
</tr>
<tr>
<td>Native authentication</td>
<td></td>
</tr>
<tr>
<td>LDBM backend</td>
<td></td>
</tr>
<tr>
<td>Password encryption</td>
<td></td>
</tr>
<tr>
<td>Basic replication</td>
<td></td>
</tr>
<tr>
<td>Multi-server</td>
<td></td>
</tr>
<tr>
<td>Kerberos</td>
<td></td>
</tr>
<tr>
<td>Native authentication</td>
<td></td>
</tr>
<tr>
<td>CDBM backend</td>
<td></td>
</tr>
<tr>
<td>Password encryption</td>
<td></td>
</tr>
<tr>
<td>Basic replication</td>
<td></td>
</tr>
<tr>
<td>Advanced replication</td>
<td></td>
</tr>
<tr>
<td>Multi-server</td>
<td></td>
</tr>
<tr>
<td>Kerberos</td>
<td></td>
</tr>
<tr>
<td>Native authentication</td>
<td></td>
</tr>
<tr>
<td>EXOP backend</td>
<td></td>
</tr>
</tbody>
</table>

#### Sample ds.conf for CDBM and LDBM:

```plaintext
# File name ds.conf

# Global section
sizelimit 500
timelimit 3600
adminDn "ou=LDAP Administrator,o=My Company
listen ldapi://:389

# LDBM backend section
database ldbm GLDBLD31/GLDBLD64
suffix "o=My Company"
```
# CDBM backend section
database cdbm GLDBCD31/GLDBCD64
useAdvancedReplication on

## Configuring an EXOP backend

The configuration example in this section uses an EXOP backend and shows the configuration file checklist followed by the corresponding sample configuration file.

### Table 19. Sample checklist (using EXOP)

<table>
<thead>
<tr>
<th>Section</th>
<th>Check</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global</td>
<td></td>
</tr>
<tr>
<td>SSL/TLS</td>
<td></td>
</tr>
<tr>
<td>Sysplex</td>
<td></td>
</tr>
<tr>
<td>Kerberos</td>
<td></td>
</tr>
<tr>
<td>SDBM backend</td>
<td></td>
</tr>
<tr>
<td>Kerberos</td>
<td></td>
</tr>
<tr>
<td>GDBM backend</td>
<td></td>
</tr>
<tr>
<td>Multiserver</td>
<td></td>
</tr>
<tr>
<td>TDBM backend</td>
<td></td>
</tr>
<tr>
<td>Password encryption</td>
<td></td>
</tr>
<tr>
<td>Basic replication</td>
<td></td>
</tr>
<tr>
<td>Multi-server</td>
<td></td>
</tr>
<tr>
<td>Kerberos</td>
<td></td>
</tr>
<tr>
<td>Native authentication</td>
<td></td>
</tr>
<tr>
<td>LDBM backend</td>
<td></td>
</tr>
<tr>
<td>Password encryption</td>
<td></td>
</tr>
<tr>
<td>Basic replication</td>
<td></td>
</tr>
<tr>
<td>Multi-server</td>
<td></td>
</tr>
<tr>
<td>Kerberos</td>
<td></td>
</tr>
<tr>
<td>Native authentication</td>
<td></td>
</tr>
<tr>
<td>CDBM backend</td>
<td></td>
</tr>
<tr>
<td>Password encryption</td>
<td></td>
</tr>
<tr>
<td>Basic replication</td>
<td></td>
</tr>
<tr>
<td>Advanced replication</td>
<td></td>
</tr>
<tr>
<td>Multi-server</td>
<td></td>
</tr>
<tr>
<td>Kerberos</td>
<td></td>
</tr>
<tr>
<td>Native authentication</td>
<td></td>
</tr>
<tr>
<td>EXOP backend</td>
<td></td>
</tr>
</tbody>
</table>

Sample **ds.conf** for EXOP:

```bash
# Filename ds.conf

# Global section
listen ldap://:pc

# EXOP backend section
database exop GLDXPD31/GLDXPD64
```

---

Chapter 8. Customizing the LDAP server configuration 127
Chapter 9. Running the LDAP server

This topic describes what is necessary to get the LDAP server running.

Setting up the PDSE for the LDAP server DLLs

The LDAP server searches for and loads a number of dynamic load libraries (DLLs) during its startup processing. All DLLs for the LDAP server are shipped in PDSE format only. In order for these DLLs to be located by the LDAP server at runtime, the PDSE which contains these DLLs (SYS1.SIEALNKE) must either be in the LNKLST (the default installation) or referenced in a STEPLIB DD card. In addition, the SYS1.SIEALNKE dataset must be APF-authorized. If a STEPLIB DD statement is used, all datasets in the concatenation must be APF-authorized. The LDAP server also requires the hlq.SCEERUN and hlq.SCEERUN2 datasets. Both of these datasets must be APF-authorized.

Setting up and running the LDAP server

The LDAP server must be run as a started task. To do this, you must define the started task for the LDAP server and then you can run the LDAP server using JCL. The LDAP server can be run in either 31-bit mode or 64-bit mode. You must use 31-bit mode if you are using a TDBM backend or the DB2-based GDBM backend. All other backends, including the file-based GDBM backend, run in both modes.

Defining the started task for the LDAP server

After you create the LDAPSrv user ID (described in Requirements for a user ID that runs the LDAP server), you must define the DSSRV started task. The examples and the sample startup procedure use the name DSSRV for this task, but you can use any name for it.

To define the started task for the user ID you just created, you can use the following RACF commands.

```
RDEFINE STARTED DSSRV.** STDATA(USER(LDAPSRV))
SETROPTS RACLIST(STARTED) REFRESH
```

Note: When using dsconfig to configure the LDAP server, the started task was already defined.

Running the LDAP server using the sample JCL

The JCL needed to run the LDAP server as a started task is provided with the product as a procedure. This JCL can be found in the DSSRV member of GLDHLQ.SG LDSAMP on the system where the LDAP server is installed. If you have a ServerPac installation, GLDHLQ will be GLD. Use DSSRV for starting the LDAP server in 31-bit or 64-bit mode. The JCL procedure can be started in the System Display and Search Facility (SDSF) or from the operator's console, once the sample JCL has been placed into the installation-specific library for procedures.

The JCL must be tailored before it can be run. In particular, you must change the program name to either GLDSRV31 to run the LDAP server in 31-bit mode or GLDSRV64 to run in 64-bit mode.

To start the LDAP server in SDSF, enter:
```
/s dssrv
```

To start the LDAP server from the operator's console, enter:
```
s dssrv
```

The LDAP server has the following optional command-line parameters. One or more of these may be specified when starting the LDAP server.

- `f pathname`
  Name of LDAP server configuration file to be read. The file name is case-sensitive unless it refers
to an MVS dataset or DD statement. An MVS dataset is indicated by `//dataset-name` while a DD statement is indicated by `//DD:dd-name`. The configuration dataset specified by the `CONFIG DD` statement will be used if the `-f` parameter is not specified. The `/etc/ldap/ds.conf` configuration file will be used if the `-f` parameter is not specified and the `CONFIG DD` statement is not defined.

- **-l ldap_URL**

  Host name or IP address and port number on which the LDAP server should bind and listen for incoming requests. See the `listen` parameter on page 86 for information about the `ldap_URL` parameter. The `-l` parameter can be specified multiple times to add additional `ldap_URL` values. The values specified using the `-l` command-line parameter override the values specified for the `listen` option in the LDAP server configuration file.

  It is advisable to reserve the port number or numbers chosen here in your TCP/IP profile data set. Also, be aware that port numbers below 1024 might require additional specification. See [z/OS Communications Server: IP Configuration Guide](https://www.ibm.com/support/knowledgecenter/SSLTBW_2.2.1/com.ibm.zos.v2r11.cics.icsr/zos_commserver_ipvip.htm) for more information.

- **-m**

  Start the server in maintenance mode rather than normal mode. Maintenance mode severely restricts access to the server. See Basic replication maintenance mode for more information. Also, see Advanced replication maintenance mode for more information about advanced replication maintenance mode.

- **-p port**

  **Note:** The `port` option is deprecated when the `listen` option is specified. See page 86 for information about the `listen` option.

  Port number where the LDAP server will listen for nonsecure communications. If you specify this parameter, it overrides the `port` value that may be in the LDAP server configuration file if there are not any `listen` values specified in the configuration file or `-l` parameters on the command line.

- **-s securePort**

  **Note:** The `securePort` option is deprecated when the `listen` option is specified. See page 86 for information about the `listen` option.

  Port number where the LDAP server will listen for secure communications. If you specify this parameter, it overrides the `securePort` value that may be in the LDAP server configuration file if there are not any `listen` values specified in the configuration file or `-l` parameters on the command line.

- **-d debug_level**

  The debug level is a mask that may be specified as follows:
  - A decimal value (for example, 32)
  - A hexadecimal value (for example, x20 or X20)
  - A keyword (for example, FILTER)
  - A construct of those values using plus and minus signs to indicate inclusion or exclusion of a value. For example:
    - `32768+8` is the same as specifying `x8000+x8`, or `ERROR+CONNS`
    - `2147483647-16` is the same as specifying `x7FFFFFFF-x10` or `ANY-BER`
    - By beginning the debug level with a minus sign, you can deactivate debug collection for a debug level. For example, `"-CONNS"` modifies an existing debug level by deactivating connection traces.
    - By beginning the debug level with a plus sign, you can activate debug collection for a debug level. For example, `"+CONNS"` modifies an existing debug level by activating connection traces.

  **Note:** Specifying the debug level using decimal or hex values with a plus or minus sign is not necessarily the same as specifying the sum or difference as the debug level. For example, specifying `7+1` activates the 'TRACE', 'PACKETS', and 'ARGS' debug levels,
while specifying '8' activates only the 'CONNS' debug level. Similarly, specifying '16-1'
activates only the 'BER' debug level, while specifying '15' activates 'TRACE', 'PACKETS',
ARGS', and 'CONNS'.

Table 20 lists the debug levels and the related decimal, hexadecimal, and keyword values.

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Decimal</th>
<th>Hexadecimal</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFF</td>
<td>0</td>
<td>0x00000000</td>
<td>No debugging</td>
</tr>
<tr>
<td>TRACe</td>
<td>1</td>
<td>0x00000001</td>
<td>Entry and exit from routines</td>
</tr>
<tr>
<td>PACKets</td>
<td>2</td>
<td>0x00000002</td>
<td>Packet activity</td>
</tr>
<tr>
<td>ARGs</td>
<td>4</td>
<td>0x00000004</td>
<td>Data arguments from requests</td>
</tr>
<tr>
<td>CONNs</td>
<td>8</td>
<td>0x00000008</td>
<td>Connection activity</td>
</tr>
<tr>
<td>BER</td>
<td>16</td>
<td>0x00000010</td>
<td>Encoding and decoding of data, including ASCII and EBCDIC translations, if applicable</td>
</tr>
<tr>
<td>FILTER</td>
<td>32</td>
<td>0x00000020</td>
<td>Search filters</td>
</tr>
<tr>
<td>MESSage</td>
<td>64</td>
<td>0x00000040</td>
<td>Messaging subsystem activities and events</td>
</tr>
<tr>
<td>ACL</td>
<td>128</td>
<td>0x00000080</td>
<td>Access Control List activities</td>
</tr>
<tr>
<td>STATs</td>
<td>256</td>
<td>0x00000100</td>
<td>Operational statistics</td>
</tr>
<tr>
<td>THREAD</td>
<td>512</td>
<td>0x00000200</td>
<td>Threading activities</td>
</tr>
<tr>
<td>REPLication</td>
<td>1024</td>
<td>0x00000400</td>
<td>Replication activities</td>
</tr>
<tr>
<td>PARSe</td>
<td>2048</td>
<td>0x00000800</td>
<td>Parsing activities</td>
</tr>
<tr>
<td>PERFORMANCE</td>
<td>4096</td>
<td>0x00001000</td>
<td>Performance statistics</td>
</tr>
<tr>
<td>SDBM</td>
<td>8192</td>
<td>0x00002000</td>
<td>Backend activities (SDBM)</td>
</tr>
<tr>
<td>REFerral</td>
<td>16384</td>
<td>0x00004000</td>
<td>Referral activities</td>
</tr>
<tr>
<td>ERROR</td>
<td>32768</td>
<td>0x00008000</td>
<td>Error conditions</td>
</tr>
<tr>
<td>SYSPlex</td>
<td>65536</td>
<td>0x00010000</td>
<td>Sysplex/WLM activities</td>
</tr>
<tr>
<td>MULTIserver</td>
<td>131072</td>
<td>0x00020000</td>
<td>Multi-server activities</td>
</tr>
<tr>
<td>LDAPBE</td>
<td>262144</td>
<td>0x00040000</td>
<td>Connection between a frontend and a backend</td>
</tr>
<tr>
<td>STRBuf</td>
<td>524288</td>
<td>0x00080000</td>
<td>UTF-8 support activities</td>
</tr>
<tr>
<td>TDBM</td>
<td>1048576</td>
<td>0x00100000</td>
<td>Backend activities (TDBM)</td>
</tr>
<tr>
<td>SCHEma</td>
<td>2097152</td>
<td>0x00200000</td>
<td>Schema support activities</td>
</tr>
<tr>
<td>BECApabilities</td>
<td>4194304</td>
<td>0x00400000</td>
<td>Backend capabilities</td>
</tr>
<tr>
<td>CACHE</td>
<td>8388608</td>
<td>0x00800000</td>
<td>Cache activities</td>
</tr>
<tr>
<td>INFO</td>
<td>16777216</td>
<td>0x01000000</td>
<td>General processing information</td>
</tr>
<tr>
<td>LDBM</td>
<td>33554432</td>
<td>0x02000000</td>
<td>Backend activities (LDBM)</td>
</tr>
<tr>
<td>PLUGin</td>
<td>67108864</td>
<td>0x04000000</td>
<td>Plug-in extension activities</td>
</tr>
<tr>
<td>ANY</td>
<td>2147483647</td>
<td>0x7FFFFFFF</td>
<td>All levels of debug</td>
</tr>
<tr>
<td>ALL</td>
<td>2147483647</td>
<td>0x7FFFFFFF</td>
<td>All levels of debug</td>
</tr>
</tbody>
</table>

Note: The minimum abbreviation for each keyword is shown in uppercase letters.

The debug level for the server can be set at a number of different times.
- The initial debug level is OFF.
- before starting the server, the LDAP_DEBUG environment variable may be set. The server uses this value first. For example,
export LDAP_DEBUG='ERROR+TRACE'

- When starting the server, the -d parameter may be specified. The debug level specified on this parameter either replaces, adds to or deletes from the preceding debug level. For example,
  - s dssrv,parms='-d ERROR'
    replaces the current debug level that is either off or has been set by the LDAP_DEBUG environment variable with the new debug level of only ERROR.
  - s dssrv,parms='-d +ERROR'
    adds the ERROR debug level to the current debug level that is either off or has been set by the LDAP_DEBUG environment variable.
  - s dssrv,parms='-d -ERROR'
    removes the ERROR debug level from the current debug level that is either off or has been set by the LDAP_DEBUG environment variable.
- It is possible to change the debug level while the server is running. See Dynamic debugging for more information.

When the LDAP server has been started and is ready, the message

GLD1004I LDAP server is ready for requests.

is displayed.

Running the LDAP server using data sets discusses using a data set for the LDAP server configuration file. In order to specify the configuration file as either a data set name or a DD name in SDSF, some special syntax is necessary.

In order to specify a full data set name, it might be necessary to be in the expanded input screen for SDSF. This is accomplished by entering a slash (/) in SDSF. On the expanded screen, it is then possible to specify a data set name for the configuration file. Assuming that the configuration file has been established in data set MYUSERID.DS.CONF, the start command for the LDAP server in expanded input SDSF or the console is:

s dssrv,parms=' '-f //''''MYUSERID.DS.CONF''''

or, if additional parameters are needed:

s dssrv,parms=' '-f //''''MYUSERID.DS.CONF'''' -l ldap://:999'

If a DD name, CONFIG, was established in the DSSRV procedure, as follows:

CONFIG DD DSN=MYUSERID.DS.CONF,DISP=SHR

the LDAP server could be started from expanded input SDSF or the console by entering:

s dssrv,parms=' '-f //DD:CONFIG'

To stop the LDAP server in SDSF, enter:

/p dssrv

To stop the LDAP server from the operator’s console, enter:

p dssrv

This command causes the LDAP server to shut down.
**LDAP server messages and debug output**

The LDAP server writes messages to **stdout** and **stderr**. Messages sent to **stdout** and **stderr** appear in **DD:DSOUT** in the provided JCL when running the LDAP server. **DSOUT** appears in the started task log for the LDAP server and can be viewed through SDSF. See [z/OS SDSF Operation and Customization](#) for information about how to use SDSF.

Output from the LDAP server debug facility is directed to the file specified by the **LDAP_DEBUG_FILENAME** environment variable. If this environment variable is not set, the output is sent to **stdout**, which is redirected to **DSOUT** as explained above.

**Running the LDAP server using data sets**

**Note:** Using the LDAP configuration utility (**dsconfig**) to configure your server creates all the necessary files in a partitioned data set.

The LDAP server accepts several of its files as data sets. Data set versions of the configuration file and the environment variables file are not shipped with the LDAP server, but can be created using the **OGET** command to copy the file system versions of the files into data sets. (See [z/OS DCE Command Reference](#) for information about the use of the **OGET** command.)

The default data set characteristics for record format and record length (V 255) which **OGET** will use when creating a new data set are not acceptable for JCL when submitting for batch processing. In order to avoid this, allocate the MYUSER.DSNTIJCL sequential data set to be fixed block 80 before performing the **OGET** operation.

A data set version of the **DSNAOINI** file needed for the TDBM and GDBM (when DB2-based) backends can be created by copying and editing the default file provided by DB2. See step 4. The **DSNAOINI** file name can be specified in the **dsnaini** option in the LDAP server configuration file, in a **DSNAOINI DD** statement in the DSSRV procedure, or in a **DSNAOINI** environment variable. The **DD** statement takes precedence, followed by the environment variable, and then the configuration option.

**Note:** Be sure that use of sequence numbers is turned off when editing this dataset.

Once the data set versions of these files are available, they can be specified in the DSSRV procedure. The configuration file can be specified using the **CONFIG DD** statement, the environment variables file can be specified using the **ENVVAR DD** statement, and the **DSNAOINI** file can be specified using the **DSNAOINI DD** statement.

**Verifying the LDAP server**

The following examples show how you can verify your LDAP server using the **ldapsearch** utility. Note that you can use any LDAP client to do this.

- **Verifying TDBM and LDBM**
  
  In the command below, substitute the **suffix** value from your LDAP server configuration file for the -b parameter. The command can be run multiple times to verify each suffix defined in the configuration file.

  ```bash
  ldapsearch -h 127.0.0.1 -s base -b "o=Your Company" "objectclass=*"
  ```

  **Notes:**
  1. If **allowAnonymousBinds off** is specified in the LDAP server configuration file, you must specify a distinguished name to bind with using the -D and -w options on the **ldapsearch** utility.
  2. The LDAP search will return the message "No such object" if the suffix entries have not been loaded into the directory.

- **Verifying SDBM**
For SDBM, you must bind with a valid RACF-style DN to perform the search. Substitute a RACF ID of your choice in the racfid portion of the DN on the -D and -b parameters below. Also, replace cn=myRacf with your SDBM suffix in the DN on the -D and -b parameters. The RACF password for the user ID used in the -D parameter must be specified in the -w parameter.

```
ldapsearch -h 127.0.0.1 -D racfid=IBMUSER,profiletype=user,cn=myRacf
   -w password_for_IBMUSER -b racfid=IBMUSER,profiletype=user,cn=myRacf "objectclass=*"
```

**Verifying GDBM**

For GDBM, you must bind with the LDAP administrator DN or another DN authorized to search the change log.

```
ldapsearch -h 127.0.0.1 -D binddn -w passwd -s base -b cn=changelog "objectclass=*"
```

**Verifying CDBM**

For CDBM, you must bind with the LDAP administrator DN or another DN authorized to search the cn=ibmpolicies and cn=configuration CDBM suffixes.

```
ldapsearch -h 127.0.0.1 -D binddn -w passwd -s base -b cn=ibmpolicies "objectclass=*"
ldapsearch -h 127.0.0.1 -D binddn -w passwd -s base -b cn=configuration "objectclass=*"
```

The previous ldapsearch examples assume a default port of 389. If your port is not 389, use the -p parameter to specify the correct port.

Be sure to substitute the correct TCP/IP host name or TCP/IP address for the 127.0.0.1 after the -h parameter. The -b parameter specifies the starting point for the search. The use of the quotation marks around the filter prevents the asterisk (*) from being interpreted by the shell.

Note that this can be done from TSO as well, substituting LDAPSRCH for ldapsearch.

See IBM Tivoli Directory Server Client Programming for z/OS for more information about ldapsearch.

### Finalizing setup of LDAP backends

To finalize setup of LDAP backends, follow these steps:

1. If you have configured an LDBM or TDBM backend, modify the LDAP server schema entry to contain the schema needed for your usage of these backends. The schema.user.ldif and schema.IBM.ldif files found in the /usr/lpp/ldap/etc directory may contain the schema you need. schema.IBM.ldif requires that you first load schema.user.ldif. Additional schema might also needed by the applications that are going to use the LDBM or TDBM directory. See Chapter 14, "LDAP directory schema," on page 227 for more information.

   **Notes:**
   
   a. The distinguished name (DN) of the LDAP server schema is cn=schema. If the ldif file containing your schema has a DN of cn=schema.suffix, then update the file to change the DN to cn=schema.
   
   b. Do not add any additional schema to use the GDBM, CDBM, or SDBM backends unless you are using SDBM to access RACF customized fields or adding user defined entries to CDBM. The initial schema built into the LDAP server is usually sufficient for these backends.

2. Load the suffix entries for each configured LDBM and TDBM backend. Do not try to load a suffix entry for the GDBM, CDBM, or SDBM backend. The GDBM, CDBM, and SDBM suffix entries are generated automatically by the LDAP server.

   **Note:** If you intend to load large amounts of data in LDIF format into a TDBM directory, see ldiff2ds utility on page 196 for instructions on using the ldiff2ds utility. In this case, do not load the suffix entry separately. Include the suffix instead with the rest of the entries to be loaded by ldiff2ds.

   Use the ldapadd utility to load the suffix entry.

```
ldapadd -h ldaphost -p ldapport -D binddn -w passwd -f file
```
file is a file containing the entries to be loaded in LDIF format. For example, if suffix.ldif contains
the following suffix entry:

dn: o=Your Company
objectclass: organization
o: Your Company

then the suffix entry can be added by ldapadd as follows:

dladd -h myhost -p 389 -D cn=admin -w secret -f suffix.ldif

See IBM Tivoli Directory Server Client Programming for z/OS for more information about the ldapadd
utility.

3. Load additional entries into the LDBM, TDBM, or CDBM backend, as you want. You can use the
ldapadd utility to load entries into these backends. For loading a large number of entries into TDBM,
you should consider using the ldif2ds utility. See "ldif2ds utility" on page 196 for more information. You
must load entries in hierarchical order and an entry cannot be loaded before its parent entry is loaded.
You can also add entries to SDBM (using ldapadd), but this actually results in adding profiles to
RACF. See Chapter 16, “Accessing RACF information,” on page 269 for more information. You cannot
load entries into the GDBM backend. GDBM entries are only created by the LDAP server itself.

4. If GDBM is configured, set an appropriate ACL for controlling access to change log entries in the
GDBM backend. See Chapter 26, “Change logging,” on page 453 for more information.

5. If CDBM is configured, set an appropriate ACL for controlling access to the cn=ibmpolicies suffix
because by default all users have access to that suffix. The ACL set on the cn=configuration suffix
only allows the LDAP administrator authority to that suffix. See Chapter 24, “Advanced replication” for
more information.

6. After initial set up of the LDAP server, it is recommended that you remove the adminPW configuration
option from the LDAP server configuration file. See “Establishing the administrator DN and basic
replication replica server DN and passwords” on page 118 for more information.

Environment variables used by the LDAP server

There are a number of environment variables that are processed by the LDAP server and utilities. Except
for LDAP_DS_ENVVARS_FILE, they can be specified in the LDAP server environment variables file. By
default, the file name is /etc/ldap/ds.envvars. The name can be reset using the
LDAP_DS_ENVVARS_FILE environment variable or, if that is not set, by the ENVVAR DD in the
procedure used to start the LDAP server. Environment variables are read once, during LDAP server
initialization. The LDAP server must be stopped and restarted to put a change to an environment variable
into effect.

Below are some rules for setting up the environment variables file:

* The file must be in code page IBM-1047.
* An environment variable line consists of name=value, starting in column one. Blanks at the end of each
  line are removed, but no other blanks are removed.
* A line can be continued by putting a backslash (\) as the last non-blank character on the line. The
  backslash is removed and the next line is appended to the previous line. The maximum length of the
  initial line plus all continuation lines is 1024 characters.
* The value can be entirely enclosed in quotes ('') or double quotes (""") to include trailing blanks and to
  process a trailing backslash as part of the value. The quotes are removed from the value.
* A line that begins with a pound sign (#) in column one is a comment line and is ignored. A trailing
  backslash (\) on a comment line is ignored and the comment line is not continued.
* If the name corresponds to an environment variable that is already set and value is specified, the new
  value is ignored and the existing value is not changed. If value is not specified, the environment variable
  is deleted.
* Processing continues with the next line if any error occurs.
The list below describes the LDAP server environmental variables:

**GLDLOG_MICROSECONDS=ON | anything_else**

Controls whether all generated activity log records contain microseconds in their time stamps. Microseconds are added to the time stamp if the value is set to ON. Microseconds are not included if the value is not ON or if the environment variable is not specified. See Activity logging for more information.

**GLDLOG_MSG=MSGS | NOMSGS**

Controls whether activity log records are generated when messages are created by the LDAP server. Messages are not written to the log if the environment variable is not specified. See Activity logging for more information.

**GLDLOG_OPS=WRITEOPS | ALLOPS | SUMMARY**

Controls which operations generate LDAP server activity log records. No operations are logged if the environment variable is not specified. See Activity logging for more information.

**GLDLOG_TIME=TIME | NOTIME**

Controls whether LDAP server activity log records are generated when the operation being logged ends. Log records are not generated when an operation ends if the environment variable is not specified. See Activity logging for more information.

**IBMSLAPD_REPL_UPDATE_EXTRA_SECS=**interval

Specifies, in seconds, how long the advanced replication engine waits for a replication operation to complete before setting an LDAP_TIMEOUT error code. By default, the advanced replication engine waits 60 seconds.

**LDAP_ADVREPL_CLEANUP_INTERVAL=**interval

Specifies, in seconds, how often backends participating in advanced replication delete replicated updates. If an incorrect value or 0 is specified, the value is set to 900 (delete replicated updates every 15 minutes). This is also the value used if the environment variable is not specified.

**LDAP_CONSOLE_LEVEL=I | W | E | A**

Specifies the message severity level for sending a message created by the LDAP server to the operator console. Messages with a severity equal to or higher than the specified severity are sent to the operator console in addition to the normal output destination. Note that some LDAP server messages are always written to the operator console and are not affected by this value. Messages with a severity of E or higher are sent to the operator console if the environment variable is not specified.

**LDAP_CTRACE_BUFFSIZE=**size

Sets the amount of storage that will be allocated for CTRACE records within the LDAP server. The minimum value that can be specified is 1024000 bytes. This is also the value that is used if an incorrect value is specified or if the environment variable is not specified. On the average, each CTRACE record is about 120 bytes long. See CTRACE in-memory trace records for more information.

**LDAP_DEBUG=**level

Specifies the needed debug level. The value for LDAP_DEBUG is a mask that you can specify in the following ways:

- A decimal value (for example, 32)
- A hexadecimal value (for example, x20 or X20)
- A keyword (for example, FILTER)
- A construct of these values using plus and minus signs to indicate inclusion or exclusion of a value.

See Table 20 for more information.

**LDAP_DEBUG_FILENAME=**filename | CTRACE_ONLY

Specifies the fully-qualified name of the LDAP debug output file. The debug output is written to
stdout if this environment variable is not specified. The debug file is not used if LDAP debugging
is not active. If using an output file, make sure that the file is not being used for any other
purpose.

The current process identifier is included as part of the debug file name when the name contains a
percent sign (%).

Example: If LDAP_DEBUG_FILENAME is set to /tmp/ldap.%.trc and the current process identifier
is 247, the debug file name is /tmp/ldap.247.trc.

If the value is CTRACE_ONLY, then all debug output is only written to the internal CTRACE
buffers.

LDAP_DS_ENVVARS_FILE=filename
Specifies the name of the name of the LDAP server environment variables file. The file name in
the ENVVAR DD in the start procedure for the LDAP server is used if the environment variable is
not specified. If the ENVVAR DD is not specified, the file name defaults to /etc/ldap/ds.envvars.

LDAP_ERROR_LOGGING=STDOUT | STDERR | BOTH
Specifies how error messages are logged. The following values can be specified:
STDOUT Error messages are written to standard output as specified by the
LDAP_STDOUT_FILENAME environment variable.
STDERR Error messages are written to standard error as specified by the
LDAP_STDERR_FILENAME environment variable.
BOTH Error messages are written to both standard output and to standard error.

Error messages are written to standard error if this environment variable is not specified.

LDAP_NETWORK_POLL=interval
Specifies, in minutes, how often the LDAP server will poll an network interface to determine if it
has failed or has become active. If 0 is specified, the value is set to 5 (poll every 300 seconds).
This is also the value used if the environment variable is not specified.

LDAP_PRINT_CONFIG=1 | anything_else
Controls whether the configuration options used by the LDAP server are displayed when the LDAP
server is started. The options are displayed if the value is 1 and are not displayed for any other
value. The options are displayed if the environment variable is not specified.

LDBM_SHUTDOWN_FAST=1 | anything_else
Controls how a file-based backend (LDBM, CDBM, and file-based GDBM) in the LDAP server
stops. Server shutdown normally frees all storage allocated and held by each backend. This can
potentially be a large amount of storage for LDBM because it holds all its entries in memory,
therefore, can be very time consuming. When the value is set to 1, storage allocated by the LDBM
backend is not released before the LDAP server stops (the storage is eventually released by
MVS). When the value is not 1, the storage is freed before the LDAP server stops. This is also the
processing that occurs if the environment variable is not specified.

LDAP_STDERR_FILENAME=filename
Specifies the fully-qualified name of the file to receive standard error messages generated using
LDAP message services. Messages are written to stderr if this environment variable is not
specified. Make sure that the output file is not being used for any other purpose.

LDAP_STDOUT_FILENAME=filename
Specifies the fully-qualified name of the file to receive standard output messages generated using
LDAP message services. Messages are written to stdout if this environment variable is not
specified. Make sure that the output file is not being used for any other purpose.

LDAP_TDBM_CACHEDELAY=interval
Specifies the number of seconds the LDAP server delays before it examines a TDBM DB2
database to detect out of date caches, rather than checking on every LDAP request. This can be
used to reduce the cost of checking for up to date cache information. The caches affected are the
referral cache, ACL (access control list) caches, and group cache. This environment variable is
only used for a TDBM backend which is running in multiserver mode with a DB2 database that
can be shared with a z/OS Integrated Security Services LDAP server (therefore, the
DB_VERSION value of the database is less than ‘4.0’ or the serverCompatLevel configuration
option is 3).

The valid range for the value is 0 to 2147483647. A value of 0 (zero) causes the DB2 database to
be examined once per request. This is the default behavior if this environment variable is not
specified.

This environment variable should not be set if there are applications which cannot tolerate
temporarily out of date cache information in the LDAP server. It is best suited when most LDAP
operations are search requests and the directory information is mostly static. It can also be useful
even if there are many LDAP update operations, if the updates do not affect referrals, ACLs, or
nested and dynamic groups.

---

**Dynamic debugging**

When the LDAP server is running it is possible to dynamically turn the debugging facility on and off. You
can also replace the current debug levels, add to the current debug levels, or remove from the current
debug levels. The following command can be sent to the LDAP server from the SDSF or the operator’s
console. Note that if the command is entered from SDSF, it must be preceded by a slash (/). In the
command:

```
f dssrv,debug debug_level
```

the `debug_level` is a mask that specifies the needed debug level. See Table 20 for an explanation of the
debug level values.

Debug information will be added to the output associated with the LDAP server.

To turn the debug tracing off, enter the same command providing the value zero (0) for `debug_level`.

---

**CTRACE in-memory trace records**

CTRACE provides an in-memory tracing interface that is common with other z/OS products. The LDAP
CTRACE support captures the following LDAP server output:

- All messages
- All debug message output for the ERROR debug level, regardless of debug level setting
- All debug message output for active debug levels. See Table 20 for more information about debug
  levels

By always capturing ERROR debug output, the CTRACE support provides FFDC (First Failure Data
Capture) to assist in problem debugging without the performance overhead of running with the debug
facility enabled.

The default size of the storage used for the in-memory trace table is 1024000 bytes, which holds about
8000 CTRACE entries. The table wraps when the end is encountered, overwriting the oldest CTRACE
entries. The trace table size can be increased by setting the LDAP_CTRACE_BUFFSIZE environment
variable before the LDAP server is started. See Environment variables used by the LDAP server for more
information. There is no maximum trace table size except that imposed by the availability of system
storage.

The LDAP CTRACE support is initialized during LDAP server startup. If CTRACE initialization fails, an
error message is issued and server startup continues without CTRACE support. The LDAP CTRACE
support cannot be turned off.
When the debug facility is active, debug output by default goes to both the CTRACE in-memory table and the output file indicated by the `LDAP_DEBUG_FILENAME` environment variable (or `stdout` if the environment variable is not set). If `LDAP_DEBUG_FILENAME=CTRACE_ONLY` is set, the debug output is only sent to CTRACE. The LDAP server DEBUG operator modify command can be used while the LDAP server is running to change whether debug output is sent only to CTRACE or to both CTRACE and the debug output file.

To direct debug output to CTRACE only, issue:

```
f dssrv,debug output=memory
```

To direct debug output to both CTRACE and to the debug output file, issue:

```
f dssrv,debug output=both
```

### Viewing LDAP server CTRACE output

The LDAP server CTRACE component name is GLDSRVR. The LDAP server creates a subnode under GLDSRVR, identified by the hex address space identifier (ASID) of the LDAP server. Each LDAP server running on the system has its own subnode. The subnode is deleted when the LDAP server terminates normally. To see the current subnodes, issue the following command from SDSF or the operator console:

```
d trace,comp=gldsrvr,sublevel,n=20
```

The results are similar to:

```
D TRACE,COMP=GLDSRVR,SUBLEVEL,N=20
IEE843I 13.49.52 TRACE DISPLAY
SYSTEM STATUS INFORMATION
ST=(ON,0256K,00512K) AS=ON BR=OFF EX=ON MT=(ON,064K)
TRACENAME
==========
GLDSRVR
MODE BUFFER HEAD SUBS
=====================
OFF HEAD 1
NO HEAD OPTIONS
SUBTRACE MODE BUFFER HEAD SUBS
--------------------------------
ASID(004E) OFF
ASIDS *NOT SUPPORTED*
JOBNAMES *NOT SUPPORTED*
OPTIONS *NONE*
WRITER *NOT SUPPORTED*
```

To view CTRACE records, an SDUMP, SADMP, or SYSMDUMP dump must be performed for the LDAP server address space. The CTRACE records in the resulting dump can be processed by the IPCS CTRACE facility, using the LDAP format table, GLDSCTFT, that resides in SYS1.SIEAMIGE. See [z/OS MVS IPCS Commands](https://www.ibm.com/support资v2帮助资源) for more information about IPCS. Here is an excerpt of the LDAP server CTRACE output (the message lines are wrapped here to fit on the page):

```
ctrace comp(gldsrvr) sub((asid(004e))) full

COMPONENT TRACE FULL FORMAT
COMP(GLDSRVR) SUBNAME((ASID(004E)))
**** 05/20/2005
SYSTYPE MMNEMONIC ENTRY ID TIME STAMP DESCRIPTION
-------- ----------- ----------- -------------- --------------
DCESET3 ERROR 00000001 18:49:47.517060 LDAP_TRACE_ERROR
060615 14:49:47.516866 GLD6012E Unable to open database file /usr/include/LDBM-1.db: 141/05620066 - EDC5141I Read-only file system.
```

---

Chapter 9. Running the LDAP server 139
Displaying performance information and server settings

The LDAP server provides a display operator modify command that is used to display information about the server, including performance information. The output of the display command goes to the operator console.

The syntax of the display command for LDAP is:

```
f dssrv,display report
```

Where report is the name of an LDAP server report, shown below.

The LDAP server provides a reset operator modify command that is used to reset the statistics shown in some of the reports.

The syntax of the reset command for LDAP is:

```
f dssrv,reset report
```

Where report is the name of an LDAP server report, shown below.

The following is a description of each of the LDAP server reports. Where applicable, the reset processing is also described.

**AUDIT**

This option shows the current audit settings. It shows if auditing is ON or OFF and what level of auditing is active for each event.

Following is sample output if you request an AUDIT report:

```
GLD1190I Audit status
Option     Setting
AUDIT      ON
ERROR      BNGD
ALL         ADD CONNECT DELETE DISCONNECT MODIFY SEARCH
NONE       MODIFYDN
```

**BACKENDS**

This option shows the current state of all configured GDBM, LDBM, CDBM, SDBM, and TDBM backends.

The state is composed of two parts:

- The first part indicates the backend allows entries be updated (READWRITE) or only read (READONLY).
- The second part indicates the backend processes requests (AVAILABLE), is not available because DB2 is down (UNAVAILABLE), or is disabled because it could not load its directory (DISABLED).

The state is set to FAILED CONFIGURATION if the backend fails to start.

Following is sample output if you request a BACKENDS report:

```
GLD1224I Backend status
Backend   State
mycdbm    READONLY,AVAILABLE
GDBM-001  READWRITE,AVAILABLE
TDBM-002  READWRITE,UNAVAILABLE
mTDBM     FAILED CONFIGURATION
Dept1LDBM READONLY,AVAILABLE
Dept2LDBM READWRITE,DISABLED
SDBM-003  READWRITE,AVAILABLE
```

**DEBUG**

This option shows the current DEBUG setting.
Following is sample output if you request a DEBUG report with debug set to THREAD+PERF+TDBM:

GLD1226I Debug settings  
THREAD  
PERFORMANCE  
TDBM

**LEVEL**  
This option shows the current server version, including service level.

Following is sample output if you request a LEVEL report:

GLD1001I LDAP server version 3.21, Service level OA17247.

**LOCKS**  
This option shows current server lock contention. The output indicates how often there has been contention on a lock and the average duration of contention (in microseconds). If a lock is currently held, the output indicates what function and thread is holding the lock and how long the lock has been held.

Following is sample output if you request a LOCKS report:

GLD1226I Server lock statistics

<table>
<thead>
<tr>
<th>Lock</th>
<th>Shared</th>
<th>Exclusive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity Log</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SRV WLM</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SRV Operations Monitor</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Schema - DN Cache</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>DN Cache</td>
<td>26</td>
<td>60</td>
</tr>
<tr>
<td>LDBM Replicate</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>LDBM - Filter Cache</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>LDBM Filter Cache</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>LDBM Database</td>
<td>65</td>
<td>69</td>
</tr>
<tr>
<td>LDBM Database</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>TDBM - DN to EID Cache</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>TDBM - Entry Cache</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>TDBM - Filter Cache</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>TDBM - Entry Owner Cache</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>TDBM - ACL Source Cache</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>TDBM Replication</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>TDBM Group Cache</td>
<td>674</td>
<td>21</td>
</tr>
</tbody>
</table>

The lock statistics can be reset to 0 by using the RESET LOCKS command.
**MAINTMODE**  This option shows if maintenance mode is currently ON or OFF for the LDAP server.

Following is sample output if you request a **MAINTMODE** report:

GLD1225I Maintenance Mode status
Maintenance mode OFF

**MONITOR**  This option shows the monitor report. This report provides operational statistics for both the server and for each backend that successfully started, including information about the number of specific operations targeted at the LDAP server or to a given backend. This information can be used to better tune backend caches for improved performance.

The operations monitor statistics are not returned in this report. These statistics can be retrieved by performing an LDAP search of the operations monitor. See "Monitoring performance with cn=monitor" on page 497 for more information.

Following is sample output if you request a **MONITOR** report:

Monitor Statistics
------------------

Server LDAP Server
Current Time: Mon Sep 22 16:30:13.826724 2008
Number of Resets: 0

Server Totals:
--------------

<table>
<thead>
<tr>
<th>Description</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Config Max Connections</td>
<td>24982</td>
</tr>
<tr>
<td>System Max Connections</td>
<td>25000</td>
</tr>
<tr>
<td>Total Connections</td>
<td>20351</td>
</tr>
<tr>
<td>Current Connections</td>
<td>1</td>
</tr>
<tr>
<td>MaxReached Connections</td>
<td>15</td>
</tr>
<tr>
<td>Search Entries Sent</td>
<td>4156</td>
</tr>
<tr>
<td>Message Bytes Sent</td>
<td>1564644k</td>
</tr>
<tr>
<td>Search References Sent</td>
<td>0</td>
</tr>
</tbody>
</table>

Operation Requested Completed
-----------------------------
| Abandons | 0 | 0 |
| Adds     | 2288 | 2288 |
| AllOps   | 61052 | 61052 |
| Binds    | 20351 | 20351 |
| Compares | 0  | 0  |
| Deletes  | 2190 | 2190 |
| ExtOps   | 0  | 0  |
| Modifies | 11284 | 11284 |
| ModifyDNs| 433  | 433  |
| Searches | 4156 | 4156 |
| Unbinds  | 20350 | 20350 |
| Unknown  | 0  | 0  |

Backend Statistics: TDBM1
--------------------------
| Naming Context: C=CA   |
| Naming Context: C=TDBM |
| Naming Context: CN=MOVER |
| Naming Context: CN=MOVING |

Description Count
------------------
| Search Entries Sent | 476 |
| Message Bytes Sent  | 88467 |
| Search References Sent | 0 |
### Backend Statistics: LDBM1

<table>
<thead>
<tr>
<th>Cache Name</th>
<th>Size</th>
<th>Entries</th>
<th>Hits</th>
<th>Misses</th>
<th>Pct</th>
<th>Refresh Count</th>
<th>Refresh AvgEnts</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACLSource</td>
<td>100</td>
<td>1</td>
<td>2518</td>
<td>1</td>
<td>100.0%</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>DnToEID</td>
<td>1000</td>
<td>565</td>
<td>162090</td>
<td>3438</td>
<td>97.9%</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Entry</td>
<td>5000</td>
<td>566</td>
<td>316664</td>
<td>1050</td>
<td>99.7%</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>EntryOwner</td>
<td>100</td>
<td>1</td>
<td>2518</td>
<td>1</td>
<td>100.0%</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Filter</td>
<td>5000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0%</td>
<td>2037</td>
<td>0</td>
</tr>
</tbody>
</table>

Filter Bypass Limit: 100

### Backend Statistics: SDBM

<table>
<thead>
<tr>
<th>Cache Name</th>
<th>Size</th>
<th>Entries</th>
<th>Hits</th>
<th>Misses</th>
<th>Pct</th>
<th>Refresh Count</th>
<th>Refresh AvgEnts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter</td>
<td>5000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0%</td>
<td>16195</td>
<td>0</td>
</tr>
</tbody>
</table>

Filter Bypass Limit: 100

### Naming Context: CN=MYRAF

<table>
<thead>
<tr>
<th>Description Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search Entries Sent</td>
</tr>
<tr>
<td>Message Bytes Sent</td>
</tr>
<tr>
<td>Search References Sent</td>
</tr>
</tbody>
</table>

### Operation Requested Completed

<table>
<thead>
<tr>
<th>Operation</th>
<th>Requested</th>
<th>Completed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abandons</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Adds</td>
<td>2288</td>
<td>2288</td>
</tr>
<tr>
<td>AllOps</td>
<td>20349</td>
<td>20349</td>
</tr>
<tr>
<td>Binds</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Compares</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Deletes</td>
<td>2190</td>
<td>2190</td>
</tr>
<tr>
<td>ExtOps</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ModifyDNs</td>
<td>433</td>
<td>433</td>
</tr>
<tr>
<td>Searches</td>
<td>4154</td>
<td>4154</td>
</tr>
<tr>
<td>Unbinds</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Unknown</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### Cache Name: Filter

<table>
<thead>
<tr>
<th>Cache Name</th>
<th>Size</th>
<th>Entries</th>
<th>Hits</th>
<th>Misses</th>
<th>Pct</th>
<th>Refresh Count</th>
<th>Refresh AvgEnts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter</td>
<td>5000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0%</td>
<td>16195</td>
<td>0</td>
</tr>
</tbody>
</table>

Filter Bypass Limit: 100

### Naming Context: C=AU

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search Entries Sent</td>
</tr>
<tr>
<td>Message Bytes Sent</td>
</tr>
<tr>
<td>Search References Sent</td>
</tr>
</tbody>
</table>

### Operation Requested Completed

<table>
<thead>
<tr>
<th>Operation</th>
<th>Requested</th>
<th>Completed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abandons</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Adds</td>
<td>314</td>
<td>314</td>
</tr>
<tr>
<td>AllOps</td>
<td>2519</td>
<td>2519</td>
</tr>
<tr>
<td>Binds</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Compares</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Deletes</td>
<td>305</td>
<td>305</td>
</tr>
<tr>
<td>ExtOps</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ModifyDNs</td>
<td>46</td>
<td>46</td>
</tr>
<tr>
<td>Searches</td>
<td>476</td>
<td>476</td>
</tr>
<tr>
<td>Unbinds</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Unknown</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### Cache Name: ACLSource

<table>
<thead>
<tr>
<th>Cache Name</th>
<th>Size</th>
<th>Entries</th>
<th>Hits</th>
<th>Misses</th>
<th>Pct</th>
<th>Refresh Count</th>
<th>Refresh AvgEnts</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACLSource</td>
<td>100</td>
<td>1</td>
<td>2518</td>
<td>1</td>
<td>100.0%</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Filter Bypass Limit: 100

### Naming Context: C=LDBM

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search Entries Sent</td>
</tr>
<tr>
<td>Message Bytes Sent</td>
</tr>
<tr>
<td>Search References Sent</td>
</tr>
</tbody>
</table>

### Operation Requested Completed

<table>
<thead>
<tr>
<th>Operation</th>
<th>Requested</th>
<th>Completed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abandons</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Adds</td>
<td>314</td>
<td>314</td>
</tr>
<tr>
<td>AllOps</td>
<td>2519</td>
<td>2519</td>
</tr>
<tr>
<td>Binds</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Compares</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Deletes</td>
<td>305</td>
<td>305</td>
</tr>
<tr>
<td>ExtOps</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ModifyDNs</td>
<td>46</td>
<td>46</td>
</tr>
<tr>
<td>Searches</td>
<td>476</td>
<td>476</td>
</tr>
<tr>
<td>Unbinds</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Unknown</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### Cache Name: ACLSource

<table>
<thead>
<tr>
<th>Cache Name</th>
<th>Size</th>
<th>Entries</th>
<th>Hits</th>
<th>Misses</th>
<th>Pct</th>
<th>Refresh Count</th>
<th>Refresh AvgEnts</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACLSource</td>
<td>100</td>
<td>1</td>
<td>2518</td>
<td>1</td>
<td>100.0%</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Chapter 9. Running the LDAP server 143
<table>
<thead>
<tr>
<th>Operation</th>
<th>Requested</th>
<th>Completed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abandons</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Adds</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AllOps</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Binds</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Compares</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Deletes</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ExtOps</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Modifies</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ModifyDNs</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Searches</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Unbinds</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Unknown</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Backend Statistics: Monitor**

**Naming Context:** CN=MONITOR

<table>
<thead>
<tr>
<th>Description</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search Entries Sent</td>
<td>2</td>
</tr>
<tr>
<td>Message Bytes Sent</td>
<td>2306</td>
</tr>
<tr>
<td>Search References Sent</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Operation</th>
<th>Requested</th>
<th>Completed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abandons</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Adds</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AllOps</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Binds</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Compares</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Deletes</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ExtOps</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Modifies</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ModifyDNs</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Searches</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Unbinds</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Unknown</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Backend Statistics: Schema**

**Naming Context:** CN=SCHEMA

<table>
<thead>
<tr>
<th>Description</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search Entries Sent</td>
<td>0</td>
</tr>
<tr>
<td>Message Bytes Sent</td>
<td>0</td>
</tr>
<tr>
<td>Search References Sent</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Operation</th>
<th>Requested</th>
<th>Completed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abandons</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Adds</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AllOps</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Binds</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Compares</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Deletes</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ExtOps</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Modifies</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ModifyDNs</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Searches</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Unbinds 0 0
Unknown 0 0

<table>
<thead>
<tr>
<th>Cache Name</th>
<th>Size</th>
<th>Entries</th>
<th>Hits</th>
<th>Misses</th>
<th>Pct</th>
<th>Refresh Count</th>
<th>Refresh AvgEnts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dn</td>
<td>1000</td>
<td>1000</td>
<td>121579</td>
<td>21620</td>
<td>84.9%</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Backend Statistics: RootDSE

<table>
<thead>
<tr>
<th>Description</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search Entries Sent</td>
<td>0</td>
</tr>
<tr>
<td>Message Bytes Sent</td>
<td>0</td>
</tr>
<tr>
<td>Search References Sent</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Operation</th>
<th>Requested</th>
<th>Completed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abandons</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Adds</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AddOps</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Binds</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Compares</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Deletes</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ExtOps</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Modifies</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ModifyDNs</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Searches</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Unbinds</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Unknown</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Note:** Some statistics may exceed the width of the report columns above. In this case, a suffix is appended indicating the number is expressed in larger units. These suffixes are as follows:
- **k** - kilo - displayed number times 10(3)
- **M** - mega - displayed number times 10(6)
- **G** - giga - displayed number times 10(9)
- **T** - tera - displayed number times 10(12)
- **P** - peta - displayed number times 10(15)

The monitor statistics can be reset by using the `RESET MONITOR` command, as described below. The reset statistics are reflected in future `DISPLAY MONITOR` reports and `cn=monitor` searches. `RESET MONITOR` has no effect on the values reported in the Activity Log. When monitor statistics are reset:

- **Last Reset Time** is set to **Current Time**, **Number of Resets** is incremented, and **MaxReached Connections** is set to the value of **Current Connections**.
- **Search Entries Sent**, **Message Bytes Sent**, **Search References Sent**, and all Operation Requested and Completed values are reset to zero for both the **Server Totals** and **Backend Statistics**.
- **Cache statistics** **Hits**, **Misses**, **Pct Hit**, **Refresh Count**, and **Refresh AvgEnts** for each cache are reset to zero.

The `RESET MONITOR` command also resets operations monitor statistics. See "[Monitoring performance with cn=monitor]" on page 497 for more information.

Resetting the monitor statistics has no effect on the Cache **Size** for each cache, nor on the **Filter Bypass Limit**, because these are configured values. It also has no effect on the Cache **Entries** for each cache, because the contents of the caches are not altered by a reset of statistics.
Some caches might get invalidated and refreshed because of directory update operations. When this occurs, Cache **Refresh Count** is incremented and the Cache **Entries** is set to zero to reflect the refreshed, or empty, cache. The Cache statistics **Hits**, **Misses**, and **Pct Hit** values are accumulated across cache invalidation and refresh.

The same statistics in this report can also be obtained by using **ldapsearch** to do a subtree search of the **cn=monitor** entry.

See **Monitoring performance with cn=monitor** for more information about searching using **cn=monitor**.

**NETWORK**

This option shows the current state of all configured network interfaces.

Following is sample output if you request a **NETWORK** report:

```
GLD1084I Network interface status
Interface        Port  Status
127.0.0.1        3389  ACTIVE
127.0.0.1        389   ACTIVE
127.0.0.1        1492  ACTIVE
9.12.47.95       3389  ACTIVE
9.12.47.95       389   ACTIVE
9.12.47.95       1492  ACTIVE
```

**REPLICAS**

This option shows the current state of each basic replication replica. The report indicates how many updates are currently in each replication queue, how many updates have been successfully replicated, and how many failed replication attempts have been set aside. This information can be used to determine if replication has stalled for a certain replica, and, if it has, can determine how many updates are left to process for that replica. If advanced replication is configured, this command does not display the status of any configured consumer servers.

Following is sample output if you request a **REPLICAS** report:

```
GLD1163I Replication status
Backend   Replica               Queued/total/Set Aside
TDBM1     ldap2.ibm.com:389    10 /134 /0
TDBM1     ldap1.ibm.com:389    0  /134 /0
```

**THREADS**

This option shows the current activity statistics for the communication threads in the LDAP server. These threads are used to handle the network communications between the LDAP server and clients and to process LDAP operations requested by the clients.

The values returned for a **THREADS** report are:

- **commThreads defined**
  - number of communication threads. This is the value of the **commThreads** option in the LDAP server configuration file.

- **current active commThreads**
  - number of communication threads currently active

- **max concurrent commThreads**
  - largest number of communication threads that have been active at the same time

- **average thread activity**
  - percentage of time that the communication threads have been active

- **current active LDAP operations**
  - number of communication threads that are currently processing LDAP operations

- **max concurrent LDAP operations**
  - largest number of communication threads that have been processing LDAP operations at the same time

- **LDAP operations per second**
  - average number of LDAP operations processed per second
client network connections per second
average number of LDAP client connections processed per second

Following is sample output if you request a **THREADS** report:
GLD1109I Server activity statistics
20 commThreads defined
2 current active commThreads
16 max concurrent commThreads
20.59% average thread activity
2 current active LDAP operations
15 max concurrent LDAP operations
39.66 LDAP operations per second
13.22 client network connections per second

These statistics, except for the 'commThreads defined’ value, are reset to 0 using the **RESET THREADS** command.

**XCF**
This option shows the current state of all servers in the sysplex group. This report indicates which server is the group owner and what servers are currently active in the group.

Following is sample output if you request an **XCF** report:
GLD1135I Sysplex status
Server System Status
DCEIMGVV0066-BF31AD8E-2EFB4F4E DCEIMGVV0066 ACTIVE/REPLICA
DCEIMGVV004E-BF31AD83-ACE95202 DCEIMGVV004E ACTIVE/OWNER

The System name for an LDAP server in a sysplex group is displayed in message GLD1146I when the server is started.

**Size limitations**
If any of the above reports exceeds 256 lines of output, the report is truncated and the following message is displayed as the last line of the output:
GLD1086I Maximum number of lines displayed.

If the **MONITOR** report is truncated, the complete output can be found in the LDAP server job log. The same statistics can also be obtained by using **ldapsearch** to do a subtree search of the **cn=monitor** entry. See [Monitoring performance with **cn=monitor**](#) for more information about searching **cn=monitor**.

**Activity logging**
- The LDAP server can record server activity for the purpose of load analysis. The activity log file can contain information about operations handled by the server, client IP addresses, messages generated by the server, and summary statistics. Client requests handled by a plug-in are not logged in the activity log file. In general, each record consists of a time stamp followed by the record data. Operations are logged when they successfully begin processing. Optionally, a log record can be created when the operation completes processing. The following is an example of a typical operation log record followed by the ending log record.

  Wed May 21 11:53:52 2008 Search: connid = 4, base = cn=monitor, filter = (objectclass=*)
  scope = 2, attrs = , IP = 1.2.3.4
  Wed May 21 11:53:52 2008 End Search: connid = 4, base = cn=monitor, filter = (objectclass=*)
  scope = 2, count = 7, rc = 0, IP = 1.2.3.4

- Note that when the client is connecting to the LDAP server over the PC interface, the IP address will be reported as 'PC'. If a request is handled by the cross-system facility (XCF), the IP address will be reported as 'XCF'.

Log records created for messages appear in the log file with the time stamp followed by the message. The following is an example of a message log record.

  Wed May 21 11:30:49 2008 GLD0210I Modify command has been processed successfully: LOG,MSGS

---

Chapter 9. Running the LDAP server 147
Summary records are created on an hourly basis if log records are being collected or when an LDAP server LOG operator modify command is processed. The summary log records contain information about the operations that the server has processed. The following is an example of the summary log records.

Wed May 21 11:45:25 2008 total operations started = 3662306
Wed May 21 11:45:25 2008 total operations completed = 3662304
Wed May 21 11:45:25 2008 total abandons completed = 0
Wed May 21 11:45:25 2008 total adds completed = 0
Wed May 21 11:45:25 2008 total binds completed = 1245
Wed May 21 11:45:25 2008 total compares completed = 0
Wed May 21 11:45:25 2008 total deletes completed = 0
Wed May 21 11:45:25 2008 total extendedops completed = 0
Wed May 21 11:45:25 2008 total modifies completed = 47
Wed May 21 11:45:25 2008 total modifydns completed = 0
Wed May 21 11:45:25 2008 total searches completed = 3661012
Wed May 21 11:45:25 2008 total unbinds completed = 0
Wed May 21 11:45:25 2008 total unknown ops completed = 0
Wed May 21 11:45:25 2008 total search entries sent = 4123452
Wed May 21 11:45:25 2008 total search references sent = 0
Wed May 21 11:45:25 2008 total bytes sent = 572465214
Wed May 21 11:45:25 2008 total connections processed = 2
Wed May 21 11:45:25 2008 current connections = 2
Wed May 21 11:45:25 2008 connection high water mark = 2

The location of the log file is controlled by the `logfile` option in the LDAP server configuration file. The activity log can be written to a z/OS UNIX System Services file or to an MVS dataset. If the log file is an MVS dataset, it must be created (allocated) before its use by the LDAP server. It is recommended that when a z/OS UNIX System Services file is needed that the file specification be fully qualified. The following is an example of a `logfile` option specifying a z/OS UNIX System Services file.

```plaintext
logfile /etc/ldap/ldap.activity.log
```

`/etc/ldap/gldlog.output` is the default location of the log file.

The MVS dataset can be specified through either a DDNAME or as a specific dataset. The following is an example for both methods.

```plaintext
logfile //dd:logout
logfile //'mysys.ldap.actlog'
```

The default data collection setting is to collect no data. The default is modified by either environment variables or through the usage of the LDAP server LOG operator modify command. The environment variables are read as the server starts up while the LOG operator modify command can be specified once the server has started.

An operator modify command can be sent to the LDAP server from the SDSF or the operator's console. Note that if the command is entered form SDSF, it must be preceded by a slash (/). The format of the LDAP server LOG operator modify command is:

```plaintext
f dssrv,log writeops | allops | summary | time | notime | msgs | nomsgs | flush | stop
```

The `GLDLOG_OPS` environment variable and the LOG operator modify command can be used to control which operations generate log records. Before specifying an operation setting, no operation logging is performed. The settings are:

- **writeops** indicates that log records are to be created at the start of add, delete, modify, modrdn and extended operations.
- **allops** indicates that log records are to be created as for **writeops** plus the start of, bind, search, compare, and unbind operations are to be logged.
- **summary** indicates that only hourly summary statistics are to be logged. If any logging is being collected, summary data is produced on an hourly basis.
The **GLDLOG_TIME** environment variable and the **LOG** operator modify command can be used to control whether log records are generated when logged operations end. The settings are:

- **time** indicates that ending log records are to be created for all operations that are being logged.
- **notime** indicates that ending log records are not to be created for operations that are being logged.
  
  This is the default before specifying a time setting.

The **GLDLOG_MICROSECONDS** environment variable controls whether all generated log records contain microseconds in their time stamps. This setting cannot be modified by a **LOG** operator modify command. The default is to not include microseconds in the time stamps. The setting to activate microseconds is:

- **ON** indicates that all activity log time stamps will include microseconds.

The following are examples of activity log records containing microseconds:

- **Wed May 21 11:53:52.113408 2008 Search: connid = 4, base = cn=monitor, filter = (objectclass=*)**, `scope = 2, attrs = , IP = 1.2.3.4`

- **Wed May 21 11:30:49.240021 2008 GLD0210I Modify command has been processed successfully: LOG,MSGS**

The **GLDLOG_MSG** environment variable and the **LOG** operator modify command can be used to control whether log records are generated when messages are created by the LDAP server. The settings are:

- **msgs** indicates that messages generated by the LDAP server are to be written to the activity log in addition to the normal target.
- **nomsgs** indicates that messages generated by the LDAP server are not to be written to the activity log.
  
  This is the default before specifying a messages setting.

As the log records are produced, some buffering of the output is performed by the system. The buffers are flushed before the server shuts down. However, you can force the server to flush the buffers by issuing the following **LOG** operator modify command:

- **f dssrv,log flush**

The server can be told to stop collecting activity data by issuing the following **LOG** operator modify command:

- **f dssrv,log stop**

Activity logging can be started again by specifying a **LOG** operator modify command with a new setting.

### LDAP SMF auditing

The LDAP server can be configured to generate SMF type-83 subtype 3 audit records. The SMF type-83 log records containing LDAP events can be unloaded by using the RACF SMF data Unload utility for further analysis by auditing tools. These audit records contain information provided on LDAP client operation requests. The LDAP server can be configured to write audit records when the operation successfully completes, when the operation fails or for either case. SMF type-83 subtype 3 audit records are not created by the LDAP server when a request is handled by a plug-in. The LDAP server uses the **R_audix** callable service to write the record to SMF. See [Additional setup for generating audit records](#) for setup information.

### Auditing events

Auditing of LDAP operations can be set up by using the **audit** option in the LDAP server configuration file. See page 77 for more information.

While the LDAP server is running, auditing can be turned on or off and the specifications of which operations are to be audited and their associated audit level can be changed using the LDAP server **AUDIT** operator modify command. The format of the **AUDIT** operator modify command is:

- **f dssrv,audit on | off | all,operations | error,operations | none,operations**
When auditing is on, an LDAP SMF type 83 subtype 3 audit record is generated for an operation if the operation is specified on an audit level and the operation result matches the audit level.

A separate audit configuration option or AUDIT operator modify command must be issued to turn auditing on or off and to set each audit level. Multiple operations can be specified for a level by either putting a + between them on the audit option or AUDIT command, or by specifying multiple audit options or AUDIT commands with the same level.

Operations can be audited all the time or only when they fail. The following audit levels are supported:

- **all** An LDAP audit record will be generated for the specified operations.
- **error** An LDAP audit record will be generated for the specified operations when they fail.
- **none** An LDAP audit record is not generated for the specified operations.

The supported values for operations can be one or more of: add, bind, compare, connect, delete, disconnect, exop, modify, modifydn, search, unbind

If an operation is specified in more than one level, the last level is used for the operation. If an operation is not specified in any level, the level defaults to none for that operation. Turning auditing off does not change the setting for the audit levels. If auditing is later turned on, the audit levels will remain as they last were.

For example, the following AUDIT operator modify commands turn auditing on for modify and search operation failures and for all bind operations. The other operations are not audited.

```plaintext
f dssrv,audit error,modify+search+bind
f dssrv,audit all,bind
f dssrv,audit on
```

The current audit settings can be displayed using the following LDAP server DISPLAY operator modify command:

```plaintext
f dssrv,display audit
```

The results for the AUDIT operator modify commands issued above are:

```
GLD1190I Audit status
Option Setting
AUDIT ON
ERROR MODIFY SEARCH
ALL BIND
NONE
```

### Working with audit records

The LDAP events are logged in an SMF dataset as type 83 subtype 3 records. The audit record is a mixture of binary and EBCDIC data. The general format of SMF type 83 records is described in [z/OS Security Server RACF Macros and Interfaces](https://www.ibm.com/support/knowledgecenter/SSLTBK_2.4.0/com.ibm.zos.msk2.sesref/r000056.htm) in the topic for SMF records, Record type 83: Security events.

The RACF SMF Unload utility can be used to reformat the LDAP SMF type 83 subtype 8 records for easier analysis. These audit records can be reformatted in the following different forms:

- A tabular format, suitable for import to a relational database manager.
- eXtensible Markup Language (XML) documents
Three samples are shipped with the LDAP server to use with the RACF SMF Unload utility to reformat the LDAP audit records. The samples can be found in GLDHLQ.SGLDSAMP.

Two samples are used to create the tabular format. LDAPDBTB defines the DB2 table spaces and tables needed to contain the LDAP audit records and LDAPDBLD loads the formatted audit records into the tables.

A single sample is used to create the XML format. GLDLDPSC provides an XML schema to define the XML tag language used to reformat the LDAP audit records.

For the format and content of the LDAP SMF Audit records and the RACF SMF unload utility tabular description, see Appendix E, “SMF records.”

### Monitoring LDAP server resources

The LDAP server monitors the basic resources that it uses to ensure that they are still available. If a resource becomes unavailable, the LDAP server can be configured to either end or to operate without the resource until the resource becomes available.

#### Server backends and plug-ins during startup

When the LDAP server is started, the server processes the LDAP server configuration file and then initializes each of the backends or plug-ins that are configured. If an error is detected during initialization of a backend or plug-in, that backend or plug-in is not usable. Based on the `srvStartUpError` option in the LDAP server configuration file, the LDAP server either shuts itself down or continues running with the those backends or plug-ins that successfully start. After the problem encountered during backend or plug-in initialization is fixed, the LDAP server must be restarted to make that backend or plug-in available. See Chapter 8, “Customizing the LDAP server configuration” for the description of the `srvStartUpError` configuration option. The option does not apply to resource problems encountered after the LDAP server backends or plug-ins have started. The LDAP server response to those problems is described below.

#### DB2

The LDAP server uses DB2 to store the directory information for the TDBM and GDBM (when DB2-based) backends. If one of these backends is configured, the LDAP server attempts to connect to DB2 when the server is started. If DB2 is not available, the LDAP server, based on the `db2StartUpRetryLimit` and `db2StartUpRetryInterval` configuration options in the LDAP configuration file, can periodically try again to connect to DB2. If a DB2 connection cannot be established, the LDAP server can, based on the `srvStartUpError` configuration option, shut itself down or continue running without access to TDBM and GDBM (when DB2-based). See page 81 for the descriptions of the `db2StartUpRetryLimit`, `db2StartUpRetryInterval`, and `srvStartUpError` configuration options.

When the LDAP server has connected to DB2 and a TDBM or DB2-based GDBM backend is running, the LDAP server monitors DB2 and detects when it shuts down. Based on the `db2Terminate` option in the LDAP server configuration file, the LDAP server shuts itself down or continues running without access to the TDBM and GDBM backends and reconnects to DB2 when it becomes available. See page 81 for the description of the `db2Terminate` configuration option.

#### Network communications

The LDAP server uses TCP/IP for its client communications, except for applications using the LDAP Program Call support. The LDAP server also monitors TCP/IP and detects when one of the network interfaces in use by the server has failed. Based on the `tcpTerminate` option in the LDAP server
configuration file, the LDAP server can then either shut itself down or try to reestablish the failed interface. See page 106 for the description of the **tcpTerminate** configuration option.

The **tcpTerminate** option also controls the LDAP server response to a failure when initializing the SSL and Kerberos interfaces if they have been configured. The LDAP server either terminates or continues processing but does not use the failed interface. After the problem is fixed, the LDAP server must be restarted to make that interface available.

Similarly, the **srvStartUpError** option also covers a failure when initializing the LDAP PC callable support interface if that has been configured. The LDAP server either terminates or continues processing but does not use the PC support. After the problem is fixed, the LDAP server must be restarted to make the PC interface available. Note that only one LDAP server on a system can try to activate the PC interface. If another LDAP server has already tried to initialize (successfully or unsuccessfully) the PC interface, stop that server before starting this server with PC support configured.

### Client connections

As the number of concurrent client connections approaches the maximum number of client connections allowed on the LDAP server, the LDAP server issues warning messages to the console when additional clients attempt to bind to the LDAP server. To avoid overloading the console with messages, these warning messages are issued, at most, once per minute for a maximum of 60 times while the number of concurrent client connections remains at a high level. If the number of concurrent client connections on the LDAP server falls below a safe threshold, another console message is issued stating that the number of concurrent client connections is now at a safe level. After this, the cycle of warning messages can begin again if the number of concurrent client connections again approaches the maximum number of connections allowed on the LDAP server.

The issuance of these console warning messages on a fairly regular basis might signify that the **maxConnections** option in the LDAP server configuration file is set to a low value and should be increased. It is also possible that the MAXFILEPROC statement or the MAXSOCKETS option on the NETWORK statement within BPXPRMxx may need to be adjusted upward to support a higher value of **maxConnections**. The activity log on the LDAP server can be used to monitor the number of client connections. See Activity logging for more information about activity logging.

Another way of controlling the number of clients concurrently connected to the LDAP server is by using the **idleConnectionTimeout** option in the LDAP server configuration file. This option controls the amount of time that the LDAP server allows a client connection to remain connected to the server when the client is not actively sending or receiving data on the connection. When the **idleConnectionTimeout** setting has been exceeded, the LDAP server severs the idle LDAP client connection, therefore, freeing the resource for a new client connection. See page 85 for more information.

### File system

The LDAP server uses the z/OS UNIX System Services file system to store the directory information for the LDBM, CDBM, GDBM (file-based) backends. The LDAP server detects when file system errors occur, such as no space available or inability to write to required files or file directories. Based on the **fileTerminate** option in the LDAP server configuration file, the LDAP server can then either shut itself down or continue running with the affected LDBM, GDBM, or CDBM backend in read-only mode (updates to the directory are rejected). When the file system problem has been dealt with, the operator can use the LDAP server **BACKEND operator modify command to change the LDBM, GDBM, or CDBM backend back to read-write mode. See Chapter 8, “Customizing the LDAP server configuration” for the description of the **fileTerminate** configuration option.

### LDAP server abnormal termination

The LDAP server registers itself with the z/OS Automatic Restart Management (ARM) service if the **armName** option is specified in the LDAP server configuration file. ARM can be set up to automatically
restart the LDAP server if it stops unexpectedly. See page 76 for the description of the armName configuration option. See z/OS MVS Setting Up a Sysplex for more information about ARM.

**LDAP server operator commands**

The following LDAP server commands can be entered using the operator modify command. For example, if the LDAP server started task is DSSRV, the operator command is:

```
f dssrv, cmd-name cmd-options
```

The list below describes the supported LDAP server operator commands and their command options:

**AUDIT audit_controls**
- Turn LDAP server auditing on or off and control what server activities result in creating an audit record. See LDAP SMF auditing for more information about controlling LDAP server usage of audit.

**BACKEND backendName=RDWR | READ**
- Change a specific backend to read-write (normal) mode or read-only mode. The LDAP server can place a file-based backend into read-only mode if the LDAP server cannot access its checkpoint or database files. After correcting the access problem, the operator can use this command to reset the backend to read-write mode.

**COMMIT**
- Force all the file-based backends to commit their changes to the database files by merging in the changes from the checkpoint files. The changes are removed from the checkpoint files. This can be done to prevent the checkpoint files from growing too large.

**DEBUG level | OUTPUT=MEMORY | OUTPUT=BOTH**
- Set the level of debugging. See Dynamic debugging for more information about setting the debug levels. The command can also control where debug output is sent: to just the internal CTRACE table or to the CTRACE table and the normal debug output destination.

**DISPLAY AUDIT | BACKENDS | DEBUG | LEVEL | LOCKS | MAINTMODE | MONITOR | NETWORK | REPLICAS | THREADS | XCF**
- Display a variety of information about the LDAP server. See Displaying performance information and server settings for a description of the DISPLAY output.

**LOG log_control**
- Turn LDAP server activity logging on or off and control what server activities are logged. See Activity logging for more information.

**MAINTMODE ON | OFF**
- Change the LDAP server between normal mode and maintenance mode. Access to the LDAP server is restricted to certain users when in maintenance mode, therefore, this mode can be used to “fix” the LDAP server. See Basic replication maintenance mode for more information about basic replication maintenance mode. Also, see Advanced replication maintenance mode for more information about advanced replication maintenance mode.

**REFRESH DB2RUNSTATS | SSL**
- When DB2RUNSTATS is specified, refresh the TDBM or GDBM (DB2-based) database statistics. This causes the LDAP server to reexamine the current database statistics recorded in the DB2 catalog by the RUNSTATS utility. This is useful if the catalog statistics have been changed by running the RUNSTATS utility after the LDAP server completed initialization. See Chapter 29, “Performance tuning” for detailed information about running the DB2 RUNSTATS utility.

When SSL is specified, initialize the SSL environment again. This might be necessary, for example, if SSL replaces expired certificates.
If LOCKS, MONITOR, or THREADS are specified, then various counters maintained by the LDAP server are reset. See Displaying performance information and server settings for the counters that are reset.

If WLMEXCEPT is specified without an opid (Operations Monitor ID), the LDAP server defaults to using only the initial wlmExcept configuration option settings. If an opid is specified, this command will undo all previous WLMEXCEPT operator modify commands issued for this particular opid. The LDAP server defaults to using the wlmExcept configuration options for this particular opid only.

SNAP  Take a SNAP dump of the LDAP server. The dump is taken using the following options:
THR(ALL)  TRACE  NOFILE  NOVAR  NOBLOCK  NOSTOR  FNAME(CEEDUMP)
This command is only valid when the LDAP server is started in 31-bit mode.

WLMEXCEPT  tname,opid
Associate a search pattern in the operations monitor entry to a WLM transaction name.

The tname specifies the WLM transaction name while the opid specifies the Operations Monitor ID value from the operations monitor entry. The opid indicates which search pattern should be used from the operations monitor entry. This command causes any client search request matching the specified search pattern to run under the enclave with the specified WLM transaction name. Each time the WLMEXCEPT operator modify command is issued, the new association is added before any of the configured wlmExcept configuration options or previously issued WLMEXCEPT operator modify commands. See Workload manager (WLM) for more information about running the LDAP server with WLM.
Chapter 10. Migrating to z/OS IBM TDS

This topic discusses migration issues. Your plan for migrating to z/OS IBM TDS should include information from a variety of sources. These sources of information describe topics such as coexistence service and optional migration actions. See the [z/OS Migration] book on how to migrate from various z/OS releases. Also, see the [z/OS Summary of Message and Interface Changes] book for interface changes.

Also, see [z/OS Integrated Security Services LDAP Server Administration and Use] for information about the Integrated Security Services LDAP server in z/OS V1R10 and prior releases.

The following documentation, which is supplied with your product order, provides information about installing your z/OS system. In addition to specific information about the LDAP server, this documentation contains information about all of the z/OS elements.

- [z/OS Planning for Installation]
  This information describes the installation requirements for z/OS at a system and element level. It includes hardware, software, and service requirements for both the driving and target systems. It also describes any coexistence considerations and actions.

- [z/OS Program Directory]
  This document, which is provided with your z/OS product order, leads you through the specific installation steps for the LDAP server and the other z/OS elements.

- [serverPac: Installing Your Order]
  This is the order-customized, installation book for using the serverPac Installation method. Be sure to review the /usr/lpp/ldap/etc/ds.conf file, data sets supplied, jobs or procedures that have been completed for you, and product status. IBM might have run jobs or made updates to PARMLIB or other system control data sets. These updates could affect your migration.

Within this topic you can find information about the specific updates and considerations that apply to this release of z/OS LDAP.

- [Migration roadmap]
  This section identifies the migration paths that are supported with the current level of the LDAP server. It also describes the additional publications that can assist you with your migration to the current level.

- [z/OS V1R11 overview]
  This section describes the specific updates that were made to LDAP for the current release. For each item, this section provides an overview of the change, a description of any migration and coexistence tasks that might be considered, and where you can find more detailed information in the z/OS LDAP library or other element libraries.

### Actions required for migrations from previous releases of z/OS IBM TDS

The following topics describe common activities and considerations that are typically required, or should be considered, when you migrate from a previous release of z/OS IBM TDS to the current release. After each heading in this section, a table indicates the z/OS IBM TDS releases you are migrating from, showing an X for each release the particular section pertains to.

#### Native authentication and SDBM operations

<table>
<thead>
<tr>
<th>z/OS IBM TDS V1R9</th>
<th>z/OS IBM TDS V1R10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>
When native authentication is configured for an LDBM or TDBM backend, the RACF user ID that is used during a native authentication bind is kept with the bind information. The bound client can then issue SDBM operations, performed under the context of that RACF user ID. The results of these operations depend on the RACF authorization of the RACF user ID. You cannot disable SDBM operations specifically for native authentication bound clients.

Review the RACF authorizations of the RACF user IDs used during native authentication to ensure they can only perform SDBM operations that are allowed.

### Migrating to changed format of SMF 83 audit records

<table>
<thead>
<tr>
<th>z/OS IBM TDS V1R9</th>
<th>z/OS IBM TDS V1R10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

The format of the SMF type 83 audit records created by the LDAP server is changed. Two fields, LDAP_MAPCERT_OPT (111) and LDAP_MAPPED_SAFID (112), have been added to the common relocates section and field LDAP_RESERVED_04 (110) has been shortened. These fields are described in [Appendix E, “SMF records.”](#) The sample programs (LDAPDBTB, LDAPDBLD, and GLDLDPSC) that are used to process the audit records are updated because of these changes.

If you have created a procedure to process the SMF records created by the LDAP server, update the procedure to take account of the changed format of these records.

### Actions required for migrations from z/OS ISS LDAP server

- Starting in z/OS V1R11, the Integrated Security Services (ISS) LDAP server is no longer available. You must migrate to the IBM Tivoli Directory Server (IBM TDS) component of z/OS. Any migration steps involving the ISS LDAP server must be performed on the prior level of z/OS where the ISS LDAP server is running. They cannot be performed on z/OS V1R11.

The following sections describe common activities and considerations that are typically required (or should be considered) if you decide to migrate data from the ISS LDAP server to z/OS IBM TDS.

**Note:** The migration procedures sometimes involve using the `tdbm2ldif` utility to unload schema or backend entries. A line in the output file from the `tdbm2ldif` utility can have trailing blanks that are part of an attribute value. If you need to edit the file, make sure to use an editor that does not remove trailing blanks on lines. See the description of the `tdbm2ldif` utility in [z/OS Integrated Security Services LDAP Server Administration and Use](#) for more information.

After each heading in this section, a table indicates the z/OS ISS LDAP server releases you could be migrating from, showing an X for each release the particular section that it pertains to. If, for example, you are migrating from z/OS V1R9, the sections that show an X under z/OS ISS V1R9 and V1R10 apply to your installation and should be considered.

#### Migrating server configuration and started task procedure

<table>
<thead>
<tr>
<th>z/OS ISS V1R9 and V1R10</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
</tr>
</tbody>
</table>

There are several changes between the configuration file, environment variables file, and started task procedure used by the ISS LDAP server and z/OS IBM TDS.
You can use the IBM TDS configuration utility, dsconfig, to create the new versions of these files and procedure. dsconfig is described in [dsconfig utility]. The input files for dsconfig are not the same as those that are used by the ISS LDAP server utility, ldapcnf. If you used ldapcnf in the past, use the values that you specified in its input files to create the dsconfig input files. If you did not use ldapcnf in the past, use the values you specified in your configuration file to create the dsconfig input files. Make sure to specify the same suffixes as in your current server and, for TDBM, the same database owner. After running dsconfig, the only output members that you need to use are DSCONFIG (the configuration file), DSENVVAR (the environment variables file), and the procedure member (the server started task procedure). Do not submit any of the JCL jobs produced by dsconfig because your system should already be set up to run an LDAP server. Also, do not run the TDBM or GDBM SPUFI script output from dsconfig. This destroys the contents currently in the TDBM or GDBM directory. If you are using TDBM, then follow the instructions in [Migrating TDBM from an ISS LDAP server to TDBM on z/OS IBM TDS].

If you are manually creating the configuration files and started task procedure, use the information in the rest of this section.

The default name of the IBM TDS configuration file is ds.conf. slapd.conf remains the default name for the ISS LDAP server configuration file. The following explains some of the changes that need to be made to allow an ISS LDAP server configuration file to be used with z/OS IBM TDS. See the descriptions in [Configuration file options] for more information about the new and changed options.

1. Copy the ISS LDAP server configuration file to /etc/ldap/ds.conf and edit your copy.
2. Replace the DLL names in all occurrences of the database option. The DLL names are specified in the dblibpath parameter of the database option. Replace GLDBTDBM with GLDBTD31, GLDBSDBM with GLDBSD31/GLDBSD64, GLDBGDBM with GLDBGD31/GLDBGD64, and GLDXPDIR with GLDXPD31/GLDXPD64. For example, change
database tdbm GLDBTDBM
to
database tdbm GLDBTD31
3. If you are using sysplex support, replace the sysplexServerName and sysplexGroupName options with the serverSysplexGroup option.
4. Move the schemaReplaceByValue and dnCacheSize options, if specified, from the TDBM backend section to the global section.
5. Remove the databaseName option, if specified.
6. If the dsnaoini option is specified in a TDBM or GDBM backend, specify it in every TDBM and GDBM backend and specify the same value. Do the same for the serverName option.
7. If GDBM is configured, the changeLoggingParticipant option in the GDBM backend controls whether change log entries are created for changes to the LDAP server schema. Set changeLoggingParticipant off if you do not want change log entries for schema changes.

There are several changes between the environment variables files used by the ISS LDAP server and z/OS IBM TDS. The default name of the z/OS IBM TDS environment variables file is ds.envvars. slapd.envvars remains the default name for the ISS LDAP server environment variables file. The following explains some of the changes that need to be made to allow an ISS LDAP server environment variables file to be used with z/OS IBM TDS. See the environment variables descriptions in [Chapter 9, "Running the LDAP server"] for more information about the new and changed environment variables.

1. Copy the /usr/lpp/ldap/etc/ds.envvars file to /etc/ldap and edit your copy.
2. Compare the default environment variables file to your ISS LDAP server environment variables file and make any appropriate changes.

The following explains some of the changes that need to be made to create the z/OS IBM TDS started task procedure:

1. A sample z/OS IBM TDS started task procedure can be found in GLDHLQ.SGLDSAMP(DSSRV). Copy this procedure to your procedure library.
2. Update the started task procedure, adding the same options that you specified when starting the ISS LDAP server. You must also indicate if the server is to run in 31-bit or 64-bit mode. Make sure to specify the new configuration and environment variables files where appropriate.

3. Make sure you have another copy of the ISS LDAP server procedure (it might be necessary later to restart the ISS LDAP server). Then rename DSSRV to the name of the ISS LDAP server procedure. Using the same procedure name will maintain any automation that you have set up for the LDAP server.

Migrating schema from an ISS LDAP server to z/OS IBM TDS

<table>
<thead>
<tr>
<th>z/OS ISS V1R9 and V1R10</th>
<th>X</th>
</tr>
</thead>
</table>

z/OS IBM TDS has a single global schema, named `cn=schema`, for all the backends. The ISS LDAP server has a separate schema in each TDBM backend. This schema is named `cn=schema,suffix`, where `suffix` is any suffix in the TDBM backend. The TDBM schema in the ISS LDAP server must be added to the z/OS IBM TDS global schema.

If you plan to run the TDBM backend in z/OS IBM TDS, it automatically merges the TDBM backend schema into the z/OS IBM TDS global schema the first time that z/OS IBM TDS is started with the TDBM backend configured. In this case, you do not need to perform any additional schema migration.

If you are migrating a TDBM backend from an ISS LDAP server to an LDBM backend in z/OS IBM TDS, the schema must be manually migrated from the TDBM backend schema to the z/OS IBM TDS global schema. The process is explained in Migrating TDBM from an ISS LDAP server to LDBM on z/OS IBM TDS.

Note:

1. There are several combinations of Boolean syntax and matching rules that are supported in the TDBM backend schema in the ISS LDAP server but are not supported in the z/OS IBM TDS global schema. If the TDBM backend schema in the ISS LDAP server contains attributes that use these combinations, the attribute definitions must be modified before the schema can be added to z/OS IBM TDS. The combinations are Boolean syntax (1.3.6.1.4.1.1466.115.121.1.7) with any of the following:
   - `caseExactIA5Match` equality rule - change to `caseExactMatch`
   - `caseIgnoreIA5Match` equality rule - change to `caseIgnoreMatch`
   - `caseExactOrderingMatch` ordering rule - remove the ordering rule
   - `caseIgnoreOrderingMatch` ordering rule - remove the ordering rule
   - `caseExactSubstringsMatch` substring rule - remove the substring rule
   - `caseIgnoreSubstringsMatch` substring rule - remove the substring rule

2. The TDBM backend schema must include a definition for every attribute and object class in use in an entry in the TDBM directory. z/OS IBM TDS will not start if an attribute or object class definition is missing.

Migrating TDBM from an ISS LDAP server to TDBM on z/OS IBM TDS

<table>
<thead>
<tr>
<th>z/OS ISS V1R9 and V1R10</th>
<th>X</th>
</tr>
</thead>
</table>

After migrating the configuration file, you need to make several changes to the TDBM database to migrate it to the level used by z/OS IBM TDS. This involves adding a new column to the DIR_MISC table and adding 3 new replication tables. A TDBM migration SPUFI script is provided to make these changes.
Notes:

1. The changes will not affect usage of the TDBM backend in an ISS LDAP server, therefore, the TDBM database can be shared between z/OS IBM TDS and ISS.

2. z/OS IBM TDS requires that the size of the VALUE column of the DIR_SEARCH table and the size of the DN_TRUNC column of the DIR_ENTRY table both be at least 8. If the size of either of these columns is less than 8 in the TDBM database you are migrating, you must increase the size of the column. This requires unloading the TDBM directory, redefining the table, and then reloading the TDBM directory.

Follow these steps:

1. Copy the TDBMMGRT member of the GLDHLQ.SGLDSAMP dataset to your own SPUFI input dataset. Edit your version of TDBMMGRT. Read the commentary to understand what this SPUFI script is going to do. There are several changes you must make and several you need to consider:
   a. You must replace -DB2_NAME-, -DB2_USERID-, -MISC_TABLESPACE-, -REPLICA_TABLESPACE-, and -STORAGEGROUP- with the appropriate values for the TDBM database you are migrating.
   b. If you are using replication in the ISS LDAP server, make sure that the DIR_CHANGE replication table is empty. The SPUFI script explains how to do this.
   c. If you do not plan to share the TDBM database with an ISS LDAP server, uncomment the SQL statement that changes the DB_VERSION to '4.0'. This will allow the TDBM backend to fully use all the new function in z/OS IBM TDS, including basic replication and caching support in a sysplex. '4.0' is the same version level as is set for a new TDBM database created using z/OS IBM TDS.

2. Stop the ISS LDAP server.

3. Use the DB2 SPUFI facility to run your version of TDBMMGRT. The script must be run under a user ID with DB2 SYSADM authority. When the script completes running, scan the output to ensure that it ran successfully.

4. Start z/OS IBM TDS. The server tries to merge the schema in the TDBM backend into the LDAP server global schema. The merge fails if the two schemas have different definitions for the same attribute or object class or if the TDBM backend schema uses a combination of syntax and matching rule that is not supported by the z/OS IBM TDS global schema. In that case you must:
   a. Unload the schema from the TDBM backend in the ISS LDAP server into LDIF format using the ISS unload utility. For example:
      ```
      tdbm2ldif -o tdbmSchema.ldif -s cn=schema,suffix -f /etc/ldap/slapd.conf
      ```
      where suffix is any one of the suffixes in the TDBM backend.
   b. Edit tdbmSchema.ldif, using an editor that does not remove trailing blanks, and fix (or remove) the schema definitions that do not match or that are not supported in the z/OS IBM TDS global schema.
   c. If you changed the DB_VERSION to '4.0', you must change it back to '3.0' before starting the ISS LDAP server. To do this, perform the following SQL operation through SPUFI:
      ```
      UPDATE USERID.DIR_MISC SET DB_VERSION='3.0';
      ```
      where USERID is the value of the dbuserid option in the TDBM backend section of slapd.conf for the ISS LDAP server.
   d. Start the ISS LDAP server. Make sure that schemaReplaceByValue off is not specified in the slapd.conf file.
   e. Update the TDBM backend schema in the ISS LDAP server using the ldapmodify command. For example:
      ```
      ldapmodify -D adminDN -w adminPW -f tdbmSchema.ldif
      ```
   f. Stop the ISS LDAP server.
g. Start z/OS IBM TDS. z/OS IBM TDS merges the TDBM backend schema into the LDAP server global schema.

h. If you changed the DB_VERSION to ‘3.0’, you should change it back to ‘4.0’ before starting the ISS LDAP server. To do this, stop z/OS IBM TDS and perform the following SQL operation through SPUFI:

```
UPDATE USERID.DIR_MISC SET DB_VERSION='4.0';
```

where USERID is the value of the dbuserid option in the TDBM backend section of slapd.conf for the ISS LDAP server.

5. Save a backup copy of the z/OS IBM TDS global schema using the z/OS IBM TDS unload utility. For example:

```
ds2ldif31 -o /home/u123/schemaBackup.ldif -s cn=schema -f /etc/ldap/ds.conf
```

### Migrating TDBM from an ISS LDAP server to LDBM on z/OS IBM TDS

#### z/OS ISS V1R9 and V1R10

**X**

**Note:** The LDBM backend is optional. You can continue to use TDBM by following the instructions in [Migrating TDBM from an ISS LDAP server to TDBM on z/OS IBM TDS](#). Also, LDBM is meant to be used for small to medium sized directories. See [Setting up for LDBM](#) for more information about LDBM storage usage and size considerations.

Moving directory entries from TDBM to LDBM requires an unload of the TDBM directory to LDIF and then a load of the LDIF file into LDBM.

1. Stop z/OS IBM TDS, if it is running.

2. Configure z/OS IBM TDS for LDBM, using the same suffixes as used for the TDBM backend in the ISS LDAP server configuration file. Configure multiple LDBM sections if you are migrating multiple TDBM backends. Do not include those TDBM backends in the configuration.

3. Start z/OS IBM TDS in maintenance mode. Maintenance mode restricts server updates to just several users, including the LDAP administrator.

   ```
s dssrv -m
   ```

4. Stop the ISS LDAP server.

5. Migrate the schema, using the following:
   a. Unload the schema from the TDBM backend in the ISS LDAP server into LDIF format using the ISS unload utility. For example:

   ```
tdbm2ldif -o tdbmSchema.ldif -s cn=schema,suffix -f /etc/ldap/slapd.conf
   ```

   where suffix is any one of the suffixes in the TDBM backend.

   b. Update the z/OS IBM TDS global schema using the `ldapmodify` command. For example:

   ```
   ldapmodify -D adminDN -w adminPW -f tdbmSchema.ldif
   ```

   **Note:** If the schema modification fails because of conflicting schema definitions or because the TDBM backend schema uses a combination of syntax and matching rule that is not supported by the ISS LDAP server global schema, you must resolve any problems in the unloaded schema and then reissue the `ldapmodify` command. When editing the unloaded schema file, make sure to use an editor that does not remove trailing blanks.

   c. Repeat steps a and b for the schema in any additional TDBM backends that need to be migrated.

   d. Save a backup copy of the z/OS IBM TDS global schema using the z/OS IBM TDS unload utility. For example:

   ```
ds2ldif31 -o /home/u123/schemaBackup.ldif -s cn=schema -f /etc/ldap/ds.conf
   ```
6. Unload the directory entries from the TDBM backend in the ISS LDAP server into LDIF format using the ISS unload utility. If there is only one TDBM backend in the slapd.conf file, use an unload command such as:
   
   tdbm2ldif -o tdbmEntries.ldif -f /etc/ldap/slapd.conf

   If there is more than one TDBM backend, make sure that the slapd.conf file contains a name parameter on the database option for the TDBM backend to be migrated and use an unload command such as:
   
   tdbm2ldif -o tdbmEntries.ldif -n name -f /etc/ldap/slapd.conf

7. Load the directory entries into the LDBM backend in z/OS IBM TDS, using the ldapadd command. For example:
   
   ldapadd -D adminDN -w adminPW -f tdbmEntries.ldif

8. Verify that the TDBM backend has been successfully migrated to LDBM. You can use the ldapsearch command to display LDBM entries or the z/OS IBM TDS unload utility (ds2ldif) to create a backup of the LDBM database. At this point, the TDBM DB2 database is no longer needed.

9. Repeat steps 6 through 8 for each TDBM backend that is being migrated.

10. Turn off maintenance mode using the LDAP server MAINTMODE operator modify command:
    
    f dssrv,maintmode off

---

**Basic replication considerations when migrating TDBM from an ISS LDAP server to z/OS IBM TDS**

<table>
<thead>
<tr>
<th>z/OS ISS V1R9 and V1R10</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
</tr>
</tbody>
</table>

There is no migration of basic replication data between an ISS LDAP server and an IBM TDS. All replication activity on the ISS LDAP server must be completed before migration. Otherwise, replication updates to one or more replicas are lost and it will be necessary to synchronize these replicas with the master server. See [Recovering from basic replication out-of-sync conditions](#) for more information.

In the ISS LDAP server, replication is performed for all TDBM backends even if only one of the TDBM backends contains a replica object defining a replica of the LDAP server. In the IBM TDS, basic replication is performed only for the TDBM and LDBM backends that contain a replica definition. Therefore, when migrating from an ISS LDAP server which is using replication and has multiple TDBM backends, you must ensure each z/OS IBM TDS TDBM and LDBM backend has the appropriate replica objects. The following procedure can be performed on z/OS IBM TDS after the TDBM backends have been migrated.

1. Start z/OS IBM TDS in maintenance mode. Maintenance mode restricts server updates to just several users, including the LDAP administrator.
   
   s dssrv -m

2. Determine what replica objects are currently defined, using the ldapsearch command to search for entries that have an object class of replicaObject. The results are in LDIF format. For example:
   
   ldapsearch -D adminDN -w adminPW -L -s sub -b "" objectclass=replicaObject > myReplicas.ldif

3. Determine if additional replica objects need to be defined in the server by reviewing the replica objects contained in myReplicas.ldif. If a TDBM or LDBM backend does not contain a replica object and basic replication is required for that backend, you will need to add replica objects to that backend.

4. Edit myReplicas.ldif. Update the DN's of the replica objects to be within the naming context of the TDBM or LDBM backend requiring the replica object. Duplicate replica objects if they need to be added to more than one backend and remove replica objects that do not need to be added to any backend.

5. Use the ldapadd command to add the required replica objects to the TDBM and LDBM backends. For example:
lf
ldapadd -D adminDN -w adminPW -f myReplicas.ldif

6. Verify that all required replica objects have been added, using an `ldapsearch` command. For example:

```
ldapsearch -D adminDN -w adminPW -s sub -b "" objectclass=replicaObject
```

7. Turn off maintenance mode using the LDAP server MAINTMODE operator modify command:

```
f dssrv,maintmode off
```

---

**Migrating GDBM (change log) from an ISS LDAP server to DB2-based GDBM on z/OS IBM TDS**

| z/OS ISS V1R9 and V1R10 | X |

The DB2-based GDBM on z/OS IBM TDS can use the entries in a GDBM backend generated on an ISS LDAP server without any migration of data. Use the information in `Migrating server configuration and started task procedure` to determine if you need to change the GDBM backend section of the z/OS IBM TDS configuration file.

**Note:** z/OS IBM TDS requires that the size of the VALUE column of the DIR_SEARCH table and the size of the DN_TRUNC column of the DIR_ENTRY table both be at least 8. If the size of either of these columns is less than 8 in the GDBM database you are migrating, then you cannot migrate the GDBM database. Instead, ensure that all the entries in the GDBM database have been processed and then redefine the table.

---

**Migrating GDBM (change log) from an ISS LDAP server to file-based GDBM on z/OS IBM TDS**

| z/OS ISS V1R9 and V1R10 | X |

1. The file-based GDBM on z/OS IBM TDS cannot use the entries in a GDBM backend generated on an ISS LDAP server. Before migrating to a file-based GDBM, you must ensure that all entries in the change log have been processed. At this point, the GDBM DB2 database is no longer needed.

2. Remove all of the DB2-based specific options in the GDBM backend section of z/OS IBM TDS configuration file, including the `dbuserid` option. Add any file-based options that are appropriate, including `databaseDirectory` if you do not want to use the default file directory. See `Configuration file checklist` for more information. Also see `Migrating server configuration and started task procedure` for other considerations concerning the GDBM configuration options.

---

**Changing DB_VERSION setting for TDBM on z/OS IBM TDS**

| z/OS ISS V1R9 and V1R10 | X |

If you decide to no longer share a TDBM backend between an ISS LDAP server and z/OS IBM TDS, you can change the TDBM database level to ‘4.0’ to allow the TDBM backend to fully utilize the sysplex and replication support of z/OS IBM TDS.

**Note:** The TDBM backend can no longer be shared with an ISS LDAP server after this change is made.

To determine the current TDBM database level, use SPUFI to run the following SQL:

```
SELECT DB_VERSION FROM USERID.DIR_MISC;
```
where USERID is the value of the \textit{dbuserid} option in TDBM backend section in the z/OS IBM TDS configuration file.

To change the level, use SPUFI to run the following SQL:

\texttt{UPDATE USERID.DIR_MISC SET DB_VERSION='4.0';}

\section*{Schema updates}

<table>
<thead>
<tr>
<th>z/OS ISS V1R9 and V1R10</th>
<th>X</th>
</tr>
</thead>
</table>

The \texttt{schema.user.ldif} and \texttt{schema.IBM.ldif} schema files shipped with z/OS IBM TDS have been updated. If you are using these files, you need to update the z/OS IBM TDS global schema. See \texttt{Upgrading schema} for LDBM, TDBM, and CDBM for more information. All future schema service will assume that the latest version of these files have been applied to customer’s schema.

\section*{Migration considerations for applications}

<table>
<thead>
<tr>
<th>z/OS ISS V1R9 and V1R10</th>
<th>X</th>
</tr>
</thead>
</table>

There are many differences between z/OS IBM TDS and the ISS LDAP server that can affect LDAP applications that run with z/OS LDAP:

1. Message numbers, message text, reason code numbers, and reason code text have changed. These are not an application programming interface, but some applications might be parsing these values to determine operation results.

   One specific area of change are the return codes and reason codes returned during bind. The following tables indicate the reason code changes for various bind and password change operations.

   \textbf{TDBM bind results under native authentication:}

<table>
<thead>
<tr>
<th>z/OS ISS LDAP server</th>
<th>z/OS IBM TDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>R004108 TDBM backend _passwd API resulted in an internal error.</td>
<td>R004108 Native user ID ‘name’ is either not defined or no UID is present in the OMVS segment</td>
</tr>
</tbody>
</table>

   \textbf{TDBM modify password under native authentication:}

<table>
<thead>
<tr>
<th>z/OS ISS LDAP server</th>
<th>z/OS IBM TDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>R004108 TDBM backend _passwd API resulted in an internal error.</td>
<td>R004108 Native user ID ‘name’ is either not defined or no UID is present in the OMVS segment</td>
</tr>
</tbody>
</table>

2. The output from a \texttt{cn=monitor} search or the LDAP server DISPLAY MONITOR operator modify command in z/OS IBM TDS is mostly compatible with the output from the ISS LDAP server. There are some differences:

   a. The DN cache statistics are returned as part of the server statistics in z/OS IBM TDS because there is a single DN cache for the entire LDAP server. In the ISS LDAP server, DN cache statistics are reported separately for each TDBM and GDBM backend.

   b. z/OS IBM TDS returns statistics for each LDBM backend that is configured.

   c. The monitor statistics returned for a DB2-based GDBM backend are similar in the two servers. However, the monitor statistics returned for a file-based GDBM backend in z/OS IBM TDS are different and more resemble the LDBM monitor statistics.
3. In z/OS TDS, the SDBM backend uses the RACF **R_admin** profile extract interface to retrieve a RACF user, group, or connection profile. In the ISS LDAP server, SDBM uses the RACF **R_admin** run-command interface to issue RACF **LISTUSER** and **LISTGRP** commands to obtain the profiles. There are many differences in the output that RACF returns, which results in differences in the output that LDAP returns when searching user, group, and connection entries.

a. When a user or group profile keyword has no value, the RACF **LISTUSER** and **LISTGRP** commands substitute in a fake default value (for example, 'NONE') to indicate this, while the **R_admin** profile extract interface just does not return the keyword. Therefore, the ISS LDAP server output contains the corresponding LDAP attribute along with its pseudo default value, but z/OS IBM TDS output does not contain the attribute.

The entry returned for a RACF user by the SDBM backend in z/OS IBM TDS does not contain the following pseudo default values:

- racfprogrammername=UNKNOWN
- racfpasswordchangedate=00.000
- racfattributes=NONE
- racfrevokedate=NONE
- racfresumedate=NONE
- racflastaccess=UNKNOWN
- racfclassname=NONE
- racfinstallationdata=NO-INSTALLATION-DATA
- racfdatasetmodel=NO-MODEL-NAME
- racflogontime=ANYTIME
- racfsecuritylevel=NONE SPECIFIED
- racfsecuritycategorylist=NONE SPECIFIED
- racfsecuritylabel=NONE SPECIFIED
- racfoperatoridentification=
- racfterminaltimeout=
- racfdceuuid=NONE
- racfdchomecelluuid=NONE
- racfprimarylanguage=NOT SPECIFIED
- racfsecondarylanguage=NOT SPECIFIED
- racflnotesshortname=NONE
- racfnndusername=NONE
- racfomvsuid=NONE
- racfomvmsmaximumcputime=NONE
- racfomvsmmaximumaddressspacesize=NONE
- racfomvmsmaximumfilesperprocess=NONE
- racfomvmsmaximumprocessesperuid=NONE
- racfomvmsmaximumthreadsperprocess=NONE
- racfomvsmmaximummemorymaparea=NONE
- racfovmuid=NONE
- racflldapbindpw=NO

The entry returned for a RACF group by the SDBM backend in z/OS IBM TDS does not contain the following pseudo default values:

- racfinstallationdata=NO INSTALLATION DATA
- racfdatasetmodel=NO MODEL DATA SET
- racfsubgroupname=NO SUBGROUPS
- racfomvsgroupid=NONE
- racfomvsgroupid=NONE

The entry returned for a RACF connection by the SDBM backend in z/OS IBM TDS does not contain the following pseudo default values:

- racfconnectlastconnect=UNKNOWN
- racfconnectattributes=NONE
- racfconnectrevokedate=NONE
- racfconnectresumedate=NONE

b. Attributes with numeric values do not have leading 0’s. For example:

<table>
<thead>
<tr>
<th>ISS LDAP server</th>
<th>z/OS IBM TDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>racfomvsgroupid=0000012345</td>
<td>racfomvsgroupid=12345</td>
</tr>
</tbody>
</table>
c. A keyword with multiple values is returned by SDBM in z/OS IBM TDS as a multi-valued attribute. The ISS LDAP server sometimes returns this as a single-valued attribute in which the value is a list. For example:

<table>
<thead>
<tr>
<th>ISS LDAP server</th>
<th>z/OS IBM TDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>racfcrypttype=DES DES3 DESD</td>
<td>racfcrypttype=DES</td>
</tr>
<tr>
<td></td>
<td>racfcrypttype=DES3</td>
</tr>
<tr>
<td></td>
<td>racfcrypttype=DESD</td>
</tr>
<tr>
<td>racfattributes= ADSP SPECIAL</td>
<td>racfattributes=ADSP</td>
</tr>
<tr>
<td></td>
<td>racfattributes=SPECIAL</td>
</tr>
</tbody>
</table>

d. The format is changed for attribute values that are dates.

For `racfauthorizationdate`, `racfconnectauthdate`, `racfconnectlastconnect`, `racflastaccess`, `racfpassphraselastchangedate`, and `racfpasswordchangedate`, the date format has changed from `yy.ddd` to `mm/dd/yy`. For example:

<table>
<thead>
<tr>
<th>ISS LDAP server</th>
<th>z/OS IBM TDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>racfauthorizationdate=04.287</td>
<td>racfauthorizationdate=10/13/04</td>
</tr>
</tbody>
</table>

For `racfconnectresumedate`, `racfconnectrevokedate`, `racfresumedate`, and `racfrevokedate`, the date format has changed from `month dd, yyyy` to `mm/dd/yy`.

<table>
<thead>
<tr>
<th>ISS LDAP server</th>
<th>z/OS IBM TDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>racfresumedate=DECEMBER 11, 2009</td>
<td>racfresumedate=12/11/09</td>
</tr>
</tbody>
</table>

e. The format is changed for some attribute values that are times. For example:

<table>
<thead>
<tr>
<th>ISS LDAP server</th>
<th>z/OS IBM TDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>racflogontime=09:00 - 15:00</td>
<td>racflogontime=0900:1500</td>
</tr>
<tr>
<td>racfterminaltimeout=99:00</td>
<td>racfterminaltimeout=9900</td>
</tr>
</tbody>
</table>

f. The format of range values for attributes is changed to display all the values in the range. For example:

<table>
<thead>
<tr>
<th>ISS LDAP server</th>
<th>z/OS IBM TDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>racfroutcodekeyword=1:4</td>
<td>racfroutcodekeyword=1</td>
</tr>
<tr>
<td></td>
<td>racfroutcodekeyword=2</td>
</tr>
<tr>
<td></td>
<td>racfroutcodekeyword=3</td>
</tr>
<tr>
<td></td>
<td>racfroutcodekeyword=4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ISS LDAP server</th>
<th>z/OS IBM TDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>racflogondays=ANYDAY</td>
<td>racflogondays=SUNDAY</td>
</tr>
<tr>
<td></td>
<td>racflogondays=MONDAY</td>
</tr>
<tr>
<td></td>
<td>racflogondays=TUESDAY</td>
</tr>
<tr>
<td></td>
<td>racflogondays=WEDNESDAY</td>
</tr>
<tr>
<td></td>
<td>racflogondays=THURSDAY</td>
</tr>
<tr>
<td></td>
<td>racflogondays=FRIDAY</td>
</tr>
<tr>
<td></td>
<td>racflogondays=SATURDAY</td>
</tr>
</tbody>
</table>

`racflogondays=WEEKDAYS` is similarly changed.

g. Some return codes and reason codes returned by SDBM are changed because SDBM in z/OS IBM TDS no longer issues the RACF `LISTUSER` and `LISTGRP` commands. For example, when using SDBM to retrieve a user entry that does not exist:
Coexistence with the ISS LDAP server

There are several ways in which the z/OS IBM TDS V1R11 can run with an Integrated Security Services (ISS) LDAP server on an earlier version of z/OS:

1. The servers can participate in basic master-replica replication or basic peer-to-peer replication. This configuration allows you to continue to use the ISS LDAP server for production while testing migration to z/OS IBM TDS. After z/OS IBM TDS is primed with data from the production server, replication can be set up to keep the directories in sync during the test cycle.

2. The servers can share TDBM and DB2-based GDBM directories in a sysplex. There are several restrictions when running in this manner, including that the TDBM backend cannot participate in basic peer-to-peer replication or as a master in basic master-replica replication. See Sharing a TDBM or DB2-based GDBM backend between z/OS IBM TDS and an ISS LDAP server for information about how to set up this environment and the restrictions.

Sharing a TDBM or DB2-based GDBM backend between z/OS IBM TDS and an ISS LDAP server

The following steps indicate how to share a TDBM or DB2-based GDBM backend between z/OS IBM TDS and an ISS LDAP server. Restrictions pertaining to sharing TDBM in this way are also discussed.

Note: As described in Migration considerations for applications, the same operation can return different results depending on whether an ISS LDAP server or z/OS IBM TDS processes the client request. An LDAP application running with a shared TDBM backend must be able to run with both servers.

1. Ensure that both servers are configured to share the backend.
   a. On the ISS LDAP server, specify multiserver on in the TDBM or GDBM backend section of the configuration file.
   b. On z/OS IBM TDS, specify the serverSysplexGroup option in the global section and multiserver on in the TDBM or GDBM backend section of the configuration file.
   c. Specify the same TDBM or GDBM backend section options (including dbuserid and pwEncryption) in both the ISS LDAP server and z/OS IBM TDS configuration files. You cannot specify the following option values in the TDBM backend section of the z/OS IBM TDS configuration file:
      - pwCryptCompat off
      - secretEncryption AES or DES
      - pwEncryption AES
      - pwEncryption DES unless the DES keys are stored in ICSF

2. Follow the instructions in Actions required for migrations from z/OS ISS LDAP server to migrate the TDBM or GDBM backend to z/OS IBM TDS. The GDBM backend must remain DB2-based on z/OS IBM TDS when sharing GDBM with an ISS LDAP server.
   When migrating a TDBM database, do not update the DB_VERSION value to '4.0'. TDBM cannot be run on an ISS LDAP server if the database is '4.0'.

3. There are several restrictions when sharing TDBM or DB2-based GDBM between an z/OS IBM TDS and an ISS LDAP server.
a. A schema update can only be performed on z/OS IBM TDS. A schema modify operation or `ldif2tdbm -s` command targeted to the TDBM backend on the ISS LDAP server is rejected. When the z/OS IBM TDS global schema is updated, z/OS IBM TDS merges the schema update into the TDBM backend schema to make the updated schema available to the ISS LDAP server.

b. A schema update that contains a non-numeric object identifier for an attribute or an object class is rejected by z/OS IBM TDS because non-numeric object identifiers are not supported by the TDBM backend schema in the ISS LDAP server.

Similarly, a schema update for an attribute that contains one of the following combinations of syntax and matching rules is rejected because these combinations are not supported by the TDBM backend schema in the ISS LDAP server. The rejected combinations and a suggested alternative are:

- Integer syntax (1.3.6.1.4.1.1466.115.121.1.27) and `integerFirstComponentMatch` equality rule: replace equality rule with `integerMatch`
- Object Identifier syntax (1.3.6.1.4.1.1466.115.121.1.38) and `objectIdentifierFirstComponentMatch` equality rule: replace equality rule with `objectIdentifierMatch`
- UTC Time syntax (1.3.6.1.4.1.1466.115.121.1.53) and `generalizedTimeMatch` equality rule: replace equality rule with `utcTimeMatch`

c. The enhanced sysplex support in z/OS IBM TDS is disabled for the TDBM or GDBM backend. In particular, the TDBM backend cannot be a basic replication read-write master or peer replica server. The TDBM backend can be a basic replication (read-only) replica. Also, the filter and dn-to-eid caches are disabled.

d. Features of z/OS IBM TDS that affect the format of data in the TDBM directory cannot be used because these features are not supported in the ISS LDAP server. This includes the AES and triple DES encryption algorithms, `secretKey`, `replicaCredentials`, `ibm-slapdMasterPW`, and `ibm-replicaKeyPwd` encryption, and the ASCII crypt algorithm. Data encrypted using these features on z/OS IBM TDS cannot be used by the ISS LDAP server. This can result in failures during bind and replication processing on the ISS LDAP server. The TDBM backend does not start on z/OS IBM TDS if any of these features are configured.

Notes:

1) Even if encryption is not configured for the `secretKey`, `replicaCredentials`, `ibm-slapdMasterPW`, and `ibm-replicaKeyPwd` attributes, z/OS IBM TDS adds the tag `{none}` before the attribute value when the value is added or modified by z/OS IBM TDS. This tag is removed when the attribute value is retrieved by z/OS IBM TDS, but the tag is not removed when the value is retrieved by the ISS LDAP server.

2) There is also another difference between the LDAP servers when creating a change log record for a change to the `secretKey`, `replicaCredentials`, `ibm-slapdMasterPW`, or `ibm-replicaKeyPwd` attributes. If the change is made on the ISS LDAP server, the change log record shows the attribute value in plain text. If the change is made on z/OS IBM TDS, the attribute value in the change log record is replaced by “ComeAndGetIt” for security purposes.

e. z/OS IBM TDS does not require that an equal sign (=) be escaped within an attribute value in a distinguished name (DN). It also does not require that a pound sign (#) be escaped unless it is the first character in an attribute value in a DN. The ISS LDAP server requires escaping for any equal sign or pound sign in an attribute value. For example, `cn=cn=ken+ou=\#myOrg#,c=US` is an acceptable DN in z/OS IBM TDS. If an entry with this DN is added to that server, it cannot be accessed by the IBM ISS LDAP server. To avoid this problem, ensure that all equal signs and pound signs are escaped to make the DN accessible to both LDAP servers. In the example above, the DN that should be used is `cn=cn\=ken+ou=\#myOrg\#,c=US`. This applies also when the distinguished name is an attribute value within an entry, such as for a `member` attribute.
You might find it necessary to return to using the ISS LDAP server (running on a level of z/OS that supports it) after migrating a TDBM backend to a TDBM or LDBM backend on z/OS IBM TDS. This is not difficult, but can be more complicated if features of the z/OS IBM TDS that are not supported in the ISS LDAP server have been used. The following discusses z/OS IBM TDS functions that can cause fallback problems.

Note: The fallback procedures sometimes involve using the ds2ldif utility to unload schema or backend entries. A line in the output file from the ds2ldif utility can have trailing blanks that are part of an attribute value. If you need to edit the file, make sure to use an editor that does not remove trailing blanks on lines. See the usage notes in ds2ldif utility for more information.

1. Encryption problems:
   a. Use of the secretEncryption option in the z/OS IBM TDS configuration file can result in encrypted values for the secretKey, replicaCredentials, ibm-slapdMasterPW, and ibm-replicaKeyPw attributes. If encryption is not configured, z/OS IBM TDS adds the tag (none) before the attribute value when the value is added or modified by z/OS IBM TDS. The ISS LDAP server does not support encryption or encryption tags on these attributes. Therefore, the ISS LDAP server processes these tagged attribute values and (possibly) encrypted values as the actual values. Specifically, replication will fail in the ISS LDAP server if the replicaCredentials value is created or modified by z/OS IBM TDS.
   b. A similar problem exists when specifying the new AES algorithm in the pwEncryption option in the z/OS IBM TDS configuration file because this algorithm is not supported by the ISS LDAP server. In the ISS LDAP server, these userPassword values will not be usable and binds using the userPassword values will fail.
   c. z/OS IBM TDS allows keys for DES encryption to be stored in files instead of ICSF. The ISS LDAP server requires usage of ICSF for DES encryption. In the ISS LDAP server, userPassword values encrypted in this way are not usable and binds using the userPassword values will fail.
   d. If the pwCryptCompat off and pwEncryption crypt options are specified in the z/OS IBM TDS configuration file, userPassword values are encrypted using an ASCII version of the crypt algorithm. The ISS LDAP server only supports an EBCDIC version of the crypt algorithm. In the ISS LDAP server, userPassword values encrypted in this way are not usable and binds using the userPassword values will fail.

For the secretEncryption, AES, and DES encryption features, unloading the entries containing these attributes from z/OS IBM TDS decrypts the values and the entries can then be loaded into the ISS LDAP server without a problem. If the entries are not unloaded, the values must be replaced in the ISS LDAP server after fallback processing is complete to make them usable. Crypt is a one-way encryption algorithm, therefore, unloading will not decrypt those values. In this case, the values must be replaced in the ISS LDAP server after fallback processing is complete.

2. Schema problems:
   a. Non-numeric object identifiers in the schema are supported by z/OS IBM TDS but not the ISS LDAP server. A TDBM backend will not start in an ISS LDAP server if the backend schema contains a non-numeric object identifier.
   b. Similarly, a TDBM backend will not start if the backend schema contains an attribute that uses one of the combinations of syntax and matching rule listed below. The combinations of syntax and matching rule, and a suggested alternative, are:
      - Integer syntax (1.3.6.1.4.1.1466.115.121.1.27) and integerFirstComponentMatch equality rule: replace equality rule with integerMatch
      - Object Identifier syntax (1.3.6.1.4.1.1466.115.121.1.38) and objectIdentifierFirstComponentMatch equality rule: replace equality rule with objectIdentifierMatch
      - UTC Time syntax (1.3.6.1.4.1.1466.115.121.1.53) and generalizedTimeMatch equality rule: replace equality rule with utcTimeMatch

168 z/OS V1R11.0 IBM Tivoli Directory Server Administration and Use for z/OS
If a new TDBM DB2 database is created during fallback processing:
- unload the z/OS IBM TDS global schema into an LDIF file
- edit the file (using an editor that does not remove trailing blanks)
- replace the non-numeric object identifiers with numeric object identifiers
- either remove attributes using the unsupported syntax and matching rule combinations (if the attributes are not used in entries) or change the matching rule
- load the schema into the TDBM backend

If the TDBM DB2 database is not recreated, the z/OS IBM TDS global schema needs to be modified before fallback processing can be done:
- Use a NOID change operation to replace non-numeric identifiers (see Updating a numeric object identifier (NOID) for more information).
- Either remove attributes using the unsupported syntax and matching rule combinations (if the attributes are not used in entries) or change the matching rule.

3. Basic replication considerations:
   a. Basic replication in a TDBM backend in the z/OS IBM TDS does not use the same tables as in the ISS LDAP server and the data in the replication table cannot be transferred between the servers. z/OS IBM TDS must process its basic replication data before the fallback procedure is performed. Otherwise, one or more replicas can be out of sync with this LDAP server.
   b. Each TDBM or LDBM backend in z/OS IBM TDS must have its own replica objects. In an ISS LDAP server, a replica object in one TDBM backend applies to all TDBM backends. When using the fallback procedure on more than one backend, make sure that there is not more than one replica object for a replica in the ISS LDAP server.

4. Distinguished name problems: If a distinguished name (DN) of an entry contains an unescaped equal sign (=) or an unescaped pound sign (#) in an attribute value, that entry cannot be accessed by the ISS LDAP server. Similarly, if the DN occurs in an attribute value within an entry, such as the member attribute, that attribute cannot be accessed by the ISS LDAP server. The DN needs to be fully escaped before fallback processing can be done. For the DN of an entry, rename the entry to the fully escaped DN. To do this, you must first rename the entry to some other DN entirely and then rename that entry to the fully escaped DN. For a DN as an attribute value within an entry, the entry has to be modified to delete the DN value and then add the fully escaped version of the DN.

**Fallback from a TDBM backend on z/OS IBM TDS to a TDBM backend on an ISS LDAP server**

This procedure involves unloading the schema from z/OS IBM TDS and loading it into the TDBM backend on the ISS LDAP server (running on a level of z/OS that supports it). Other TDBM database table changes also need to be made.

**Note**: The TDBM entries on z/OS IBM TDS are not unloaded during this procedure, therefore, are susceptible to the encryption problems described in Fallback procedures after migrating a TDBM backend to z/OS IBM TDS. Values encrypted using the enhanced encryption features described in that section must be replaced after fallback processing is complete. As an alternative to avoid this, follow the process described in Fallback from an LDBM backend in z/OS IBM TDS to a TDBM backend in an ISS LDAP server, which unloads the entries.

1. Stop z/OS IBM TDS, if it is running.
2. Configure the ISS LDAP server for TDBM, using the same suffix and dbuserid options as used for the TDBM backend in the z/OS IBM TDS configuration file. You must also specify the databaseName option.
   Remove the TDBM backend section from the z/OS IBM TDS configuration file.
3. Unload the z/OS IBM TDS global schema into LDIF format using the z/OS IBM TDS unload utility. For example:
   
   ```bash
ds2ldif31 -o /home/u123/itdsSchema.ldif -s cn=schema -f /etc/ldap/ds.conf [-j]
   ```
Edit the output file using an editor that does not remove trailing blanks and change the first line to:

cn=schema,,suffix

where suffix is any of the suffixes you have configured for the TDBM backend.

4. Several TDBM database table changes need to be made to allow the database to be used by the ISS LDAP server.

The DB_VERSION value must be reset to '3.0'. Also, the TIMESTAMP column value should be removed to prepare for future migration back to z/OS IBM TDS. To do this, perform the following SQL operations through SPUFI:

UPDATE USERID.DIR_MISC SET DB_VERSION='3.0';
UPDATE USERID.DIR_MISC SET SCHEMA_TIMESTAMP=NULL;

where USERID is the value of the dbuserid option in the TDBM backend section of ISS LDAP server configuration file.

5. If the replica tables were not dropped when the TDBM backend was migrated to z/OS IBM TDS, the tables need to be emptied. To do this, perform the following SQL operations through SPUFI:

DELETE * FROM USERID.DIR_REGISTER;
DELETE * FROM USERID.DIR_PROGRESS;
DELETE * FROM USERID.DIR_CHANGE;
DELETE * FROM USERID.DIR_LONGCHANGE;

If the replica tables were dropped when the TDBM database was migrated to z/OS IBM TDS, you must re-create the DIR_REGISTER, DIR_PROGRESS, DIR_CHANGE, and DIR_LONGCHANGE tables and the DIR_REGISTERX1, DIR_PROGRESSX1, DIR_CHANGEX1, and DIR_LONGCHANGEX1 indexes. Use the appropriate parts of the TDBMDB and TDBMINDX SPUFI files that you used to originally create these tables and indexes for the ISS LDAP server.

6. Stop the ISS LDAP server (if it is running) and then start the ISS LDAP server.

7. Load the schema into the TDBM backend in the ISS LDAP server, using the ldapmodify command. For example:

   ldapmodify -D adminDN -w adminPW -f /home/u123/itdsSchema.ldif

The ldapmodify command fails if the schema in the TDBM backend does not contain the aclentry attribute. In that case, edit the schema file using an editor that does not remove trailing blanks. Add the following two lines to the end of the schema file:

   -
   add: x

   and move the aclentry and aclpropagate lines that are near the end of the file to after the new lines at the end of the file. Do the same thing for the entryowner and ownerpropagate lines if the TDBM schema does not contain the entryowner attribute.

8. Verify that the TDBM backend in the z/OS IBM TDS has been successfully restored to the TDBM backend in the ISS LDAP server. You can use the ldapsearch command to display TDBM entries or the ISS LDAP server unload utility (tdbm2ldif) to create a backup of the TDBM database.

**Fallback from an LDBM backend in z/OS IBM TDS to a TDBM backend in an ISS LDAP server**

This procedure involves unloading the schema and the LDBM data from z/OS IBM TDS and loading it into the TDBM backend on the ISS LDAP server (running on a level of z/OS that supports it).

**Note:** The LDBM entries on z/OS IBM TDS are unloaded during this procedure, therefore, avoiding some (but not all) of the problems described in Fallback procedures after migrating a TDBM backend to z/OS IBM TDS.
This procedure can also be used for moving a TDBM backend back to the ISS LDAP server. This might be desirable when the backend contains encrypted attribute values, which will be decrypted when the entries are unloaded during this procedure.

1. Stop the ISS LDAP server and z/OS IBM TDS, if they are running.
2. Unload the z/OS IBM TDS global schema into LDIF format using the z/OS IBM TDS unload utility. For example:
   ```
   ds2ldif31 -o /home/u123/itdsSchema.ldif -s cn=schema -f /etc/ldap/ds.conf [-j]
   ```
   Edit the output file using an editor that does not remove trailing blanks and change the first line to:
   ```
   cn=schema,suffix
   ```
   where `suffix` is any of the suffixes you have configured for the LDBM backend.
   Also add the following two lines before the four lines at the end of the output file that contain the `aclentry`, `aclpropagate`, `entryowner`, and `ownerpropagate` attributes.
   ```
   add: x
   ```
3. Unload the directory entries from the LDBM backend in z/OS IBM TDS into LDIF format using the z/OS IBM TDS unload utility. If there are no other LDBM or TDBM backends in the `ds.conf` file, use an unload command such as:
   ```
   ds2ldif31 -o /home/u123/ldbmEntries.ldif -f /etc/ldap/ds.conf [-j]
   ```
   If there are other LDBM or TDBM backends, make sure that the `ds.conf` file contains a `name` parameter on the `database` option for the LDBM backend to be migrated and use an unload command such as:
   ```
   ds2ldif31 -o /home/u123/ldbmEntries.ldif -n name -f /etc/ldap/ds.conf [-j]
   ```
4. Create a TDBM database for the ISS LDAP server using the instructions in the z/OS Integrated Security Services LDAP Server Administration and Use. You must drop the existing database if you did not do this after migrating TDBM from the ISS LDAP server to z/OS IBM TDS.
5. Configure the ISS LDAP server for TDBM, using the same `suffix` options as used for the LDBM backend in the z/OS IBM TDS configuration file.
   Remove the LDBM backend section from the z/OS IBM TDS configuration file.
6. Load the schema and entries into the TDBM backend on the ISS LDAP server, using one of the following methods:
   a. Use the ISS LDAP server utility, ldif2tdbm, to load both the schema and the entries. See z/OS Integrated Security Services LDAP Server Administration and Use for more information about ldif2tdbm.
   
   or
   
b. Use the `ldapmodify` and `ldapadd` commands, as follows:
   1) Add the `masterServerDN` and `masterServerPW` options to the TDBM backend section of the ISS LDAP server configuration file. These are needed so that the backend can be updated in maintenance mode (this allows the `ibm-entryuuid` attribute to be included in the entries to be added).
   2) Start the ISS LDAP server in maintenance mode, using the `-m` option.
   3) Load the schema into the TDBM backend in the ISS LDAP server, using the `ldapmodify` command. For example:
      ```
      ldapmodify -D masterServerDN -w masterServerPW -f /home/u123/itdsSchema.ldif
      ```
   4) Load the LDBM directory entries into the TDBM backend in the ISS LDAP server, using the `ldapadd` command. For example:
      ```
      ldapadd -D masterServerDN -w masterServerPW -f /home/u123/ldbmEntries.ldif
      ```
5) Stop the ISS LDAP server and remove the `masterServerDN` and `masterServerPW` options from the LDAP server configuration file.

7. Start the ISS LDAP server.

8. Verify that the LDBM backend has been successfully restored to the TDBM backend in the ISS LDAP server. You can use the `ldapsearch` command to display TDBM entries or the ISS LDAP server unload utility (tdbm2ldif) to create a backup of the TDBM database. At this point, the LDBM database files are no longer needed.

### Fallback procedures after migrating a GDBM backend to z/OS IBM TDS

You might find it necessary to return to using the ISS LDAP (running on a level of z/OS that supports it) server after migrating a GDBM backend to a DB2-based or file-based GDBM backend on z/OS IBM TDS. The schema used by GDBM is always part of the schema used by z/OS IBM TDS and the ISS LDAP server, therefore, no schema migration processing needs to be done.

#### Fallback from a DB2-based GDBM backend in z/OS IBM TDS to a GDBM backend in an ISS LDAP server

This procedure is very simple because the GDBM entries can be used directly by the ISS LDAP server (running on a level of z/OS that supports it). No changes need to be made to the GDBM database tables because none were made during migration to z/OS IBM TDS. Use the following procedure:

1. Stop z/OS IBM TDS, if it is running.
2. Configure the ISS LDAP server for GDBM, using the same `dbuserid` and other options as used for the GDBM backend in the z/OS IBM TDS configuration file. See "Migrating server configuration and started task procedure" for considerations concerning GDBM configuration options.
3. Stop the ISS LDAP server (if it is running) and then start the ISS LDAP server.
4. Verify that the GDBM backend has been successfully restored in the ISS LDAP server, using the `ldapsearch` command to display GDBM entries.

#### Fallback from a file-based GDBM backend in an z/OS IBM TDS to a GDBM backend in an ISS LDAP server

The entries in a file-based GDBM backend in z/OS IBM TDS cannot be restored to the ISS LDAP server (running on a level of z/OS that supports it). The GDBM entries must be processed from z/OS IBM TDS before the fallback procedure is performed. After this is done, the fallback procedure is as follows:

1. Stop z/OS IBM TDS, if is running.
2. Create a GDBM database for the ISS LDAP server using the instructions in "z/OS Integrated Security Services LDAP Server Administration and Use". You must drop the existing database if you did not do this after migrating GDBM from the ISS LDAP server to z/OS IBM TDS.
3. Configure the ISS LDAP server for GDBM.
4. Stop the ISS LDAP server (if it is running) and then start the ISS LDAP server.

### Migration roadmap

This section describes the migration paths that are supported by the current release of the z/OS IBM TDS server.
**z/OS V1R11 update summary**

The following table summarizes the updates introduced to the z/OS IBM TDS in z/OS Version 1 Release 11 (z/OS V1R11). If you are migrating from releases before z/OS V1R11, you should review the information in the detailed section for each item.

<table>
<thead>
<tr>
<th>For information about:</th>
<th>See topic:</th>
</tr>
</thead>
<tbody>
<tr>
<td>RACF class and general resource support</td>
<td>Chapter 16, “Accessing RACF information”</td>
</tr>
<tr>
<td>GDBM change logging for RACF resources</td>
<td>Chapter 26, “Change logging”</td>
</tr>
<tr>
<td>Workload Manager (WLM)</td>
<td>Chapter 29, “Performance tuning”</td>
</tr>
<tr>
<td>Advanced replication</td>
<td>Chapter 24, “Advanced replication”</td>
</tr>
<tr>
<td>CDBM backend</td>
<td>Setting up for CDBM</td>
</tr>
<tr>
<td>Control values in entries unloaded by ds2ldif</td>
<td>ds2ldif utility</td>
</tr>
</tbody>
</table>

**z/OS V1R11 overview**

This section describes the new and changed IBM TDS functions introduced for z/OS Version 1 Release 11 (V1R11). The information about each item includes:

- Description
- Summary of the IBM TDS server tasks or interfaces that might be affected
- Coexistence considerations, if any, that are associated with the item
- Migration procedures, if any, that are associated with the item
- References to other publications that contain additional detailed information.

**RACF class and general resource support:**

*Description:* The SDBM backend is enhanced to display and manage RACF general resource profiles using SDBM add, compare, delete, modify, and search operations. In addition, the hierarchy of RACF classes and the resource profiles in each class can be displayed. RACF SETROPTS options that affect classes (such as refreshing RACTION) can also be performed using LDAP. New attributes and object classes are added to the LDAP schema to represent the fields in RACF classes, general resource profiles, and class options.

*What this change affects:* The LDAP schema is enhanced with new attributes and object classes. New SDBM entries representing classes, resource profiles, and RACF class options can be displayed and managed.

*Dependencies:* None.

*Coexistence considerations:* The additional schema is ignored in previous releases. In a sysplex containing mixed levels of LDAP server, SDBM search results are different if the new LDAP server is configured to process resources. Previous releases do not display these entries. Updates using the new entries fail in previous releases.

*Migration tasks:* To use SDBM to display and manage RACF classes, general resource profiles, and class options, specify **enableResources on** in the SDBM section of the LDAP server configuration file.

*For more information:* See Chapter 16, “Accessing RACF information” for more information.

**GDBM change logging for RACF resources:**

*Description:* LDAP is enhanced to create a GDBM change log entry to represent a change to a RACF general resource profile. The change log entry is similar to those created when RACF user, group, or connection profiles are changed.
What this change affects: GDBM can now contain additional entries.

Dependencies: RACF must be configured to request that LDAP create change log entries for changes to RACF general resource profiles.

Coexistence considerations: In a sysplex containing mixed levels of LDAP server, GDBM change log entries for changes to general resource profiles can be displayed from any level of servers. Changes to resource profiles on a system that is running a previous level of LDAP do not result in the creation of a change log entry.

Migration tasks: To create change log entries for changes to general resource profiles, specify enableResources on in the SDBM section of the LDAP server configuration file. Applications that read the change log, such as metadirectory products such as IBM Tivoli Directory Integrator, might need to be updated to also process the new change log entries.

For more information: See Chapter 26, “Change logging” for more information.

Workload Manager (WLM):

Description: The LDAP server is enhanced to use Workload Manager (WLM) which enables an installation to set performance goals for LDAP server operations.

What this change affects: The LDAP server will use WLM to dispatch incoming client requests to the appropriate WLM transaction name.

Dependencies: WLM should be configured to run with the LDAP server. At a minimum, a default service class should be set up otherwise all LDAP operations will run under a discretionary profile. A discretionary profile in WLM has a low priority when measured against other work in the LDAP server.

Coexistence considerations: None. In a sysplex containing mixed levels of LDAP server, the older levels of server continue to operate as before and are not affected by the new WLM support.

Migration tasks: Issue the following RACF commands before starting the LDAP server:

```
RDEFINE BPX.WLMSERVER CLASS(FACILITY) UACC(NONE)
PERMIT BPX.WLMSERVER CLASS(FACILITY) ID(userid) ACCESS(READ)
SETROPTS RACLIST(FACILITY) REFRESH
```

where userid is the user ID that runs the LDAP server.

If WLM is not to be used for classifying operations within the LDAP server, it is strongly recommended that a default service class be set up for the LDAP server in WLM. If a default service class is not set up in WLM, all LDAP server operations will run under the discretionary profile and receive a low priority.

For more information: See Chapter 29, “Performance tuning” for more information.

Advanced replication:

Description: The new advanced replication supports many functions that are not available in basic replication. These functions provide better synchronization of data throughout an enterprise.

What this change affects: Advanced replication changes the manner in which replication is done in IBM TDS.

Dependencies: The server compatibility level must be 5 or greater to enable advanced replication. See page 100 for information about the serverCompatLevel configuration option. Also, the CDBM backend must be configured.
**Coexistence considerations:** Advanced replication cannot be used in a sysplex containing mixed levels of LDAP server. In such a sysplex, the server compatibility level must be less than 5.

**Migration tasks:** The following must be done to use advanced replication:

1. Set the server compatibility level to 5 or greater. See page 100 for information about the `serverCompatLevel` configuration option.
2. Add the CDBM section to the server configuration file. See Setting up for CDBM for more information. Within the CDBM section, specify the `useAdvancedReplication` configuration option.
3. Disable basic replication because the LDAP server does not support both basic replication and advanced replication at the same time. To do this:
   a. Comment out any instances of the `masterServer`, `masterServerDN`, `masterServerPW`, `peerServerDN`, and `peerServerPW` configuration options in the server configuration file.
   b. Remove all basic replication entries with an objectclass of `replicaObject` from all configured TDBM and LDBM backends.
4. If adding advanced replication to a TDBM backend from a previous version of z/OS IBM TDS, the DB2 database must be updated to enable TDBM to be configured in an advanced replication environment. The changes are explained in section 3 of the TDBMMGRT member of the GLDHLQ.SGLDSAMP dataset. Follow these steps:
   a. Copy the TDBMMGRT member to your own SPUFI input dataset. Edit your version of TDBMMGRT. Read the commentary in section 3 to understand what this SPUFI script is going to do. You must replace `-DB2_NAME-`, `-DB2_USERID-`, `-MISC_TABLESPACE-`, and `-STORAGEGROUP-` with the appropriate values for the TDBM database you are migrating.
   b. Stop the LDAP server.
   c. Use the DB2 SPUFI facility to run your version of TDBMMGRT. The script must be run under a user ID with DB2 SYSADM authority. When the script completes running, scan the output to ensure that it ran successfully.
   d. Start the LDAP server.

*For more information:* See Chapter 24, “Advanced replication” for more information.

**CDBM backend:**

**Description:** A new backend, CDBM, stores some LDAP server configuration information, including the information used to configure advanced replication. The CDBM backend, like the LDBM backend, uses a z/OS UNIX System Services file system to store its entries.

**What this change affects:** LDAP server configuration information can be searched and modified while the LDAP server is running.

**Dependencies:** The server compatibility level must be 5 or greater for the CDBM backend to be configured. See page 100 for information about the `serverCompatLevel` configuration option.

**Coexistence considerations:** The CDBM backend cannot be used in a sysplex containing mixed levels of LDAP server. In such a sysplex, the server compatibility level must be less than 5.

**Migration tasks:** The following must be done to use the CDBM backend:

1. Set the server compatibility level to 5 or greater. See page 100 for information about the `serverCompatLevel` configuration option.
2. Add the CDBM section to the server configuration file. See Setting up for CDBM for more information.

*For more information:* See Setting up for CDBM for more information.

**Control values in ds2ldif LDIF output file:**
Description: The ds2ldif utility has been enhanced to unload by default the replicateOperationalAttributes control for each entry that is unloaded. See replicateOperationalAttributes for more information about the replicateOperationalAttributes control.

What this change affects: Each entry that is unloaded by ds2ldif has a control value. The replicateOperationalAttributes control contains the following operational attributes and their values, base64-encoded: creatorsName, createTimestamp, modifiersName, and modifyTimestamp.

Now an entry unloaded by ds2ldif looks like the following:

dn: cn=entry1,o=tdbm
control: 1.3.18.0.2.10.19 false:: MIGnMB8KAwGgQMY3JlYXVcnoOWYwI1MQoEGNoNWF
kbb1UuMDKAQAwQPY3JlYXRvI1tZ0YwIwMRgEfjIwMDgwNjE4MjIzNjA4Ljk3MDY3NVowI0
BAdAB1tXb2RzYI1cnNOYwI1MQoEGNoPWFkbb1uMDKAQAwQPb9kaZ5V1tZ0YwIwMRg
EFjIwMDgwNjE4MjIzNjA4Ljk3MDY3NVow=  
objectclass: person
objectclass: top
cn: entry1
sn: 1

Dependencies: The server compatibility level must be 5 or greater and the -j option must not be specified on the ds2ldif utility. The -j option prevents unloading replicateOperationalAttributes control values for each entry.

Coexistence considerations: If server compatibility level is less than 5, replicateOperationalAttributes control values are not unloaded for each entry. This allows the unloaded LDIF file to continue to be usable with the ldapadd, ldapmodify, and ldif2ds utilities on earlier releases.

Migration tasks: Set the server compatibility level to 5 or greater. See page 100 for information about the serverCompatLevel configuration option.

For more information: See ds2ldif utility for more information.

z/OS V1R10 update summary

The following table summarizes the updates introduced to the z/OS IBM TDS server in z/OS Version 1 Release 10 (z/OS V1R10). If you are migrating from releases prior to z/OS V1R10, you should review the information in the detailed section for each item.

Table 22. Summary of IBM TDS server updates for z/OS V1R10

<table>
<thead>
<tr>
<th>For information about:</th>
<th>See topic:</th>
</tr>
</thead>
<tbody>
<tr>
<td>RACF custom fields support</td>
<td>Chapter 16, “Accessing RACF information”</td>
</tr>
<tr>
<td>Password phrase support</td>
<td>Chapter 18, “Native authentication” and Chapter 16, “Accessing RACF information”</td>
</tr>
<tr>
<td>Certificate bind validation</td>
<td>Support of certificate bind</td>
</tr>
<tr>
<td>Support for tagged, encrypted passwords</td>
<td>Configuring for encryption</td>
</tr>
<tr>
<td>Plug-in extensions</td>
<td>IBM Tivoli Directory Server Plug-in Reference for Z/OS</td>
</tr>
<tr>
<td>Improved DB2 partition support</td>
<td>Creating the DB2 database and table spaces for TDBM or GDBM</td>
</tr>
<tr>
<td>Waiting for DB2 to start</td>
<td>Monitoring LDAP server resources</td>
</tr>
</tbody>
</table>

z/OS V1R10 overview

This section describes the new and changed IBM TDS server functions introduced for z/OS Version 1 Release 10 (V1R10). The information about each item includes:

- Description
RACF custom fields support:

Description: The LDAP server schema and SDBM backend are enhanced to display and manage RACF custom fields in user and group profiles. You can add an LDAP attribute to the LDAP server schema to represent a RACF custom field, using the new RACFFIELD value in an ibmAttributeTypes definition in the LDAP schema. This attribute is used by SDBM to return the custom field value or to set it.

What this change affects: The LDAP schema and SDBM can provide access to RACF custom fields in user and group profiles.

Dependencies: The RACF custom fields must first be defined in RACF. The appropriate RACF profile information must be specified when defining the attribute to represent the custom field in the LDAP server schema.

Coexistence considerations: The new RACFFIELD value in the ibmAttributesType definition is ignored in previous releases. In a sysplex containing mixed levels of LDAP server, SDBM search results are different if there are RACF custom user or group fields. Updates using custom fields fail in previous releases. Applications which use SDBM to set and retrieve RACF user or group profiles must be able to handle different output depending on the level of the LDAP server.

Migration tasks: To use SDBM to display and manage RACF custom fields in user and group profiles, update the LDAP server schema using the RACFFIELD value.

For more information: See Chapter 16, “Accessing RACF information” for more information.

Password phrase support:

Description: A RACF password phrase can be used instead of a RACF password in native authentication binds to LDBM or TDBM and in simple binds to SDBM. The password phrase can be changed during these binds using the currentValue/newValue syntax. The password phrase can also be changed in the special delete-add modify for an LDBM or TDBM native authentication entry.

In addition, RACF can create change log entries in GDBM to indicate that a password phrase has changed. If RACF is configured to envelope the password phrase, an SDBM search operation can be used to retrieve the password phrase envelope.

What this change affects: The longer, more complicated password phrases can be used during LDAP authentication to the security server. Metadirectory products can be notified of password phrase changes and can retrieve the password phrase envelope from RACF.

Dependencies: Password phrases must be assigned to RACF user IDs involved in native authentication and SDBM binds. RACF must be set up to support password phrase change logging and enveloping.

Coexistence considerations: Password phrases cannot be used in previous releases and RACF password phrase change logging and envelope retrieval are not supported. In a sysplex containing mixed levels of LDAP server, binds that use a password phrase fail on previous releases. SDBM searches to retrieve password phrase envelopes do not return the envelope. Applications which do LDAP binds to the security server must be aware of the level of server to which they are binding. Metadirectory functions that need to retrieve password phrase envelopes must direct their SDBM searches to a z/OS V1R10 IBM TDS LDAP server.
Migration tasks: LDAP applications that authenticate to the security server might need to be updated to support password phrases. If you are using a metadirectory product such as IBM Tivoli Directory Integrator to synchronize passwords with RACF, you may want to also add support for password phrases.

For more information: See Chapter 18, “Native authentication” and Chapter 16, “Accessing RACF information” for more information.

Certificate bind validation:

Description: The LDAP server can be configured to check that a certificate used during a SASL external bind is associated with a RACF user ID. If not, the LDAP server can reject the bind. The RACF user ID associated with the certificate is added to the bind information. The bound user can then perform SDBM operations under the context of this RACF user ID.

In addition, TDBM and LDBM native authentication bind is enhanced to support SDBM operations after the bind. The SDBM operations are performed under the context of the RACF user ID associated with the native authentication entry.

The SMF type 83 audit records created by the LDAP server are changed to include the security server user ID (if any) associated with the bound user.

What this change affects: SASL external binds can fail if the certificate is not associated with a RACF user. SDBM operations can be performed after SASL external binds and native authentication binds. Automated processing of the SMF audit records created by the LDAP server might fail because of the changes in the audit record format.

Dependencies: RACF utilities must be used to associate a certificate with a RACF user ID.

Coexistence considerations: SASL external binds and native authentication binds are not mapped to a RACF user in previous releases. In a sysplex containing mixed levels of LDAP server, SDBM operations fail if requested after these binds are performed on a previous release.

Migration tasks: Specify the sslMapCertificate option in the LDAP server configuration file to enable certificate mapping for SASL EXTERNAL binds. Also, update any automated processing of SMF audit records created by the LDAP server to account for the changes in the audit record format.

For more information: See Support of certificate bind for more information on mapping a certificate. See Chapter 18, “Native authentication” for more information on performing SDBM operations after native authentication bind. See Appendix E, “SMF records” for more information on the format of LDAP server SMF 83 audit records. Also, see Actions required for migrations from previous releases of z/OS IBM TDS for additional migration information.

Support for tagged, encrypted passwords:

Description: A tagged, encrypted userPassword value can be stored in an entry and then specified during a bind or compare operation. The LDAP server can be configured to return the tagged, encrypted value either in binary or in base-64 encoding. This permits better interoperability with values retrieved from non-z/OS LDAP servers.

What this change affects: Bind and compare operations using a tagged, encrypted password can now succeed. userPassword values can be returned by search in a format that can be used on non-z/OS LDAP servers.

Dependencies: None.
**Coexistence considerations:** Tagged, encrypted passwords cannot be used for bind or compare in a previous release. In a sysplex containing mixed levels of LDAP server, bind operations using these passwords fail on a previous release.

**Migration tasks:** Tagged, encrypted password values that were added to an entry in a previous release cannot be used for bind. The password values must be replaced using this release of the LDAP server to become usable. Add the `pwSearchOutput` option to the LDAP server configuration file to change the format in which tagged, encrypted password values are returned.

*For more information:* See [Configuring for encryption](#) for more information.

**Plug-in extensions to the LDAP server:**

**Description:** Plug-in extensions can be added to the LDAP server to add capabilities to the server. An extension can be called before or after an LDAP operation is processed or can be called to process an LDAP operation.

**What this change affects:** A plug-in extension can change the way the LDAP server processes an LDAP operation.

**Dependencies:** None.

**Coexistence considerations:** Plug-in extensions are not supported in previous releases. In a sysplex containing mixed levels of LDAP server, the results of operations will be different if the operations involve plug-in extensions.

**Migration tasks:** If plug-in extensions are needed, they must be created and then identified to the LDAP server using the `plugin` option in the LDAP server configuration file.

*For more information:* See [IBM Tivoli Directory Server Plug-in Reference for z/OS](#) for more information.

**Improved DB2 partition support:**

**Description:** The LDAP server has added a new algorithm for assigning the entry identification numbers that it uses to keep track of entries in a TDBM backend. The new algorithm improves the distribution of entries between partitions when using a partitioned DB2 database, resulting in quicker access to the entries.

**What this change affects:** The distribution of TDBM entries in a partitioned DB2 database is improved for faster access.

**Dependencies:** None.

**Coexistence considerations:** None.

**Migration tasks:** The DB2 database must be updated to enable TDBM to take advantage of the improved algorithm. The changes are explained in section 2 of the TDBMMGRT member of the `GLDHLQ.SGLDSAMP` dataaset. Follow these steps:

1. Copy the TDBMMGRT member to your own SPUFI input dataset. Edit your version of TDBMMGRT. Read the commentary in section 2 to understand what this SPUFI script is going to do. You must replace `-DB2_NAME-`, `-DB2_USERID-`, `-MISC_TABLESPACE-`, and `-STORAGEGROUP-` with the appropriate values for the TDBM database you are migrating.
2. Stop the LDAP server.
3. Use the DB2 SPUFI facility to run your version of TDBMMGRT. The script must be run under a user ID with DB2 SYSADM authority. When the script completes running, scan the output to ensure that it ran successfully.
4. Start the LDAP server.

If you do not make these changes, the LDAP server continues to use the same algorithm for assigning TDBM entry identification numbers as used in previous releases.

For more information: See Creating the DB2 database and table spaces for TDBM or GDBM for more information.

Waiting for DB2 to start:

Description: When the LDAP server contains a backend that uses DB2 (either TDBM or DB2-based GDBM), it attempts to connect to DB2 when the LDAP server is started. If DB2 is not active, the LDAP server can be configured to periodically retry to connect to DB2.

What this change affects: The LDAP server monitors the availability of DB2 during LDAP server initialization. This allows the LDAP server to be started before DB2 is active.

Dependencies: None.

Coexistence considerations: None.

Migration tasks: Specify the `db2StartUpRetryLimit` and `db2StartUpRetryInterval` options in the LDAP server configuration file to enable the LDAP server to wait for DB2 to start.

For more information: See Monitoring LDAP server resources for more information.
Chapter 11. Running and using the LDAP server utilities

Utility programs are provided to assist in initializing, backing up, and synchronizing the data managed by the LDAP server.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Server utility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encrypt passwords in a backend directory</td>
<td>db2pwden</td>
</tr>
<tr>
<td>Unload data from backend directory to an LDIF file</td>
<td>ds2ldif</td>
</tr>
<tr>
<td>Load data into backend directory</td>
<td>ldif2ds (TDBM only)</td>
</tr>
<tr>
<td>Identify differences between two directory subtrees on two different servers and optionally synchronize the servers</td>
<td>ldapdiff</td>
</tr>
<tr>
<td>Perform an extended operation to configure and manage enhanced replication environments</td>
<td>ldapexop</td>
</tr>
</tbody>
</table>

The db2pwden, ds2ldif, ldif2ds, and ldapexop utilities can be run in the z/OS shell, as jobs using JCL and procedures, or from TSO. The ldapdiff utility can only be run in the z/OS shell.

Format and usage information for the utilities are in:
- db2pwden utility
- ds2ldif utility
- ldif2ds utility
- ldapdiff utility
- ldapexop utility

See Creating the DB2 database and table spaces for TDBM or GDBM for information about the permissions necessary to run the ds2ldif and ldif2ds utilities with a TDBM backend.

Running the LDAP server utilities in the z/OS shell

In order to run the db2pwden, ds2ldif, ldif2ds, ldapdiff, or ldapexop utilities in the shell, some environment variables need to be set properly. Sample script files are shipped with the LDAP server for the ds2ldif, ldif2ds, and ldapdiff utilities in /usr/lpp/ldap/sbin. The ds2ldif31 script file is used to unload data from an LDBM, TDBM, CDBM, or schema backend, in a 31-bit server environment. The ds2ldif64 script file is used to unload data from an LDBM, CDBM, or schema backend in a 64-bit server environment. The ldif2ds script file is used to bulkload LDIF data into a TDBM backend. The ldapdiff script file is used to compare data between a master and replica server and to synchronize replica servers. These shell scripts need to be modified to fit your environment. The PATH variable should be set in the shell scripts. Ensure that /usr/sbin is added to the PATH environment variable. You should set STEPLIB to SYS1.SIEALNKE, if the PDS has not been placed in LNKLST. If you are loading or unloading entries in a DB2-based backend and your runtime libraries for DB2 are not in LNKLST or LPA on the system, you should also specify DB2HLQ.SDSNLOAD in the STEPLIB.

If you are using the ldif2ds utility to add an entry containing userPassword, secretKey, ibm-replicaKeyPwd, replicaCredentials, or ibm-slapdMasterPw attribute values that are encrypted using ICSF, the LIBPATH variable might need to be set. See Installing ICSF for encryption for more information.

When started, db2pwden, ds2ldif31, ds2ldif64, and ldif2ds read an environment variables file. The default file is /etc/ldap/ds.envvars. This default can be changed by setting the environment variable LDAP_DS_ENVVARS_FILE to the full path name of the environment variables file you want. Some of the environment variables that can be set are NLSPATH and LANG.
Running the LDAP server utilities from JCL

Sample JCL for running db2pwden, ds2ldif, ldif2ds, and ldapexop from batch is provided with the LDAP server. The JCL includes an inline procedure, which will need to be modified by each installation to ensure that the appropriate load modules can be found. It might also be necessary to modify the JOB card for installation-specific requirements. These jobs can be run by editing the JCL member of the GLDHLQ.SGLDSAMP dataset and entering the submit command. The member names are DB2PWDEN (for db2pwden), DS2LDF31 (for using ds2ldif in a 31-bit server environment), DS2LDF64 (for using ds2ldif in a 64-bit server environment), LDF2DS (for ldif2ds), and LDAPEXOP (for ldapexop).

If you are loading or unloading entries in a DB2-based backend and your runtime libraries for DB2 are not in LNKLST or LPA on the system, make sure you specify the DB2 high-level qualifier for your DB2 installation in a STEPLIB DD card in the LDF2DS or DS2LDF31 batch job. These utilities require the following DB2 dataset:

DB2HLQ.SDSNLOAD

Running the LDAP server utilities in TSO

The db2pwden, ds2ldif, ldif2ds, and ldapexop utilities can be run from TSO. Following are the steps to do this:

1. Make the PDS (GLDHLQ.SGLDEXEC) containing the CLISTs needed to run the utilities available in SYSEXEC:
   
   alloc f(SYSEXEC) da('GLDHLQ.SGLDEXEC')

2. You can change the default environment variables file for the utilities by creating a dataset to hold the environment variables and then using the TSO alloc command as shown:
   
   alloc f(ENVVAR) da('datasetname')

If you want to specify an LDAP server configuration file that is a data set you can specify the -f option on the command. For example:

   ds2ldf31 -f "//datasetname" -o /tmp/ldif.1

   Alternately, to specify the LDAP server configuration file by associating it with a DD name, enter in TSO:

   alloc da('datasetname') fi(config) shr

   and then invoke the utility without specifying the -f option.

When setup is complete, running these utilities follows the same syntax as used if running in the z/OS shell, except you must use ds2ldf31 instead of ds2ldif31, ds2ldif64 instead of ds2ldif64, ldif2ds instead of ldif2ds. There is no difference with db2pwden or ldapexop. See Running the LDAP server utilities in the z/OS shell.

SSL/TLS information for LDAP utilities

The contents of a client's key database file is managed with the gskkyman utility. See z/OS Cryptographic Services System SSL Programming for information about the gskkyman utility. The gskkyman utility is used to define the set of trusted certification authorities (CAs) that are to be trusted by the client. By obtaining certificates from trusted CAs, storing them in the key database file, and marking them as trusted, you can establish a trust relationship with LDAP servers that use certificates issued by one of the CAs that are marked as trusted.

If the LDAP servers accessed by the client use server authentication, it is sufficient to define one or more trusted root certificates in the key database file. With server authentication, the client can be assured that the target LDAP server has been issued a certificate by one of the trusted CAs. In addition, all LDAP
transactions that flow over the SSL/TLS connection with the server are encrypted, including the LDAP credentials that are supplied on the `ldap_sasl_bind_s()` API.

For example, if the LDAP server is using a high-assurance VeriSign certificate, you should obtain a CA certificate from VeriSign, receive it into your key database file, and mark it as trusted. If the LDAP server is using a self-signed gskkyman server certificate, the administrator of the LDAP server can supply you with a copy of the server’s certificate request file. Receive the certificate request file into your key database file and mark it as trusted.

Using the `db2pwden` or `ldapexop` utilities without the `-Z` parameter or the `ldapdiff` utility without the `-sZ` or the `-cZ` parameters and calling the secure port on an LDAP server (in other words, a nonsecure call to a secure port), is not supported. Also, a secure call to a nonsecure port is not supported.

SSL/TLS encrypts the key database file, therefore, either the key database password or a stash file must be specified on the `-P keyFilePW` parameter. If a stash file is specified, it must be specified in the form file:// followed immediately (no blanks in between) by the file specification of the stash file. See z/OS Cryptographic Services System SSL Programming for information about using the gskkyman utility to create a stash file.

### Using RACF key rings

Alternately, LDAP supports the use of a RACF key ring. See the certificate/key management section in z/OS Cryptographic Services System SSL Programming for instructions on how to migrate a key database to RACF and how to use the RACDCERT command to protect the certificate and key ring.

The user ID under which the LDAP client runs must be authorized by RACF to use RACF key rings. To authorize the LDAP client, you can use the RACF commands in the following example (where `userid` is the user ID running the LDAP client utility):

```plaintext
RDEFINE FACILITY IRR.DIGTCERT.LIST UACC(NONE)
RDEFINE FACILITY IRR.DIGTCERT.LISTRING UACC(NONE)
PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(userid) ACCESS(CONTROL)
PERMIT IRR.DIGTCERT.LIST CLASS(FACILITY) ID(userid) ACCESS(CONTROL)
```

Remember to refresh RACF after doing the authorizations.

```plaintext
SETROPTS RACLIST(FACILITY) REFRESH
```

After the RACF key ring is set up and authorized, specify the RACF key ring name for the `-K keyFile`, `-cK` `keystore`, or `-sK` `keystore` options and do not specify the `-P keyFilePw`, `-cP` `keystorePw`, or `-sP` `keystorePw` options.

### Using PKCS #11 tokens

The `db2pwden` and `ldapexop` utilities support the use of PKCS #11 tokens. The gskkyman utility or the RACDCERT command can be used to create or modify PKCS #11 tokens. ICSF uses the CRYPTOZ SAF class to determine if the issuer of the gskkyman utility or the RACDCERT command is permitted to perform the operation against a z/OS PKCS #11 token. See z/OS Cryptographic Services System SSL Programming for more information about the gskkyman utility and z/OS Security Server RACF Command Language Reference for more information about the RACDCERT command.

The user ID that runs the `db2pwden` or `ldapexop` utility must be authorized by RACF to use the PKCS #11 token. To authorize the user ID, you can use the RACF commands in the following example (where `NAME` is the name of the PKCS #11 token and `userid` is the user ID running the utility).

```plaintext
SETROPTS CLASSACT(CRYPTOZ)
RDEFINE CRYPTOZ USER.NAME UACC(NONE)
RDEFINE CRYPTOZ SO.NAME UACC(NONE)
PERMIT USER.NAME CLASS(CRYPTOZ) ID(userid) ACCESS(READ)
PERMIT SO.NAME CLASS(CRYPTOZ) ID(userid) ACCESS(READ)
```
Remember to refresh RACF after doing the authorizations.

```
SETROPTS RACLST(CRYPTOZ) REFRESH
```

Once the PKCS #11 token is set up and authorized, specify the PKCS #11 token for the `-K keyFile` option in the following manner: `-K *TOKEN*/NAME`. Also, do not specify the `-P keyFilePW` option when using a PKCS #11 token.

**Using a Java keystore or RACF key ring for ldapdiff**

Keys and certificates needed for using SSL with the `ldapdiff` utility must be stored in either a Java™ keystore file (.jks) or a RACF key ring. If your existing keys and certificates are in a System SSL key database file or a PKCS #11 token, you must migrate them to a keystore file or a RACF key ring. If they are already in a RACF key ring, you can use them as is or can migrate them to a keystore file. To migrate keys and certificates, you need to first export them from their source location and then import them into their target location.

You can use the `gskkyman` utility to export from a key database file or from a PKCS #11 token. You can use the `RACDCERT` command to export from a RACF key ring or PKCS #11 token.

You can then use the `RACDCERT` command to import into a RACF key ring or the `keytool` Java application to import into a Java keystore. Make sure to specify the `-storetype JCEKS` option when using `keytool`.

When setup is complete, identify the Java keystore or RACF key ring using the appropriate `-sK keystore, -sN keystoreType, -cK keystore, -cN keystoreType, -sT truststore, -st truststoreType, -cT truststore, or -ct truststoreType` options when running the `ldapdiff` utility.

See [z/OS Cryptographic Services System SSL Programming](#) for more information about the `gskkyman` utility and migrating keys and certificates. See [z/OS Security Server RACF Command Language](#) Reference for more information about the `RACDCERT` command. See the [keyTool User Guide for SDK](#) for more information about the `keytool` Java application.

---

**Utility programs**
**db2pwden utility**

**Purpose**
The db2pwden utility is provided to encrypt all unencrypted, AES encrypted, and DES encrypted user passwords in an already loaded TDBM, LDBM, or CDBM backend. The utility runs as a client operation while the server is active, and causes the server to encrypt all the userPassword attribute values that are unencrypted, AES encrypted, or DES encrypted with the pwEncryption method configured on the LDAP server. The utility must be run by the LDAP administrator or a user who has the authority to update passwords.

**Format**
db2pwden [options]

**Parameters**

The following table shows the options you can use for the db2pwden utility:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-?</td>
<td>Print this text.</td>
</tr>
<tr>
<td>-b baseDN</td>
<td>Use baseDN as the starting point for the update instead of the default. If -b is not specified, this utility examines the LDAP_BASEDN environment variable for a baseDN definition. If you are running in TSO, set the LDAP_BASEDN environment variable using Language Environment® runtime environment variable _CEE_ENVFILE. See z/OS XL C/C++ Programming Guide for more information. If you are running in the z/OS shell, export the LDAP_BASEDN environment variable.</td>
</tr>
<tr>
<td>-d debugLevel</td>
<td>Specify the level of debug messages to be created. The debugLevel is specified in the same fashion as the debug level for the LDAP server, as described on page 130 Table 20 lists the specific debug levels. The default is no debug messages.</td>
</tr>
<tr>
<td>-D bindDN</td>
<td>Use bindDN to bind to the LDAP directory. The bindDN parameter should be a string-represented DN. The default is a NULL string. If the -S or -m option is equal to DIGEST-MD5 or CRAM-MD5, this option is the authorization DN which is used for making access checks. This directive is optional when used in this manner.</td>
</tr>
<tr>
<td>-g realmName</td>
<td>Specify the realmName to use when doing a DIGEST-MD5 bind. This option is required when multiple realms are passed from an LDAP server to a client as part of a DIGEST-MD5 challenge; otherwise, it is optional.</td>
</tr>
<tr>
<td>-h ldapHost</td>
<td>Specify the hostname or IP address on which the LDAP server is running. The default is the local host.</td>
</tr>
<tr>
<td>-K keyFile</td>
<td>Specify the name of the System SSL key database file, RACF key ring, or PKCS #11 token. If this option is not specified, this utility looks for the presence of the SSL_key ring environment variable with an associated name. If keyFile is specified as <em>TOKEN</em>/NAME, then System SSL uses the specified PKCS #11 token. Otherwise, System SSL uses a key database file or a RACF key ring. In this case, System SSL first assumes that keyFile is a key database file name and tries to locate the file. If keyFile is not a fully-qualified Unix System Services file name, the current directory is assumed to contain the key database file. The name cannot be an MVS data set. If System SSL cannot locate the file, it then assumes that keyFile is a RACF key ring name. See SSL/TLS information for LDAP utilities for information about System SSL key databases, RACF key rings, and PKCS #11 tokens. This parameter is ignored if -Z is not specified.</td>
</tr>
</tbody>
</table>
### db2pwden

**Table 23. db2pwden options (continued)**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-m mechanism</code></td>
<td>Specify the bind method to use. You can use either <code>-m</code> or <code>-S</code> to indicate the bind method. Specify GSSAPI to indicate a Kerberos Version 5 bind is requested, EXTERNAL to indicate that a certificate (SASL external) bind is requested, CRAM-MD5 to indicate that a SASL Challenge Response Authentication Mechanism bind is requested, or DIGEST-MD5 to indicate a SASL digest bind is requested. The GSSAPI method requires a protocol level of 3 and the user must have a valid Kerberos Ticket Granting Ticket in their credentials cache by using the Kerberos <code>kinit</code> command line utility. The EXTERNAL method requires a protocol level of 3. You must also specify <code>-Z</code>, <code>-K</code>, and <code>-P</code> to use certificate bind. Unless you want to use the default certificate in the key database file, RACF key ring, or PKCS #11 token, use the <code>-N</code> option to specify the label of the certificate. The CRAM-MD5 method requires protocol level 3. The <code>-D</code> or <code>-U</code> option must be specified. The DIGEST-MD5 method requires protocol level 3. The <code>-U</code> option must be specified. The <code>-D</code> option can optionally be used to specify the authorization DN. If neither <code>-m</code> nor <code>-S</code> is specified, a simple bind is performed.</td>
</tr>
<tr>
<td><code>-S mechanism</code></td>
<td></td>
</tr>
<tr>
<td><code>-N keyFileDN</code></td>
<td>Specify the label associated with the certificate in the key database file, RACF key ring, or PKCS #11 token.</td>
</tr>
<tr>
<td><code>-Z</code></td>
<td>Use a secure connection to communicate with the LDAP server. Secure connections expect the communication to begin with the SSL/TLS handshake. The <code>-K keyFile</code> option or equivalent environment variable is required when the <code>-Z</code> option is specified. The <code>-P keyFilePW</code> option is required when the <code>-Z</code> option is specified and the key file specifies a file system key database file. Unless you want to use the default certificate in the key database file, RACF key ring, or PKCS #11 token, use the <code>-N</code> option to specify the label of the certificate.</td>
</tr>
<tr>
<td><code>-p ldapPort</code></td>
<td>Specify the TCP port where the LDAP server is listening. The default LDAP non-secure port is 389 and the default LDAP secure port is 636.</td>
</tr>
<tr>
<td><code>-P keyFilePW</code></td>
<td>Specify either the key database file password or the file specification for a System SSL password stash file. When the stash file is used, it must be in the form <code>file://</code> followed immediately (no blanks) by the file system file specification (for example, <code>file:///etc/ldap/sslstashfile</code>). The stash file must be a file and cannot be an MVS dataset.</td>
</tr>
<tr>
<td><code>-U userName</code></td>
<td>Specify the <code>userName</code> for CRAM-MD5 or DIGEST-MD5 binds. The <code>userName</code> is a short name (uid) that is used to perform bind authentication. This option is required if the <code>-S</code> or <code>-m</code> option is set to DIGEST-MD5.</td>
</tr>
<tr>
<td><code>-w passwd</code></td>
<td>Use <code>passwd</code> as the password for simple, CRAM-MD5, and DIGEST-MD5 authentication. The default is a NULL string.</td>
</tr>
</tbody>
</table>

All other command line inputs result in a syntax error message and the correct syntax is displayed. If the same option is specified multiple times or if both `-m` and `-S` are specified, the last value specified is used.

**Examples**

Following are examples using the `db2pwden` utility:

- The following command:
  ```
  db2pwden -D "cn=admin" -w secret
  ```
encrypts all unencrypted, AES encrypted, or DES encrypted passwords in the TDBM, LDBM, or CDBM backend at the LDAP server on the local host. The base is defined in the LDAP_BASEDN environment variable. The encryption method used is the pwEncryption method configured on the LDAP server.

- The following command:
  
  ```
  db2pwden -h ushost -p 391 -D "cn=admin" -w secret -b "o=university, c=US"
  ```

  encrypts all unencrypted, AES encrypted, or DES encrypted passwords starting at the base "o=university, c=US" in the TDBM or LDBM backend on host ushost at port 391. The encryption method used is the pwEncryption method configured on the LDAP server.

Diagnosis

- Exit status is 0 if no errors occur. Errors result in a non-zero exit status and a diagnostic message being written to standard error.
ds2ldif

ds2ldif utility

Purpose
The ds2ldif utility is used to unload entries from a directory stored in a TDBM, LDBM or CDBM backend into a file in LDAP Data Interchange Format (LDIF). The utility is also used to obtain the LDAP server schema entry. ds2ldif cannot be used to unload a GDBM or SDBM backend.

There are two versions of the ds2ldif utility:
- ds2ldif31 runs in a 31-bit environment and can be used to unload LDBM entries, TDBM entries, CDBM entries, or the LDAP server schema entry.
- ds2ldif64 runs in a 64-bit environment and can be used to unload LDBM entries, CDBM entries, or the LDAP server schema entry.

If using the ds2ldif utility in an interoperability environment, see Appendix F, “Guidelines for interoperability between non-z/OS TDS and z/OS TDS” for more information.

Format

```
ds2ldif31 | ds2ldif64 -o outputFile [-d debugLevel] [-f confFile] [-g]
          [-j] [-k keyFile] [-r [-q filterDN]]
          [[-s subtreeDN] [-n beName] [-l]] [-t]
          [-w adminPW] [-Z] [-?]
```

Parameters
The following table shows the options you can use for the ds2ldif utility:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-?</td>
<td>Display the usage.</td>
</tr>
<tr>
<td>-d debugLevel</td>
<td>Specify the level of the debug messages to be created. The debugLevel is specified in the same fashion as the debug level for the LDAP server, as described on page 130, Table 20 lists the specific debug levels. The default is no debug messages.</td>
</tr>
<tr>
<td>-f confFile</td>
<td>Specify the name of the LDAP server configuration file to use. This configuration file only needs to contain information for the backend to be unloaded. The configuration dataset specified by the CONFIG DD statement is used if the -f parameter is not specified. The /etc/ldap/ds.conf configuration file is used if the -f parameter is not specified and the CONFIG DD statement is not defined.</td>
</tr>
<tr>
<td>-g</td>
<td>Specify to unload entries in genealogical order. This unloads entries in each subtree together, doing a depth-first traversal of the directory. Specify this option when you are unloading a large number of entries that you will later load using the ldif2ds utility with the -g option, because this order of entries improves the capacity of ldif2ds to process large numbers of entries. Unloading in this order requires more processing and impacts unload performance.</td>
</tr>
<tr>
<td>-j</td>
<td>Indicate that the replicateOperationalAttributes control value is not written to the output LDIF file. The replicateOperationalAttributes control value has the modifyTimestamp, createTimestamp, creatorsName, and modifiersName attribute types and values for the entry base64 encoded.</td>
</tr>
<tr>
<td>-k keyFile</td>
<td>Specify the name of the file containing the LDAP server encryption keys. If running ds2ldif in batch, the dataset specified by the LDAPKEYS DD statement is used if the -k option is not specified. The key file must be specified if any entries unloaded have userPassword, secretKey, replicaCredentials, ibm-replicaKeyPwd, or ibm-slapdMasterPw attribute values that are encrypted using the DES or AES algorithm and not using ICSF. These attribute values are decrypted and base64 encoded as the entry is written to the output LDIF file.</td>
</tr>
<tr>
<td>-l</td>
<td>Specify that entries under the cn=localhost suffix are unloaded from the LDBM or TDBM backend. When an entire LDBM or TDBM backend is unloaded, the entries under cn=localhost are not unloaded unless the -l option is specified. The -l option cannot be used when the -s option is specified.</td>
</tr>
</tbody>
</table>
Specify the name of the LDBM, TDBM, or CDBM backend to unload. This is the name assigned to the backend on its database record in the LDAP server configuration file or the name that is automatically generated when the LDAP server is started. This can be used to indicate which LDBM, TDBM, or CDBM backend to process when there are multiple LDBM, TDBM, or CDBM backends in the configuration file. The \( -n \) option cannot be used when the \( -s \) option is specified.

Specify the fully qualified output file to contain the unloaded directory entries.

Specify a distinguished name (DN) of a replication filter entry which contains ibm-replicationFilterAttr attribute values. These values are filters used to skip entire entries or attributes within entries while unloading the directory. See [Partial replication](#) for more information about replication filter entries.

The targeted LDAP server must be running and the \( -r \) option must be specified when the \( -q \) option is specified. Also, the CDBM backend must be configured and useAdvancedReplication on specified in the CDBM backend section of the server configuration file to perform unload filtering.

Perform an unloadRequest extended operation (1.3.18.0.2.12.62) to unload the subtree or backend. If the LDAP server that contains the backend that is to be unloaded is running, an unloadRequest extended operation can be sent to the LDAP server to unload the entries.

Identify the DN of the top entry of the subtree whose entries are to be unloaded. This entry, plus all below it in the directory hierarchy, are written to the output file. The \( -s \) option must be used to unload the LDAP server schema entry, cn=schema. The \( -s \) option cannot be used when the \( -n \) option is specified.

Specify that encrypted userPassword attribute values are unloaded with their encryption tag in clear text. See page [191](#) for more information about this option.

When using the unloadRequest extended operation, specify the LDAP administrator password to bind with. Do not specify the \( -w \) option if the adminPW option is specified in the LDAP server configuration file. In this case, the value from the server configuration file is used to perform the LDAP bind before sending the unloadRequest extended operation to the LDAP server. If the adminPW configuration option is not present and the \( -w \) option is not specified, a prompt is displayed for the LDAP administrator password.

When using the unloadRequest extended operation, use SSL to encrypt the communication between the ds2ldif utility and the LDAP server. By default, the ds2ldif utility will attempt to use SSL to communicate with the LDAP server assuming that the LDAP server configuration file has the necessary SSL options (for example, sslKeyRingFile, sslKeyRingFilePW, sslCertificate, sslKeyRingStashFile) specified along with a secure listen option (for example, listen ldaps://). If SSL cannot be used, the ds2ldif utility fails.

All other command line inputs result in a syntax error message and the correct syntax is displayed. Also specifying the same option multiple times with different values will result in a syntax error.

### Examples

1. This example invokes the 31-bit version of ds2ldif to unload all of the entries from the TDBM backend named tdbml in the LDAP server configuration file /etc/ldap/ds.conf. The replicateOperationalAttributes control is not included in the unloaded entries. The two entries that reside in the TDBM backend named tdbml are written to file /tmpdata/ldif.data.

   ```
   ds2ldif31 -j -o /tmpdata/ldif.data -n tdbml -f /etc/ldap/ds.conf
   ```

   The /tmpdata/ldif.data output file contains the following:

   ```
   version: 1
   dn: o=tdbm
   objectclass: organization
   objectclass: top
   ```
ds2ldif

o: tdbm
 ibm-entryuuid: 3A67E000-2E5C-1876-99D0-402064040959
 aclentry: cn=Anybody:normal:rsc:system:rsc
 aclpropagate: TRUE
 entryowner: CN=ADMIN
 ownerpropagate: TRUE

dn: cn=entry1,o=tdbm
 objectclass: newPilotPerson
 objectclass: person
 objectclass: top
 cn: entry1
 sn: 1
 uid: entry1
 userpassword:: c2VjcmV0
 ibm-entryuuid: 0C3DE000-F85D-1884-94B3-402084027431

2. This example invokes the 64-bit version of ds2ldif to unload all of the data at and underneath the o=ldbm subtreeDN entry by performing an unloadRequest extended operation over SSL with an LDAP administrator password of secret. By default, the replicateOperationalAttributes control is included in the unloaded entries. The entries are written to file /ldbmdata/ldif.data on the system where the LDAP server is running.

ds2ldif64 -o /ldbmdata/ldif.data -s o=ldbm -r -w secret -Z -f /etc/ldap/ds.conf

The /ldbmdata/ldif.data output file contains the following unloaded entries:

version: 1

dn: o=ldbm
control: 1.3.18.0.2.10.19 false:: MIGnMB8KAQAwGgQMY3JlYXRvcmNOYW1lMQoECGNNuPWF
 kbWiUMDAMAQAwKwQPY3JlYXRvcmNOYW1lMQoECGNNuPWFkbWiUMDAMAQAwKwQPY3JlYXRvcmNOYW1lMQoECGNNuPWF
 kbnWiUMDAMAQAwKwQPY3JlYXRvcmNOYW1lMQoECGNNuPWF

o: ldbm
objectclass: organization
objectclass: top
ibm-entryuuid: 9A063000-FA92-1901-9FBA-402084027431
aclentry: cn=Anybody:normal:rsc:system:rsc
aclpropagate: TRUE
entryowner: CN=ADMIN
ownerpropagate: TRUE

dn: cn=replfilter,o=ldbm
control: 1.3.18.0.2.10.19 false:: MIGnMB8KAQAwGgQMY3JlYXRvcmNOYW1lMQoECGNNuPWF
 kbWiUMDAMAQAwKwQPY3JlYXRvcmNOYW1lMQoECGNNuPWFkbWiUMDAMAQAwKwQPY3JlYXRvcmNOYW1lMQoECGNNuPWF
 kbnWiUMDAMAQAwKwQPY3JlYXRvcmNOYW1lMQoECGNNuPWF

o: ldbm
objectclass: ibm-replicationFilter
objectclass: top
sn: replfilter
ibm-replicationfilterattr: (objectclass=person):(cn,sn)
ibm-entryuuid: 1014C000-B229-1970-817D-402084027431

2. This example invokes the 64-bit version of ds2ldif to unload all of the data at and underneath the o=ldbm subtreeDN entry by performing an unloadRequest extended operation over SSL with an LDAP administrator password of secret. By default, the replicateOperationalAttributes control is included in the unloaded entries. The entries are written to file /ldbmdata/ldif.data on the system where the LDAP server is running.

ds2ldif64 -o /ldbmdata/ldif.data -s o=ldbm -r -w secret -Z -f /etc/ldap/ds.conf

The /ldbmdata/ldif.data output file contains the following unloaded entries:

version: 1

dn: o=ldbm
control: 1.3.18.0.2.10.19 false:: MIGnMB8KAQAwGgQMY3JlYXRvcmNOYW1lMQoECGNNuPWF
 kbWiUMDAMAQAwKwQPY3JlYXRvcmNOYW1lMQoECGNNuPWFkbWiUMDAMAQAwKwQPY3JlYXRvcmNOYW1lMQoECGNNuPWF
 kbnWiUMDAMAQAwKwQPY3JlYXRvcmNOYW1lMQoECGNNuPWF

o: ldbm
objectclass: organization
objectclass: top
ibm-entryuuid: 9A063000-FA92-1901-9FBA-402084027431
aclentry: cn=Anybody:normal:rsc:system:rsc
aclpropagate: TRUE
entryowner: CN=ADMIN
ownerpropagate: TRUE

dn: cn=replfilter,o=ldbm
control: 1.3.18.0.2.10.19 false:: MIGnMB8KAQAwGgQMY3JlYXRvcmNOYW1lMQoECGNNuPWF
 kbWiUMDAMAQAwKwQPY3JlYXRvcmNOYW1lMQoECGNNuPWFkbWiUMDAMAQAwKwQPY3JlYXRvcmNOYW1lMQoECGNNuPWF
 kbnWiUMDAMAQAwKwQPY3JlYXRvcmNOYW1lMQoECGNNuPWF

o: ldbm
objectclass: ibm-replicationFilter
objectclass: top
sn: replfilter
ibm-replicationfilterattr: (objectclass=person):(cn,sn)
ibm-entryuuid: 1014C000-B229-1970-817D-402084027431

...
This example builds on the previous example, however, uses a replication filter entry to filter the unload data. The `cn=replfilter,o=ldbm` replication filter entry from 2 on page 190 has an `ibm-replicationFilterAttr` attribute value that only allows entries that have an `objectclass` value of `person` to be unloaded and only unloads the `cn` and `sn` attributes (if any) from these entries. See Partial replication for more information about replication filtering.

```
ds2ldif31 -o /ldbmdata/filtered.ldif.data -s o=ldbm -r -w secret -q cn=replfilter,o=ldbm -f /etc/ldap/ds.conf
```

The `/ldbmdata/filtered.ldif.data` output file contains the following unloaded entry:

```
version: 1
dn: cn=entry1,ou=rochester,o=ldbm
control: 1.3.18.0.2.10.19 false:: MIGnMB8BAwGgQMY3j1YXRvcmN0YW1wM1QoECEkgWFkbnGwK0QwQZ5VGTzX00Y1wMg
BADAbGh2RZmllcnN0YX11MQoECEkgPFkbnGwK0QwQZ5VGTzX00Y1wMg
EFjIwMDkwMTE2MTYwOTUxLjMxNjM3MVo=
objectclass: newPilotPerson
objectclass: person
objectclass: top
cn: entry1
sn: entry1sn
ibm-entryuuid: 4D3BE000-B14F-1970-817D-402084027431
```

Notes:

a. The `o=ldbm, cn=replfilter, o=ldbm` and `ou=rochester, o=ldbm` entries are not unloaded because they do not have an `objectclass` attribute value of `person`.

b. The `cn=entry1, ou=rochester, o=ibm` entry is allowed to be unloaded because it has an `objectclass` attribute value of `person`. However, the `userpassword` and `description` attribute values are filtered out because they are not specified in the replication filter.

c. The `objectclass` and `ibm-entryuuid` attribute values are always unloaded by the `ds2ldif` utility although they are not specified in the replication filter.

**Using the -t (tagging) option**

When the `-t` option is used on the `ds2ldif` utility, the format of the unloaded `userPassword` attribute depends on how the `userPassword` value is encrypted.

1. If the value is encrypted using a one-way encryption algorithm (crypt, SHA, or MD5), then the tag is visible and the encrypted `userPassword` value is base64 encoded in the unloaded value. The format of the unloaded value is:

   ```
   userpassword: {tag}base64encoded_and_encryptedvalue
   ```

   where, `tag` is `crypt`, MD5, or SHA. For example:

   ```
   userpassword: {crypt}0fCik9fUqZniuxKkYQ==
   userpassword: {SHA}24309gf[jgt
   userpassword: {MD5}34d1d21/hie8s
   ```

2. If the value is encrypted using a two-way encryption algorithm (DES or AES) or is not encrypted, then the unencrypted value is base64 encoded in the unloaded value and there is no tag. The format of the unloaded value is:
For example:
userpassword:: kfa6903axs
This is also the format used when unloading `secretKey`, `replicaCredentials`, `ibm-replicaKeyPwd`, and `ibm-slapdMasterPw` attribute values, because these values can only be encrypted using two-way encryption algorithms.

Note:
1. The LDAP server loads and uses tagged encrypted `userPassword` values that are encrypted in crypt, SHA, or MD5 and were unloaded using `ds2ldif` with the `-t` option. Also, these tagged encrypted `userPassword` values might be acceptable for other LDAP providers to load into their directory. If it is not directly loadable, this format is easily modifiable for loading by another provider into its LDAP directory.
2. The values returned by the `crypt` algorithm are portable only to other X/Open-conformant systems when the `pwCryptCompat` configuration option is set to `off`. When the `pwCryptCompat` option is set to `off`, the `crypt()` algorithm uses ASCII, which is a subset of UTF-8, when generating the encrypted `userPassword` attributes. Therefore, it is recommended that the `pwCryptCompat` option should be set to `off` when it is necessary to share `userPassword` attributes values encrypted in `crypt()` between the z/OS LDAP server and other ASCII-based LDAP servers. For more information about the `pwCryptCompat` option, see page 94.

When the `-t` option is not used on the `ds2ldif` utility, the format of the unloaded `userPassword` attribute depends upon how the `userPassword` value is encrypted.
1. If the value is encrypted using a one-way encryption algorithm (crypt, SHA, or MD5), then the tag and the encrypted `userPassword` value are base64 encoded in the unloaded value. The format of the unloaded value is:
   userpassword:: base64encodedValue
   where, `base64encodedValue` is a base64 encoding of
   `(tag)encryptedvalue`
   where, `tag` is crypt, MD5, or SHA. For example:
   userpassword:: e2NyeXB0fdHwopPX1KmZ4sbipGE=
2. If the value is encrypted using a two-way encryption algorithm (DES or AES) or is not encrypted, then the unencrypted value is base64 encoded in the unloaded value and there is no tag present. The format of the unloaded value is:
   userpassword:: base64encoded_and_unencryptedvalue
   For example:
   userpassword:: e01ENXkfa6903axs
   This is also the format used when unloading `secretKey`, `replicaCredentials`, `ibm-replicaKeyPwd`, and `ibm-slapdMasterPw` attribute values, because these values can only be encrypted using two-way encryption algorithms.

Using the `unloadRequest` extended operation

The `unloadRequest` extended operation is used to remotely unload directory data from a currently running z/OS LDAP server. The `unloadRequest` extended operation is required when attempting to unload data from an LDBM or CDBM backend that is already running with an active z/OS LDAP server. The `unloadRequest` extended operation is also required when using the `-q filterDN` option to filter entries as they are being unloaded from an LDBM, TDBM, or CDBM backend. The `-r` option can be used to force...
The ds2ldif utility does the following when an unloadRequest extended operation is performed:

1. An LDAP administrator simple bind is attempted using each secure listen option in the LDAP server configuration file until a successful secure connection is established with the LDAP server. The sslKeyRingFile option in the LDAP server configuration file indicates the key database file, RACF key ring, or PKCS #11 token that ds2ldif uses to communicate securely with the LDAP server. If the sslKeyRingFile option is a key database file, the sslKeyRingFilePW or sslKeyRingPWStashFile options in the LDAP server configuration file are used by ds2ldif to gain access to the key database file. The sslCertificate option in the LDAP server configuration file is used as the SSL certificate when ds2ldif establishes the secure connection with the LDAP server. The adminDN that is specified in the LDAP server configuration file is used as the bind DN. If there is an adminPW configuration option present, it is used as the password. If there is no adminPW configuration option, ds2ldif uses the value of the ds2ldif –w option or issues a prompt for the password if the –w option is not specified.

Notes:

a. If you are running ds2ldif from TSO or batch, the password prompt will not work. In these environments you must either have an adminPW configuration option or specify the -w option.

b. The getpass() routine used to prompt for the password returns at most PASS_MAX number of characters, truncating any additional characters. See the description of getpass() in z/OS XL C/C++ Run-Time Library Reference for more information. If the length of the administrator password is greater than PASS_MAX, you must either have an adminPW configuration option or specify the -w option.

c. Ensure that the userid running the ds2ldif utility has access to the key database file, RACF key ring, or PKCS #11 token that is specified on the sslKeyRingFile option.

2. If a secure connection is not established with the LDAP server or there are no secure listen options in the LDAP server configuration file, an LDAP administrator simple bind is attempted using each nonsecure listen option until a successful nonsecure connection is established. If a connection is not established with the LDAP server, ds2ldif ends.

Notes:

a. In order to perform the unloadRequest extended operation, it is required that the bound user be the LDAP administrator.

b. If the –Z option is specified, the ds2ldif utility communicates over a secure port if a connection is established. If a secure connection is not established, the ds2ldif utility fails.

3. Entries are unloaded into an output file on the LDAP server’s system. The name of the file is specified in the ds2ldif -o option.

4. An unloadResponse extended operation is sent back to the ds2ldif utility indicating how many entries are unloaded and whether the extended operation is successful.

Additional setup when unloading LDBM or CDBM directory data

If the ds2ldif utility is being used to unload LDBM or CDBM directory data and an unloadRequest extended operation is not being performed (the LDAP server is not running), the user ID that is running ds2ldif must have read and execute access to the directory that is specified on the databaseDirectory server configuration option of the LDBM or CDBM backend that is being unloaded. Also, the user ID must have read access to the LDBM or CDBM database and checkpoint files in that directory in order to successfully perform an unload.
ds2ldif

Usage
1. If the ds2ldif utility is invoked with neither the -s nor the -n option and there is only one TDBM,
   LDBM, or CDBM backend in the LDAP server configuration file, all of the directory entries (other than
   cn=localhost unless the -l option is specified) in that particular TDBM, LDBM, or CDBM backend are
   unloaded.

2. If a TDBM backend contains entries that have a suffix that is no longer configured in the TDBM
   section of the LDAP server configuration file, ds2ldif unloads those entries when it is invoked with the
   -n option. This is not true for an LDBM backend: only the entries that have a currently configured
   LDBM suffix are unloaded.

3. The ds2ldif utility only unloads owner and ACL information for entries that have a specific owner or
   ACL. Any entry data with an inherited owner or ACL will not have owner or ACL information unloaded.

4. The ibm-entryuuid attribute is included in each entry unloaded by the ds2ldif utility. The ldapadd
   command can be used to load an entry containing the ibm-entryuuid attribute only if the LDAP
   server is running in maintenance mode and the ldapadd is performed by the LDAP administrator. The
   ldif2ds utility supports loading entries containing the ibm-entryuuid attribute into a TDBM backend. If
   you do not need the ibm-entryuuid attribute in the loaded entry to have the same value as in the
   entry that was unloaded, then remove the attribute from each entry in the ds2ldif output file and a
   new ibm-entryuuid value will be assigned when the entry is loaded using ldapadd or ldif2ds.

5. The replicateOperationalAttributes control is added to each unloaded entry when the -j option is
   not specified and the server compatibility level is 5 or greater. The replicateOperationalAttributes
   control value has the modifyTimestamp, createTimestamp, creatorsName, and modifiersName
   attribute types and values for the entry, base64 encoded. Both the ldapadd command and the
   ldif2ds utility support the specification of the replicateOperationalAttributes control in the input
   LDIF file. See the replicateOperationalAttributes control in replicateOperationalAttributes for more
   information about the control. See the serverCompatLevel configuration option on page 100 for more
   information about the server compatibility level.

6. The LDAP Version 3 protocol has a related set of Internet Drafts which documents the introduction of
   a version mechanism for use in creating LDIF files. The ds2ldif utility always creates "tagged" LDIF
   files. The LDIF tag consists of a single line at the top of the LDIF file:

   version: 1

   All characters contained in the version one LDIF file are portable characters represented in the local
   code page. Strings containing nonportable characters (for example, textual values containing
   multi-byte UTF-8 characters) are base64 encoded.

7. To unload the LDAP server schema entry, specify:
   -s cn=schema

   The schema entry is unloaded in LDIF modify format. The current LDAP server schema entry is
   unloaded. No merging of TDBM schema into the LDAP server schema is performed when running
   ds2ldif.

8. If you want to unload the schema entry from a TDBM backend that is migrated from an Integrated
   Security Services LDAP server, specify the following:
   -s cn=schema,suffixDN

   where suffixDN is the DN of a suffix in the TDBM backend. The schema entry is unloaded in LDIF
   modify format. The current TDBM schema entry is unloaded. No merging of the LDAP server schema
   into the TDBM schema entry is performed when running ds2ldif.

9. When you edit the output file produced by ds2ldif, you should use an editor that does not delete
   blanks at the end of a line. If the output file contains a line that ends with blanks, using such an editor
   will result in deleting the blanks, therefore, changing the value of the attribute. This is very important
   when the line is continued, because the existing blanks no longer separates the last word in the line
   from the continuation on the next line. The maximum line length in a ds2ldif output file is 77;
continued lines are always 77 long when the file is created by ds2ldif. The oedit editor is an example of an editor that deletes blanks and should not be used.

10. The LDAP_DEBUG environment variable can also be used to set the debug level for ds2ldif. See page 130 for more information about specifying the debug level.

11. If running ds2ldif in 64 bit mode, ensure that the userID running the ds2ldif utility has a sufficient value specified for the MEMLIMIT and ASSIZEMAX values in the OMVS segment so that storage greater than 2GB can be used. This enables the ds2ldif utility to work properly in 64 bit mode.

12. If you are using ds2ldif to unload a large number of entries that you plan to later load using ldif2ds, specify the -g option on both ds2ldif and ldif2ds. This causes ds2ldif to unload the entries using a depth-first traversal of the directory: the root is unloaded, then each subtree directly below the root is traversed. This order results in improved ldif2ds load capacity, at the cost of some impact on ds2ldif performance. ldapadd performance does not benefit from the -g option for ds2ldif.

**Diagnosis**

Exit status is 0 if no errors occur. Errors result in a non-zero exit status and a diagnostic message being written to standard error.
ldif2ds

ldif2ds utility

Purpose
The ldif2ds utility is used to load entries specified in LDAP Data Interchange Format (LDIF) into a TDBM directory stored in a relational database. ldif2ds cannot be used to load entries into a GDBM, LDBM, CDBM, or SDBM directory. The TDBM directory must already exist. The ldif2ds utility is intended for loading a large number of entries. The utility creates load records from the entries in the LDIF input files, then runs the DB2 Load Utility to load the records into the TDBM directory.

The ldapadd command can also be used to add entries to a TDBM directory. See [When to use ldif2ds for more information about when to use ldif2ds or ldapadd. See IBM Tivoli Directory Server Client Programming for z/OS for more information about ldapadd.]

The ldif2ds utility may be used to add entries to an empty directory or to a directory that already contains entries. The utility cannot be used to modify the LDAP server schema. The schema must already be appropriate for the entries being added.

See Preparing to run ldif2ds before using ldif2ds.

Format

Parameters
The following table shows the options you can use for the ldif2ds utility:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-?</td>
<td>Display the usage.</td>
</tr>
<tr>
<td>-b creatorDN</td>
<td>The DN to be associated as creator with each loaded entry that does not already include a creatorsname attribute. If -b is not specified, the value of the adminDN option in the LDAP server configuration file is used. If neither option is specified, ldif2ds fails. The same processing is also applied for the modifier for each loaded entry that does not already include a modifiersname attribute. This option is ignored if ldif2ds is not invoked for the prepare step.</td>
</tr>
<tr>
<td>-c</td>
<td>Check that the entries in the LDIF input file or files are complete and acceptable according to the LDAP server schema. The DB2 database is not modified during the check phase.</td>
</tr>
<tr>
<td>-d debugLevel</td>
<td>Specify the level of debug messages to be created. The debugLevel is specified in the same fashion as the debug level for the LDAP server, as described on page 130. Table 20 lists the specific debug levels. The default is no debug messages.</td>
</tr>
<tr>
<td>-e ldifListFile</td>
<td>Specify the name of a file which contains a list of LDIF input files to be used as input. This is equivalent to using the -i option to specify a list of LDIF files, but is more convenient for a large list. Each record in the list file must contain the name of one LDIF input file. Blank lines and lines beginning with a # (comment lines) are ignored. See the -i explanation above for more information about using a list of LDIF files. This option is ignored if ldif2ds is not invoked for the check or prepare step.</td>
</tr>
</tbody>
</table>
### ldif2ds

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f confFile</code></td>
<td>Specify the name of the LDAP server configuration file to use. This configuration file only needs to contain information for the TDBM backend into which the entries are to be loaded. The configuration dataset specified by the CONFIG DD statement is used if the <code>-f</code> parameter is not specified. The <code>/etc/ldap/ds.conf</code> configuration file is used if the <code>-f</code> parameter is not specified and the CONFIG DD statement is not defined. This option is ignored if <code>ldif2ds</code> is not invoked for the check or prepare step.</td>
</tr>
<tr>
<td><code>-g</code></td>
<td>Specify that the entries to be loaded are in genealogical order in the input LDIF files. This is the order that results from doing a depth-first traversal of a directory. This order greatly reduces the storage usage of <code>ldif2ds</code> during the check and prepare steps, enabling more entries to be loaded at one time. Do not specify this option if the entries to load are not in genealogical order. This order is produced when entries are unloaded using <code>ds2ldif</code> with the <code>-g</code> option.</td>
</tr>
<tr>
<td><code>-l ldifFile</code></td>
<td>Specify the name of an LDIF file to use as input. If a list of file names is specified, each file in the list is processed in turn. If <code>ldif2ds</code> is invoked separately to run the check and prepare steps, make sure to specify the same set of LDIF files each time you run <code>ldif2ds</code>. See the description of the <code>ldapmodify</code> operation utility in the LDAP operation utilities chapter in IBM Tivoli Directory Server Client Programming for z/OS for more information about the general format of LDIF entries. This option is ignored if <code>ldif2ds</code> is not invoked for the check or prepare step.</td>
</tr>
<tr>
<td><code>-j</code></td>
<td>Specify that DB2 logging is used when loading the directory entries. The load operation is faster if DB2 logging is not used but the DB2 database is placed into copy-pending state after the data has been loaded. You then must run either the DB2 <code>COPY</code> or <code>REORG</code> utility in order to reset the copy-pending state. <strong>Note:</strong> This option can only be specified during the prepare step, when the JCL for the load jobs is created. It is ignored if <code>ldif2ds</code> is not invoked for the prepare step.</td>
</tr>
<tr>
<td><code>-k keyFile</code></td>
<td>Specify the name of the file containing the LDAP server encryption keys. If running <code>ldif2ds</code> in batch, the dataset specified by the <code>LDAPKEYS DD</code> statement is used if the <code>-k</code> option is not specified. The key file must be specified if any entries loaded have <code>userPassword</code>, <code>secretKey</code>, <code>ibm-replicaKeyPwd</code>, <code>ibm-slapdMasterPw</code>, or <code>replicaCredentials</code> attribute values that are encrypted using the DES or AES algorithm and not using ICSF. These attribute values are encrypted as the entry is prepared for loading into the directory.</td>
</tr>
<tr>
<td><code>-o outHlq</code></td>
<td>Invoke the DB2 Load Utility to load the TDBM database.</td>
</tr>
<tr>
<td><code>-p</code></td>
<td>Prepare DB2 table load files and JCL from the entries in the LDIF input file or files. The load and JCL files are generated as datasets, whose high-level qualifier is specified with the <code>-o</code> option. The prepare step deletes the existing contents of the datasets before writing new contents to the datasets. The DB2 database is modified during the prepare phase as entry identifiers and attribute identifiers are assigned for the new directory entries. The check phase is always performed when the <code>-p</code> option is specified, even if the <code>-c</code> option is not specified.</td>
</tr>
<tr>
<td><code>-o outHlq</code></td>
<td>Specify the high level qualifier of the datasets that will contain the DB2 load and JCL output, the status information, and the system information. These datasets must be allocated before invoking <code>ldif2ds</code>. The value cannot be more than 22 characters long. See Preparing to run <code>ldif2ds</code> for more information. This option is ignored if <code>ldif2ds</code> is not invoked for the prepare or load step.</td>
</tr>
</tbody>
</table>
Idif2ds

-q summaryFrequency

Specify the number of LDIF entries the check or prepare step should process between issuing summary messages. The default value is 1000, resulting in issuing a summary message after every 1000 entries are checked or prepared. If you have many LDIF entries, increase this value to reduce the frequency of summary messages. Specify a negative value or 0 to suppress issuing any intermediate summary messages. A final summary message is always issued. This option is ignored if idif2ds is not invoked for the check or prepare step.

All other command line inputs result in a syntax error message and the correct syntax is displayed. Also, specifying the same option multiple times with different values results in a syntax error message. Filenames are case-sensitive unless the filename refers to an MVS dataset or DD statement. An MVS dataset is indicated by //dsname while a DD statement is indicated by //DD:ddname.

Examples

Following is an example using the idif2ds utility:

```
ldif2ds -cpl -i /data2/ldif.data -o admin3.prv -d ERROR
```

This idif2ds utility invocation checks, prepares, and loads the LDIF data from /data2/ldif.data, uses the output datasets ADMIN3.PR.V.BULKLOAD.INPUT.xxx and ADMIN3.PR.V.BULKLOAD.JCL, and specifies a debug level of ERROR. By default, the server configuration file used is /etc/ldap/ds.conf, the value of the adminDN option in the LDAP server configuration file is used as the default creator of each loaded entry, no DB2 logging is done when loading the entries, and a progress message is issued after every 1000 entries processed during the check and prepare steps.

Preparing to run ldif2ds

Before invoking ldif2ds, you must

- Allocate the output datasets used by ldif2ds.
- Create the SYSTEM file used by the prepare step of ldif2ds.

Allocating datasets required by ldif2ds

The various ldif2ds processing steps use 6 load datasets and a JCL dataset. These datasets must be allocated before invoking ldif2ds and the high-level qualifiers in the dataset names must be specified as the value of the -o outHlq option on the command. The ldif2ds utility writes the contents of these datasets, except for the SYSTEM member of the JCL dataset which must be created before ldif2ds is run.

- Load record datasets
  The prepare step of ldif2ds writes the load data created from each LDIF entry into the load record datasets. Each dataset contains the records for one database table and is used as input to the DB2 Load Utility when loading that table.
  - Dataset names:
    ```
    outHlq.BULKLOAD.INPUT.DESC - DIR_DESC load dataset
    outHlq.BULKLOAD.INPUT.ENTRY - DIR_ENTRY load dataset
    outHlq.BULKLOAD.INPUT.LATTR - DIR_LONGATTR load dataset
    outHlq.BULKLOAD.INPUT.LENTRY - DIR_LONGENTRY load dataset
    outHlq.BULKLOAD.INPUT.REPLICA - DIR_REPLICA load dataset
    outHlq.BULKLOAD.INPUT.SEARCH - DIR_SEARCH load dataset
    ```
  - Dataset format:
    Sequential (non-PDS)
    Record format = VB
  - Dataset record length and block size:
    The record length depends on the page size of the corresponding table space. The actual record length may be reduced slightly by ldif2ds at runtime.
    For a 32K page size in DB2, use LRECL=32756, BLKSIZE=32760.

198 z/OS V1R11.0 IBM Tivoli Directory Server Administration and Use for z/OS
This record length and block size will also work for smaller page sizes. However, for smaller page sizes, a smaller LRECL and BLKSIZE can be used to reduce the required disk space. For example, on 3390 DASD, LRECL=27994, BLKSIZE=27998 enables 2 blocks for each track and, in general, more bytes for each track to be written. For the smaller page sizes, the LRECL and BLKSIZE must be at least pagesize + 6 and pagesize + 10, respectively.

- Dataset size:
  A rough estimate for the size (in bytes) of each dataset is as follows:
  
  `outHlq.BULKLOAD.INPUT.DESC`:
  
  \[ 20 \times (\text{average depth}) \times (\text{number of entries in the LDIF files}) \]
  
  where average depth = (average number of levels in a DN) - (average number of levels in each DN's suffix) + 1

  `outHlq.BULKLOAD.INPUT.ENTRY`, `outHlq.BULKLOAD.INPUT.LATTR`, and `outHlq.BULKLOAD.INPUT.ENTRY`:
  
  The combined space required for these files is roughly
  
  \[(\text{number of bytes in the LDIF input files}) \times 3.0\]
  
  If most directory entries are shorter than the DIR_ENTRY page size, and most attributes are shorter than the attrOverflowSize in the LDAP server configuration file, then most of the data is written in the `outHlq.BULKLOAD.INPUT.ENTRY` dataset. Otherwise, you may want to allocate each of these three datasets as this maximum size to ensure that you do not run out of space.

  `outHlq.BULKLOAD.INPUT.REPLICA`:
  
  \[(\text{number of replicas defined in the LDIF input files}) \times 24\]
  
  If there are no replica entries defined in the LDIF input files, use a minimum size to allocate the dataset, such as 1 block.

  `outHlq.BULKLOAD.INPUT.SEARCH`:
  
  \[(\text{number of bytes in the LDIF input files}) \times 2.5\]

- JCL dataset
  
  The JCL dataset is a PDS whose members contain system information, status information, and the JCL to run DB2 Load Utility for each database table that needs to be loaded. The prepare and load steps of ldif2ds use the status information. The prepare step writes the contents of the JCL members of the JCL dataset, using the information in the system member. The system member of the JCL dataset must be created by the user before ldif2ds is invoked to run the prepare step.

  - Dataset name:
    `outHlq.BULKLOAD.JCL`

  - Members:
    `outHlq.BULKLOAD.JCL(JDESC)` - DIR_DESC load JCL
    `outHlq.BULKLOAD.JCL(JENTR)` - DIR_ENTRY load JCL
    `outHlq.BULKLOAD.JCL(JLATT)` - DIR_LONGATTR load JCL
    `outHlq.BULKLOAD.JCL(JLENT)` - DIR_LONGENTRY load JCL
    `outHlq.BULKLOAD.JCL(JREPL)` - DIR_REPLICA load JCL
    `outHlq.BULKLOAD.JCL(JSRCH)` - DIR_SEARCH load JCL
    `outHlq.BULKLOAD.JCL(STATUS)` - Status information
    `outHlq.BULKLOAD.JCL(SYSTEM)` - System information - see below

  - Dataset format:
    Partitioned (PDS)
    Record format = FB
    Record length = 80

  - Dataset size:
    200K bytes

Creating the SYSTEM member
ldif2ds

The contents of the SYSTEM member, `outHlq.BULKLOAD.JCL(SYSTEM)`, must be created before invoking `ldif2ds` to run the prepare step, which uses the information in the SYSTEM member to create the JCL to invoke the DB2 Load Utility to load each of the database tables.

- Format of SYSTEM records
  
  \begin{verbatim}
  HLQ db2hlq  High level qualifier for DB2 datasets
  JOBCARD //jobname...  Job card record for JCL
  #Comment record  Ignored if first non-blank character is #
  \end{verbatim}

- The SYSTEM member must contain one or more JOBCARD records. The first JOBCARD record must contain the job name. The JOB statement may span multiple JOBCARD records and must follow the JCL continuation rules. The maximum length of each JCL record is 72 characters. JES control statements may also be included in the JOBCARD specification. For example:

  \begin{verbatim}
  JOBCARD //DB2TDBMn JOB (ACCOUNT),PGMR,CLASS=A,MSGCLASS=A,MSGLEVEL=1,
  JOBCARD // NOTIFY= &SYSUID
  JOBCARD /*JOBPARM SYSAFF=SYS1
  \end{verbatim}

  The HLQ record is optional. The DB2 high level qualifier defaults to DSN if the HLQ record is not specified. If specified, the maximum length of the HLQ value is 35.

  Make sure there are no sequence numbers at the end of each line of the SYSTEM member.

- To differentiate the load jobs in the 6 JCL members of the JCL PDS, `ldif2ds` replaces the last character of the job name with the digits 1 through 6, as follows:
  
  - Jobname ends in 1 - JCL for loading DIR_DESC table, in `outHlq.BULKLOAD.JCL(JDESC)`
  - Jobname ends in 2 - JCL for loading DIR_ENTRY table, in `outHlq.BULKLOAD.JCL(JENTRY)`
  - Jobname ends in 3 - JCL for loading DIR_LONGATTR table, in `outHlq.BULKLOAD.JCL(JLATT)`
  - Jobname ends in 4 - JCL for loading DIR_LONGENTRY table, in `outHlq.BULKLOAD.JCL(JLENT)`
  - Jobname ends in 5 - JCL for loading DIR_SEARCH table, in `outHlq.BULKLOAD.JCL(JSRCH)`
  - Jobname ends in 6 - JCL for loading DIR_REPLICA table, in `outHlq.BULKLOAD.JCL(JREPL)`

When to use ldif2ds

Both the `ldif2ds` utility and the `ldapadd` command can be used to load entries into a TDBM database. There are advantages and disadvantages to each of these. Following is a list of considerations that can assist in determining which method to use when loading entries.

\begin{table}
\centering
\begin{tabular}{|c|c|c|}
\hline
Considerations & ldif2ds & ldapadd \\
\hline
Speed of load & Faster & Slower, especially if the database is already large. \\
\hline
Operational attributes & Accepts input containing `creatorsname`, `modifiersname`, `createtimestamp`, `modifytimestamp`, and `ibm-entryuuid` operational attributes. & Input cannot contain these operational attributes. \\
\hline
Complexity & High, will take time to learn. Typical usage requires multiple invocations, with review of JCL and preparation for recovery before load. & Low. \\
\hline
Set up & User must allocate datasets and create the SYSTEM information file before running `ldif2ds` for the first time. & No set up. \\
\hline
LDAP server down time & LDAP server must be down during load step. Server can be up during check and prepare steps. & Server must be operational during adds. \\
\hline
DB2 logging & Logging is optional. Additional DB2 work is needed to make database fully usable after the load if logging is not performed. & Requires logging. \\
\hline
Recovery & Recovery from load failure is complex, involving knowledge of DB2 Utilities. & Simple recovery. \\
\hline
\end{tabular}
\caption{Considerations for using ldif2ds or ldapadd}
\end{table}
Table 24. Considerations for using ldif2ds or ldapadd (continued)

<table>
<thead>
<tr>
<th>Considerations</th>
<th>ldif2ds</th>
<th>ldapadd</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invocation</td>
<td>Must be run from z/OS system containing the database, or any image in a Parallel Sysplex running a DB2 subsystem which is a member of the DB2 data sharing group containing the database, using a user ID with DB2 privileges.</td>
<td>Can be run from any LDAP client with appropriate LDAP access.</td>
</tr>
<tr>
<td>Re-usability</td>
<td>Prepared entries are saved so that they can be re-used to reload this system.</td>
<td>No saved output.</td>
</tr>
<tr>
<td>Replication</td>
<td>New entries are not replicated.</td>
<td>New entries can be replicated.</td>
</tr>
</tbody>
</table>

Summary: The ldif2ds utility usage is complex, but it is fast. The ldapadd command usage is simpler, but it is slower.

Recommendation: For one-time additions of 100K or more entries, or for frequent additions of 10K or more entries, use ldif2ds. For infrequent additions of less than 10K entries, use ldapadd. For additions of between 10K and 100K entries, use either ldif2ds or ldapadd.

ldif2ds performance considerations
The ldif2ds utility performance considerations are:

1. ldif2ds uses a lot of storage when adding a large number of entries. Make sure you have sufficient memory available.

2. If the entries to load are in genealogical order, ldif2ds can take certain shortcuts to greatly reduce the storage used during the check and prepare steps. This allows ldif2ds to process more entries at one time. Genealogical order is the order that results from doing a depth-first traversal of a directory: the root is visited, then each subtree directly below the root is traversed. Specify the -g option to indicate that the LDIF input is in this order. The -g option of the ds2ldif utility can be used to unload entries in genealogical order. Do not specify the -g option on ldif2ds if the input is not in genealogical order.

3. If the parent of an entry being added is in the TDBM database, then ldif2ds must also check the database to ensure that the entry itself does not already exist. Therefore, to minimize the database checks, include the parents of the entries being added in the ldif2ds input whenever possible. For example, do not use ldapadd to add a suffix and then use ldif2ds to add all the entries under the suffix. Instead, include the suffix in the ldif2ds input.

4. Do not specify the debug command line option, -d. Usage of debug can impact performance even when there is little debug output. Only specify -d when an error has occurred that you cannot fix.

5. By default, for better load performance ldif2ds sets the LOG NO option in the DB2 Load Utility JCL created during the prepare (-p) step. When the load (-l) step invokes the DB2 Load Utility, the Load Utility will set the copy pending restriction against the table space. The database can be read but cannot be updated until the restriction is removed, for example, by running the DB2 REORG or COPY utility. To avoid entering the copy pending state (for example, when the database already contains a large number of entries), change LOG NO to LOG YES in the JCL. This is done by ldif2ds during the prepare step if the -j option is specified along with the -p option. It can also be performed manually in each of the outHLQ.BULKLOAD.JCL members after the prepare step if the load step is run separately from the prepare step.

ldif2ds normal usage
The typical usage of ldif2ds is:

1. Perform the necessary setup for running ldif2ds, as described in the Preparing to run ldif2ds. This consists of:
   - allocating datasets required by ldif2ds
   - creating the SYSTEM member in the outHLQ.BULKLOAD.JCL dataset
ldif2ds

Note that the same datasets can be used for loading different entries, but the contents of the datasets are overwritten each time.

2. Repeatedly invoke ldif2ds with only the prepare (-p) option until all problems in the LDIF input files have been resolved. This will perform both the check step and the prepare step. You can use the check option (-c) instead if you only want to check the entries at this time, then later invoke ldif2ds again with the -p option.

3. Run ldif2ds with the prepare (-p) option to prepare the load data.

4. Bring the LDAP server down.

5. Even if loading an empty database, make a full image copy of the table spaces for the DIR_DESC, DIR_ENTRY, DIR_LONGATTR, DIR_LONGENTRY, DIR_REPLICA, and the DIR_SEARCH tables. A full image copy is required to help recover from any potential DB2 Load Utility failures. It is performed after the ldif2ds invocation with the prepare (-p) option specified because the prepare (-p) option may update some database tables. Therefore, these updates must be captured for a successful recovery from a DB2 Load Utility failure. If the prepare (-p) and load (-l) options are specified together, the full image copy is performed before the ldif2ds invocation.

6. Review the JCL created by the prepare step in the JCL dataset, outHLQ.BULKLOAD.JCL. Ensure that the DB2 Load Utility datasets are large enough and that the DB2 Load Utility options, especially the LOG value, are acceptable. See note 5 in the ldif2ds performance considerations section for more information about the LOG value.

7. Run ldif2ds once more, with the load (-l) option to load the data into the database. This will submit 6 batch jobs to load the database.

8. Review the output from the load jobs when they end. If the loaded table spaces are in the copy pending state, see DB2 Utility Guide and Reference for instructions on removing that restriction on table spaces. This typically involves running a DB2 utility to create an image copy of the database or reorganize the database (or both).

9. Run the DB2 RUNSTATS utility to reset the statistics used by DB2 to access the database. See Chapter 29, “Performance tuning” for detailed information about running the DB2 RUNSTATS utility.

10. Run the DB2 COPY utility to make a backup image copy of the database.

ldif2ds recovery

The ldif2ds utility can determine whether the submission of the DB2 Load Utility jobs is successful, but it cannot determine if the DB2 Load Utility jobs themselves succeed. In fact, ldif2ds typically terminates before the jobs are finished. Therefore, the final ldif2ds success message displays even if one or more of the jobs eventually fails.

If a DB2 Load Utility job fails, stop all of the DB2 Load Utility invocations that failed, using

-TERM UTIL(BULx)

where x is the last number in the job name of the failing job. Then, there are two alternatives for recovery.

Recovery process 1

The first recovery alternative is not selective in nature and may result in unnecessary reloading of data. This alternative must be used for recovery when:

- The ldif2ds utility is run with the load (-l) option immediately after creating the TDBM database (that is, without first starting the server or running ldif2ds with the check (-c) or prepare (-p) options).
- Loading the DIR_ENTRY, DIR_LONGENTRY, or DIR_LONGATTR table fails and a full image copy of these tables does not exist.

Following is the first recovery alternative process:

1. If full image copies exist for all 6 table spaces, use the DB2 Recover Utility to recover all 6 table spaces from those full image copies. Otherwise, drop and re-create the DIR_DESC, DIR_ENTRY, DIR_LONGATTR, DIR_LONGENTRY, DIR_REPLICA, and DIR_SEARCH tables.
2. See [DB2 Utility Guide and Reference](#) for information about correcting the problems logged in the failing jobs.

3. Run **ldif2ds** with only the load (-l) option again.

**Recovery process 2**

The second recovery alternative requires less processing, but requires a greater understanding of the problem. It cannot be used for recovery when:

- The **ldif2ds** utility is invoked with the load (-l) option immediately after creating the TDBM database.
- Loading the DIR_ENTRY, DIR_LONGATTR, or DIR_LONGENTRY table fails and a full image copy of these tables does not exist.

Following is the second recovery alternative process:

- If the load jobs that failed involve only the DIR_DESC or DIR_SEARCH tables, follow these steps:
  1. If a full image copy exists for the failing table spaces, use the DB2 Recover Utility to recover these table spaces from those full image copies. Otherwise, drop and re-create the failing tables.
  2. See [DB2 Utility Guide and Reference](#) for information about correcting problems logged in the failing jobs.
  3. Manually resubmit the DB2 Load Utility jobs that failed.
  4. If you are sharing TDBM with a z/OS Integrated Security Services LDAP server, make sure that the DIR_DESC table contains a row with the values (-2, -2).

- If the load jobs that failed involve the DIR_ENTRY, DIR_LONGATTR, or DIR_LONGENTRY tables and a full image copy exists (you must use the first recovery alternative if a full image copy does not exist), follow these steps:
  1. If a full image copy exists for the affected table spaces, use the DB2 Recover Utility to recover these table spaces from those full image copies. Otherwise, you must use the first alternative.
  2. See [DB2 Utility Guide and Reference](#) for information about correcting problems logged in the failing jobs.
  3. Manually resubmit the DB2 Load Utility jobs for all the affected tables.
  4. If you are sharing TDBM with a z/OS Integrated Security Services LDAP server, make sure that the DIR_DESC table contains a row with the values (-2, -2).

**Usage**

1. All input files specified with the -i and -e options can be UNIX file system files or datasets. In order to use datasets with **ldif2ds**, you must use a VB record format. Further, separator lines between directory entries in the LDIF file must contain no characters.

2. The **ldif2ds** utility can be invoked to run the check (-c) and prepare (-p) steps while an LDAP server using the same TDBM database is active. The LDAP server **must be down** when the load (-l) step is run.

3. The **LDAP_DEBUG** environment variable can also be used to set the debug level for **ldif2ds**. See [130](#) for more information about specifying the debug level.

4. The DB2 **RUNSTATS** utility should be run when the data is loaded so that DB2 queries are optimized. See [Chapter 29, “Performance tuning”](#) for detailed information about running the DB2 **RUNSTATS** utility.

5. No replication is performed for entries added by **ldif2ds**. Advanced and basic replication entries can be added using **ldif2ds**, but replication does not begin until the LDAP server is started.

6. Referral entries can be added by **ldif2ds**.

7. Only one TDBM database is loaded in an invocation of **ldif2ds**. All the LDIF entries in the LDIF input files and schema input file must contain DNs that belong to the same TDBM database. The parent of each LDIF entry must either be already in the database or must be a prior LDIF entry within these LDIF input files.
ldif2ds

8. ldif2ds cannot be used to modify the LDAP server schema. The schema must already contain all the attributes and object classes used by the entries being added.

9. The aclSource and ownerSource attributes should not be specified in an LDIF entry and are ignored. These attributes are set only by the system.

10. The ldif2ds utility check processing is terminated after 100 syntax errors are detected. The ldif2ds prepare and load processing are terminated after the first error. These values cannot be modified.

11. Because typical ldif2ds usage can involve multiple invocations to check, prepare, and load the entries, ldif2ds maintains a status file to keep information about the last successful step processed. The status file is outHlq.BULKLOAD.JCL(STATUS), where outHlq is the value of the -o option on the command. Any invocation of ldif2ds with the same value for -o is considered a continuation of processing of an earlier invocation.

The status file contains a STATUS record and a TIME record. When the prepare (-p) step is successful, the status is changed to STATUS P and the TIME record is set to the current time. The load step (-l) checks the STATUS record and exits if the status is not P. When the load step is successful, the status is changed to STATUS L and the TIME record is reset to the current time.

12. The prepare step uses the current LDAP server schema to create the load records for the entries to be added. Do not change the schema between the prepare step and the load step. This can result in loading entries that might not be valid for the new schema.

13. During the prepare (-p) step, the RDN of the new LDIF entry is checked to ensure that all the values specified in the RDN are also specified as attributes in the LDIF entry. Missing attributes are added to the entry before it is loaded into the directory. No messages are issued indicating this.

14. The ldif2ds utility encrypts clear text userPassword, secretKey, ibm-replicaKeyPwd, ibm-slapdMasterPw and replicaCredentials attribute values for new entries loaded into the TDBM backend with the pwEncryption and secretEncryption methods specified in the LDAP server configuration file. The ldif2ds utility loads the LDIF format of an encrypted value unloaded by the ldif2ds utility with or without the -t option. When using the ldif2ds utility to load entries that have userPassword attribute values encrypted with crypt, the pwCryptCompat configuration option must be set in the same manner as when the values were originally encrypted. The ldif2ds utility expects that textual data contained within the LDIF file is portable and of UTF-8 origin.

15. If you are maintaining multiple identical TDBM databases, it is recommended that you run the prepare (-p) and load (-l) steps of ldif2ds against each TDBM database. Do not try to use the load data created by running the ldif2ds utility prepare step against one database to load entries on a different database. This can result in an unusable TDBM database.

16. Each loaded entry has a creatorsname, modifiersname, createtimestamp, modiftyimestamp, and ibm-entryuuid attribute. The values for these attributes can be specified in the LDIF input for the entry. If not:
   - The creatorsname and modifiersname attributes are assigned the value of the -b command line option if it is specified; otherwise, they are assigned the value of the adminDN option in the LDAP server configuration file.
   - The createtimestamp and modiftyimestamp attributes are assigned a time set during utility initialization.
   - The ibm-entryuuid attribute is assigned a unique value generated by the utility.

17. When LDIF input contains a replicateOperationalAttributes control and the control has a creatorsname, modifiersname, createtimestamp, or modiftyimestamp attribute value, that attribute value is assigned for the attribute in the entry being loaded. An attribute value assigned from the replicateOperationalAttributes control replaces any other assigned value for that attribute described in the previous usage note.

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

Purpose

The ldapdiff utility is provided to compare two directory subtrees or two schemas on two different directory servers to determine if their contents match. The utility runs as a client operation when both servers are active and determines the differences between the servers. The utility can optionally attempt to synchronize one directory subtree to match the contents of the other.

Format

```
[-n] [-o] [-v] [-x]
 -sh host [-sp port] [-sD bindDn -sw bindPwd]
 -sT truststore -st truststoreType [-sY truststorePwd]
 -sZ
 -ch host [-cp port] [-cD bindDn -cw bindPwd]
 -cT truststore -ct truststoreType [-cY truststorePwd]
 -cZ
```

Parameters

The following table shows the general options that you can use for the ldapdiff utility.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-?</td>
<td>Display the usage.</td>
</tr>
<tr>
<td>-a</td>
<td>Specify that the Server Administration control is to be sent to the consumer server or written to the output LDIF file. This control enables a server that would normally refuse updates, such as a quiesced or replica server, to allow updates. If the -F option is specified, the control is sent to the consumer server. If the -L option is specified, the control is written to the output LDIF file.</td>
</tr>
<tr>
<td>-b baseDn</td>
<td>Indicate the starting point for the subtree comparison. The ldapdiff utility requires either the -b or -S option, or both.</td>
</tr>
<tr>
<td>-C countNumber</td>
<td>Specify the maximum number of non-matching subtree entries that can be found between the two LDAP servers. If the number of non-matching subtree entries exceeds the specified number, the ldapdiff utility exits.</td>
</tr>
<tr>
<td>-d debugLevel</td>
<td>Specify the level of debug messages to be created. The default is no debug messages. Currently, only the ALL level is supported by the ldapdiff utility. This will result in a large amount of debug output being created.</td>
</tr>
<tr>
<td>-F</td>
<td>Specify that the content on the consumer replica server should be modified to match the content of the supplier server. The -F option has no effect when doing schema comparisons as requested by the -S option.</td>
</tr>
<tr>
<td>-j</td>
<td>Indicate that the four operational attributes creatorsName, createTimeStamp, modifiersName, and modifyTimeStamp are to be ignored when comparing subtrees.</td>
</tr>
<tr>
<td>-L filename</td>
<td>Specify the name of the output LDIF file to contain the differences between the supplier and consumer servers. The output LDIF file can be used to update the consumer server to eliminate the differences between the supplier and consumer. The name cannot be an MVS dataset.</td>
</tr>
<tr>
<td>-n</td>
<td>Specify that the Do Not Replicate control is to be sent to the consumer server or written to the output LDIF file. This control prevents the consumer server from sending replicated entries to the next tier of advanced replication servers. If the -F option is specified, the control is sent to the consumer server. If the -L option is specified, the control is written to the output LDIF file.</td>
</tr>
</tbody>
</table>
### ldapdiff

### Table 25. ldapdiff general options (continued)

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-O</td>
<td>Indicate that directory subtrees are compared using checksum only. This option can be used to quickly detect differences between servers without the need to retrieve all attribute types and values. DNs for non-matching entries between the supplier and consumer servers are displayed, but not the non-matching attributes. Both servers must support entry checksum. Otherwise, the ldapdiff utility does not start. The -O option overrides the -F, -j, and -L options.</td>
</tr>
<tr>
<td>-S</td>
<td>Specify that the schema is to be compared between the supplier and consumer servers. The ldapdiff utility requires either the -b or -S option, or both.</td>
</tr>
<tr>
<td>-v</td>
<td>Use verbose mode, with many diagnostics written to standard output.</td>
</tr>
<tr>
<td>-x</td>
<td>Specify that extra entries on the consumer server are to be ignored.</td>
</tr>
</tbody>
</table>

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

### Table 26. ldapdiff supplier server options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-sD bindDn</td>
<td>Specify the DN to use to bind to the supplier server. The bindDn parameter should be a string-represented DN. The default is an empty string.</td>
</tr>
<tr>
<td>-sh host</td>
<td>Specify the hostname or IP address on which the supplier server is running. There is no default host, therefore, this option is required.</td>
</tr>
<tr>
<td>-sK keystore</td>
<td>Specify the filename of the Java keystore or RACF key ring that contains the client certificate. If the supplier server is configured to perform server authentication only, a client certificate is not required. If the supplier server is configured to perform client and server authentication, a client certificate might be required. The filename cannot be an MVS data set.</td>
</tr>
<tr>
<td>-sN keystoreType</td>
<td>Specify the type of the keystore being used by the -sK option. The supported keystore types are: JCERACFKS (RACF key ring) JCEKS (Java keystore) This is a required option if the -sK option is specified. This option cannot be specified by itself.</td>
</tr>
<tr>
<td>-sp port</td>
<td>Specify the TCP port where the supplier server is listening. The default LDAP non-secure port is 389 and the default LDAP secure port is 636.</td>
</tr>
<tr>
<td>-sp keystorePwd</td>
<td>Specify the keystore password. This password is required to access the encrypted information in the keystore file, which may include one or more private keys. If the keystore specified by -sK is a RACF key ring, then this option should not be specified. This option is not allowed if the -sK option is not specified.</td>
</tr>
</tbody>
</table>
Table 26. ldapdiff supplier server options (continued)

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-st truststoreType</td>
<td>Specify the type of the truststore used by the -sT option. The supported truststore types are: JCERACFKS (RACF key ring) JCEKS (Java keystore)</td>
</tr>
<tr>
<td></td>
<td>This is a required option if the -sT option is specified. This option cannot be specified by itself.</td>
</tr>
<tr>
<td>-sT truststore</td>
<td>Specify the filename of the Java keystore or RACF keying that contains the CA (certificate authority) certificate that signed the supplier server’s certificate.</td>
</tr>
<tr>
<td></td>
<td>The filename cannot be an MVS data set.</td>
</tr>
<tr>
<td></td>
<td>If specifying a RACF key ring, use the following name format: safkeyring://userid/keyFile</td>
</tr>
<tr>
<td></td>
<td>where, userid is the RACF userid that owns the RACF keyring and keyFile is the name of the RACF keyring.</td>
</tr>
<tr>
<td></td>
<td>If you have ownership of the RACF key ring, then a simplified format may be used: safkeyring:///keyFile</td>
</tr>
<tr>
<td></td>
<td>where, keyFile is the name of the RACF keyring.</td>
</tr>
<tr>
<td></td>
<td>This option is required when using SSL. This option enables a secure connection even if the -sZ option is not specified.</td>
</tr>
<tr>
<td>-sw bindPwd</td>
<td>Specify the bind password for simple authentication. The default is an empty string.</td>
</tr>
<tr>
<td>-sY truststorePwd</td>
<td>Specify the truststore password. This password is required to access the encrypted information in the truststore file, which may include one or more private keys. If the truststore specified by -sT is a RACF key ring, then this option should not be specified.</td>
</tr>
<tr>
<td></td>
<td>This option is not allowed if the -sT option is not specified.</td>
</tr>
<tr>
<td>-sZ</td>
<td>Use a secure connection to communicate with the supplier server. Secure connections expect the communication to begin with the SSL/TLS handshake.</td>
</tr>
</tbody>
</table>

The following options apply to the consumer server and are denoted by an initial ‘c’ in the option name:

Table 27. ldapdiff consumer server options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-cD bindDn</td>
<td>Specify the DN to use to bind to the consumer server. The bindDn parameter should be a string-represented DN. The default is an empty string.</td>
</tr>
<tr>
<td>-ch host</td>
<td>Specify the host on which the consumer server is running. There is no default host, therefore, this option is required.</td>
</tr>
</tbody>
</table>
Table 27. **ldapdiff** consumer server options (continued)

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-cK keystore</td>
<td>Specify the filename of the Java keystore or RACF key ring that contains the client certificate. If the consumer server is configured to perform server authentication only, a client certificate is not required. If the consumer server is configured to perform client and server authentication, a client certificate might be required. The filename cannot be an MVS data set. If specifying a RACF key ring, use the following name format: safkeyring://userid/keyFile where, userid is the RACF userid that owns the RACF keyring and keyFile is the name of the RACF keyring. If you have ownership of the RACF key ring, then a simplified format may be used: safkeyring:///keyFile where, keyFile is the name of the RACF keyring.</td>
</tr>
</tbody>
</table>
| -cN keystoreType | Specify the type of the keystore being used by the -cK option. The supported keystore types are:  
|       | JCERACFKS (RACF key ring)  
|       | JCEKS (Java keystore)  
|       | This is a required option if the -cK option is specified. This option cannot be specified by itself. |
| -cp port | Specify the TCP port where the consumer server is listening. The default LDAP non-secure port is 389 and the default LDAP secure port is 636. |
| -cP keystorePwd | Specify the keystore password. This password is required to access the encrypted information in the keystore file, which may include one or more private keys. If the keystore specified by -cK is a RACF key ring, then this option should not be specified. This option is not allowed if the -cK option is not specified. |
| -ct truststoreType | Specify the type of the truststore used by the -cT option. The supported truststore types are:  
|       | JCERACFKS (RACF key ring)  
|       | JCEKS (Java keystore)  
|       | This is a required option if the -cT option is specified. This option cannot be specified by itself. |
| -cT truststore | Specify the filename of the Java keystore or RACF keying that contains the CA (certificate authority) certificate that signed the consumer server’s certificate. The filename cannot be an MVS data set. If specifying a RACF key ring, use the following name format: safkeyring://userid/keyFile where, userid is the RACF userid that owns the RACF keyring and keyFile is the name of the RACF keyring. If you have ownership of the RACF key ring, then a simplified format may be used: safkeyring:///keyFile where, keyFile is the name of the RACF keyring. |
| -cw bindPwd | Specify the bind password for simple authentication. The default is an empty string. |
**Table 27. ldapdiff consumer server options (continued)**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-cY truststorePwd</td>
<td>Specify the truststore password. This password is required to access the encrypted information in the truststore file, which may include one or more private keys. If the truststore specified by <code>-cT</code> is a RACF key ring, then this option should not be specified. This option is not allowed if the <code>-cT</code> option is not specified.</td>
</tr>
<tr>
<td>-cZ</td>
<td>Use a secure connection to communicate with the consumer server. Secure connections expect the communication to begin with the SSL/TLS handshake.</td>
</tr>
</tbody>
</table>

All other command line inputs will result in a syntax error message and the correct syntax is displayed. Also, specifying the same option multiple times will result in the last specified option being used.

**Examples**

Assume that two servers are set up, one as a supplier server on IP address 1.2.3.4 and other as a consumer server on IP address 1.2.3.5, and that the suffix `o=sample` is present on both servers. Assume the entries in the supplier.ldif file are in the supplier server and the entries in the consumer.ldif file are in the consumer server.

- **Contents of supplier.ldif file:**
  ```
  dn: cn=Entry1,o=sample
  objectclass: inetOrgPerson
  objectclass: organizationalPerson
  objectclass: person
  objectclass: top
  objectclass: ePerson
  sn: entry1
  cn: Entry1

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

- **Contents of consumer.ldif file:**
  ```
  dn: cn=Entry2,o=sample
  objectclass: inetOrgPerson
  objectclass: organizationalPerson
  objectclass: person
  objectclass: top
  objectclass: ePerson
  sn: abcd
  cn: Entry2

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

The following command runs the **ldapdiff** utility to modify the contents of the `o=sample` subtree on the consumer server (even if it is quiesced) to match the supplier server, binding to the supplier server on IP address 1.2.3.4 with a bind DN of `cn=admin` and a password of `supplier` and binding to the consumer server on IP address 1.2.3.5 with a bind DN of `cn=admin` and a password of `consumer`:
The resulting actions are:

1. The `cn=Entry1,o=sample` is added to the consumer server. This entry is already present on the supplier server but was not on the consumer server.
2. The `cn=Entry2,o=sample` is modified on the consumer server. The `sn` attribute value on the `cn=Entry2,o=sample` entry on the consumer server gets modified to match the value on the supplier server.
3. The `cn=Entry3,o=sample` is deleted from the consumer server. This entry does not exist on the supplier server and is deleted from the consumer server so that the same set of entries are present on both the supplier and consumer servers.

Usage

1. The `ldapdiff` utility is a Java based utility and requires IBM 31-bit SDK for z/OS, Java 2 Technology Edition, V5 (5655-N98) or later to be installed on the system.
2. The `ldapdiff` utility is used to bring a supplier and a consumer server in sync before starting advanced replication. The `ldapdiff` utility requires that the base DN, which is being compared, exist on at least one of the servers.
3. The `ldapdiff` utility is a diagnostic and corrective tool. It is not designed to be run as routine maintenance. If there are out of sync replication-related errors observed, an LDAP administrator might decide to run the utility to correct the out of sync condition.
4. When the `ldapdiff` utility is run, it is strongly recommended that there be no updates made to either of the targeted LDAP servers. If the `ldapdiff` utility is run while updates are being made, it cannot be guaranteed that all discrepancies are accurately reported or fixed. The LDAP administrator needs to manually quiesce or suspend all update activity to the two subtrees being compared on the two servers. The `ldapdiff` utility does not check at initialization time to determine if the servers are quiesced or not. The replication context subtree can be quiesced by using the `Quiesce` or `unquiesce context` extended operation on the `ldapexop` utility. See [ldapexop utility] for information on the `Quiesce` or `unquiesce context` extended operation.
5. The fix option `-F` should be used with caution because it automatically synchronizes all the entries from the supplier server to the consumer server starting from the `baseDn`. This synchronization can result in unintentional results on the consumer server such as the deletion of replication topology entries, RACF users, groups, user-group connections, resource profiles, and other entries. It is recommended that the `-L` option be used first to write any differences between the supplier and consumer servers in LDIF file format. The LDAP administrator should then analyze the output LDIF file to ensure that the updates are correct. If they are correct, the changes can be manually applied to the consumer server using the `ldapmodify` command or the `ldapdiff` utility can be re-run with the fix option `-F` specified.
   - If the `-F` option is specified, use the `Server Administration` control option `-a` to enable the `ldapdiff` utility to write to a read-only replica. The `Server Administration` control also allows `ldapdiff` to modify operational attributes such as `aclPropagate`, `aclSource`, `createTimeStamp`, `creatorsName`, `entryOwner`, `ibm-entryuuid`, `modifiersName`, `modifyTimeZone`, `ownerPropagate`, and `ownerSource`.
6. The `ldapdiff` utility performs two passes to get the supplier and consumer servers back in sync.
   - In the first pass, the `ldapdiff` utility traverses the supplier server and does the following:
     - Checks for any extra entries on the supplier server that are not on the consumer server. The extra entries are added to the consumer server if the `-F` option is specified.
     - Compares the entries that exist on both the supplier and consumer servers. If there are mismatches, the consumer server entries are modified to match the supplier server entries if the `-F` option is specified.
   - In the second pass, the `ldapdiff` utility traverses the consumer server to check for any extra entries on the consumer server. The extra entries are deleted from the consumer server if the `-F` option is specified. The second pass can be skipped by specifying the `-x` option.
7. The supplier and consumer bind DNs (-sD and -cD options) must be able to read the attributes in the entries being compared. In addition, if the fix option (-F) is specified, the consumer bind DN must also be able to update attributes and add and delete entries.

8. The ldapdiff utility traverses each entry in the directory subtree on the supplier server and compares its contents with the corresponding entry on the consumer server. Because each entry needs to be retrieved, running the ldapdiff utility can take a long time and can generate lots of read requests to the supplier and consumer servers. Depending on how many differences are found and whether the fix option (-F) is specified, the ldapdiff utility can also generate an equal amount of update (for example add, delete, modify) requests to the consumer server. Ideally, the ldapdiff utility should only be used once between servers, when replication is initially setup. For example, if your replication topology has two peer masters and two replica servers, you might want to run the ldapdiff utility between peer 1 and peer 2. Then, if replication is suspended, run ldapdiff concurrently between peer 1 and replica 1 and between peer 2 and replica 2. If replica servers are out of sync with their master servers, the ldapdiff utility can be run to identify and correct out of sync conditions.

9. The ldapdiff utility displays a message after it has finished comparing every 100th entry.

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

Idapexop utility

Purpose
The Idapexop utility provides an interface to the ldap_extended_operation() API. The utility runs as a client operation while the server is active. The Idapexop utility only performs the following advanced replication extended operations:
- cascrepl - Cascading control replication extended operation
- controlqueue - Control replication queue extended operation
- controlrepl - Control replication extended operation
- controlreplerr - Control replication error log extended operation
- quiesce - Quiesce or unquiesce context extended operation
- repltopology - Replication topology extended operation

Format
{ [-m EXTERNAL -Z [-K keyFile] [-P keyFilePwd] [-N certificateLabel]]
 [ [ [ [-D dn -w password] ]
   [-m CRAM-MD5 -w password [ [-D dn] | [-U userName [-D dn]] ]]
   [-m DIGEST-MD5 -U userName -w password [ [-D dn] [-G realmName] ]]
   [-m GSSAPI]
 ] [-Z [-K keyFile] [-P keyFilePwd] [-N certificateLabel]]
} -op extendedOperation extOptions

Parameters
The options for the Idapexop utility are divided into two categories:
1. There are general options that specify how to connect to the LDAP server. These options must be specified before the -op option. See Table 28 for an explanation about the general options.
2. There are extended operation options that identify the extended operation to be performed. These options are specified after the general options and must begin with the -op option. See Extended operations options for an explanation about these options.

All other command line inputs result in a syntax error message, after the proper syntax is displayed. Except for the -op option, if the same option is specified multiple times, the last value specified is used. The -op option must be specified only once.

Table 28. Idapexop general options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-?</td>
<td>Display the usage.</td>
</tr>
<tr>
<td>-d debugLevel</td>
<td>Specify the level of debug messages to be created. The debugLevel is specified in the same fashion as the debug level for the LDAP server, as described on page 130.</td>
</tr>
<tr>
<td>-D dn</td>
<td>Specify the DN to use to bind to the LDAP directory. The dn parameter should be a string-represented DN. The default is a NULL string. If the -m option is equal to DIGEST-MD5 or CRAM-MD5, this option is the authorization DN which is used for making access checks. This directive is optional when used in this manner.</td>
</tr>
<tr>
<td>-G realmName</td>
<td>Specify the realm name to use when doing a DIGEST-MD5 bind. This option is required when multiple realms are passed from an LDAP server to a client as part of a DIGEST-MD5 challenge; otherwise, it is optional.</td>
</tr>
<tr>
<td>-h host</td>
<td>Specify the hostname or IP address on which the LDAP server is running. The default is the local host.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>-help</td>
<td>Display the usage.</td>
</tr>
<tr>
<td>-K keyFile</td>
<td>Specify the name of the System SSL key database file, RACF key ring, or PKCS #11 token. If this option is not specified, this utility looks for the presence of the SSL_KEYRING environment variable with an associated name.</td>
</tr>
<tr>
<td></td>
<td>If keyFile is specified as <em>TOKEN</em>/NAME, then System SSL uses the specified PKCS #11 token. Otherwise, System SSL uses a key database file or a RACF key ring. In this case, System SSL first assumes that keyFile is a key database file name and tries to locate the file. If keyFile is not a fully-qualified Unix System Services file name, the current directory is assumed to contain the key database file. The name cannot be an MVS data set. If System SSL cannot locate the file, it then assumes that keyFile is a RACF key ring name. See SSL/TLS information for LDAP utilities for information about System SSL key databases, RACF key rings, and PKCS #11 tokens. This parameter is ignored if -Z is not specified.</td>
</tr>
<tr>
<td>-m mechanism</td>
<td>Specify the bind method to use. Specify GSSAPI to indicate a Kerberos Version 5 bind is requested, EXTERNAL to indicate that a certificate (SASL external) bind is requested, CRAM-MD5 to indicate that a SASL Challenge Response Authentication Mechanism bind is requested, or DIGEST-MD5 to indicate a SASL digest bind is requested. The GSSAPI method requires a protocol level of 3 and the user must have a valid Kerberos Ticket Granting Ticket in their credentials cache by using the Kerberos kinit command line utility. The EXTERNAL method requires a protocol level of 3. You must also specify -Z, -K, and -P to use certificate bind. Unless you want to use the default certificate in the key database file, RACF key ring, or PKCS #11 token, use the -N option to specify the label of the certificate. The CRAM-MD5 method requires protocol level 3. The -D or -U option must be specified. The DIGEST-MD5 method requires protocol level 3. The -U option must be specified. The -D option can optionally be used to specify the authorization DN. If -m is not specified, a simple bind is performed.</td>
</tr>
<tr>
<td>-N certificateLabel</td>
<td>Specify the label associated with the certificate in the key database file, RACF key ring, or PKCS #11 token. This parameter is ignored if -Z is not specified</td>
</tr>
<tr>
<td>-p port</td>
<td>Specify the TCP port where the LDAP server is listening. The default LDAP non-secure port is 389 and the default LDAP secure port is 636.</td>
</tr>
<tr>
<td>-P keyFilePwd</td>
<td>Specify either the key database file password or the file specification for a System SSL password stash file. When the stash file is used, it must be in the form file:// followed immediately (no blanks) by the file system file specification (for example, file://etc/ldap/sslstashfile). The stash file must be a file and cannot be an MVS dataset. This parameter is ignored if -Z is not specified.</td>
</tr>
<tr>
<td>-U userName</td>
<td>Specify the user name for CRAM-MD5 or DIGEST-MD5 binds. The userName is a short name (uid) that is used to perform bind authentication. This option is required if the -m option is set to DIGEST-MD5.</td>
</tr>
<tr>
<td>-v</td>
<td>Use verbose mode, with many diagnostics written to standard output.</td>
</tr>
</tbody>
</table>
Table 28. ldapexop general options (continued)

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-w password</td>
<td>Specify the bind password for simple, CRAM-MD5, and DIGEST-MD5 authentication. The default is a NULL string. Specify ? to prompt for the password.</td>
</tr>
<tr>
<td>-Z</td>
<td>Use a secure connection to communicate with the LDAP server. Secure connections expect the communication to begin with the SSL/TLS handshake. The -K keyFile option or equivalent environment variable is required when the -Z option is specified. The -P keyFilePwd option is required when the -Z option is specified and the key file specifies a file system key database file. Unless you want to use the default certificate in the key database file, RACF key ring, or PKCS #11 token, use the -N option to specify the label of the certificate.</td>
</tr>
</tbody>
</table>

**Extended operations options:** The -op extendedOperation option identifies the extended operation to be performed. The extOptions indicate the required and optional options for the specific extended operation. Each extended operations that is supported by the ldapexop utility is documented in detail in Appendix D, "Supported extended operations." The supported extended operations and their options are:

```bash
cascrepl -action { quiesce | unquiesce | replnow | wait } -rc contextDn [-timeout secs]
```

Perform the Cascading control replication extended operation. The requested action is applied to the specified server and also passed along to all replicas of the given context (subtree). If any of these are forwarding replicas or gateway servers, they pass the extended operation along to their replicas.

The operation cascades over the entire advanced replication topology. See Cascading control replication for more information about the Cascading control replication extended operation.

- action { quiesce | unquiesce | replnow | wait }

Specifies the Cascading control replication extended operation action to be performed.

- quiesce

Halts further updates, except by replication. Client updates under the specified context are restricted to the LDAP administrator, if using the Server Administration control (OID 1.3.18.0.2.10.15), and any replication master DNs with authority under this context.

- unquiesce

Resumes normal operation, client updates are accepted.

- replnow

Replicates all queued changes to all replica servers as soon as possible, regardless of schedule. This operation is propagated to the consumer server of each replication agreement without waiting for all queued updates to be applied.

- wait

Replicates all queued changes to all replica servers as soon as possible, regardless of schedule. This operation is propagated to the consumer server of each replication agreement after all queued updates for that agreement are applied.

-rc contextDn

Specifies the DN of the advanced replication context (subtree).

- timeout secs

Specifies the number of seconds that the extended operation has to successfully complete. If not present, or 0, the operation has an indefinite amount of time to complete.

This example performs the Cascading control replication extended operation to wait for all updates to be replicated to each consumer server in the o=acme,c=us replication context. If the extended operation does not successfully complete in 60 seconds, the operation ends with a return code of LDAP_TIMEOUT.

```
ldapexop -D adminDn -w adminPw -op cascrepl -action wait -rc "o=acme,c=us" -timeout 60
```
### controlqueue -skip { all | changeId } -ra agreementDn

Perform the **Control replication queue** extended operation. This extended operation indicates which pending changes in the advanced replication queue should be skipped and not replicated to the consumer server. See [Control replication queue](#) for more information about the **Control replication queue** extended operation.

- **-skip {all | changeId}**
  - **all** Deletes (skips) all changes currently queued for replication for the specified replication agreement DN.
  - **changeId** Deletes (skips) the change queued for replication with the change ID matching the specified number. The number must be in the range 1 - 4294967295. Change IDs can be determined by searching the `agreementDn` entry for the `ibm-replicationPendingChanges` operational attribute. Only the next change to be replicated can be skipped in this manner.

- **-ra agreementDn**
  Specifies the DN of the advanced replication agreement.

This example performs the **Control replication queue** extended operation to delete (skip) all pending updates in the advanced replication queue for the replication agreement, `cn=server3,ibm-replicaSubentry=master1-id,ibm-replicaGroup=default,o=acme,c=us`. This extended operation prevents all changes in the replication queue from being replicated to all consumer servers.

```
ldapexop -D adminDn -w adminPw -op controlqueue -skip all -ra "cn=server3,ibm-replicaSubentry=master1-id,ibm-replicaGroup=default,o=acme,c=us"
```

### controlrepl -action { suspend | resume | replnow } { -rc contextDn | -ra agreementDn }

Perform the **Control replication** extended operation. This extended operation indicates whether to suspend, resume, or immediately replicate entries in the advanced replication context (subtree) or agreement. See [Control replication](#) for more information about the **Control replication** extended operation.

- **-action { suspend | resume | replnow }**
  Specifies the **Control replication** extended operation action to be performed.
  - **suspend** Suspends advanced replication for the replication agreement or context (subtree). Any updates under the replication agreement or context are queued until the **Control replication** extended operation is performed to resume updates to consumer servers.
  - **resume** Resumes advanced replication for the replication agreement or context (subtree).
  - **replnow** Replicates immediately any outstanding updates for the replication agreement or context (subtree), regardless of schedule. **replnow** has no effect on a suspended replication agreement or context.

- **-rc contextDn | -ra agreementDn**
  Specifies the DN of the advanced replication context (subtree) or agreement the action is to be performed against. To perform the action against all agreements under a replication context, use **-rc contextDn** with the DN of the replication context. Alternatively, to perform this action against a single agreement, use **-ra agreementDn** with the DN of the replication agreement. A **contextDn** and an **agreementDn** cannot both be specified.

This example performs the **Control replication** extended operation to suspend advanced replication on the replication agreement, `cn=server3,ibm-replicaSubentry=master1-id,ibm-replicaGroup=default,o=acme,c=us`.

```
ldapexop -D adminDn -w adminPw -op controlrepl -action suspend -ra "cn=server3,ibm-replicaSubentry=master1-id,ibm-replicaGroup=default,o=acme,c=us"
```
**Idapexop**

```bash
controlreplerr -ra agreementDn { -delete { failureId | all } | -retry { failureId | all } | -show failureId }
```

Perform the **Control replication error log** extended operation. This extended operation indicates whether to delete, retry, or show a failed advanced replication update. Failing operations can be determined by searching the `agreementDn` entry for the `ibm-replicationFailedChanges` operational attribute which contains the failing change ID. See [Control replication error log](#) for more information about the **Control replication error log** extended operation.

- **-ra agreementDn**
  Specifies the DN of the advanced replication agreement.

- **-delete { failureId | all }**
  Removes one or all failed replication updates.

  - **failureId**
    Deletes only the failed update specified by the `failureId` for this agreement. The `failureId` must be in the range 1 - 4294967295.

  - **all**
    Deletes all the failed updates for this agreement.

- **-retry { failureId | all }**
  Reprocesses one or all failed replication updates.

  - **failureId**
    Retries only the failed update specified by the `failureId` for this agreement. The `failureId` must be in the range 1 - 4294967295.

  - **all**
    Retries all the failed updates for this agreement.

- **-show failureId**
  Shows the failed update specified by the `failureId` for this agreement. The `failureId` must be in the range 1 - 4294967295.

This example performs the **Control replication error log** extended operation to remove all failures from the replication error log for the replication agreement, `cn=server3,ibm-replicaSubentry=master1-id,ibm-replicaGroup=default,o=acme,c=us`.

```
ldapexop -D adminDn -w adminPw -op controlreplerr -delete all -ra "cn=server3,ibm-replicaSubentry=master1-id,ibm-replicaGroup=default,o=acme,c=us"
```

**quiesce -rc contextDn [-end]**

Perform the **Quiesce or unquiesce context** extended operation. When an advanced replication context is quiesced, update operations are only accepted from certain users. See [Quiesce or unquiesce context](#) for more information about the **Quiesce or unquiesce context** extended operation.

- **-rc contextDN**
  Specifies the DN of the advanced replication context (subtree) to be quiesced or unquiesced.

- **-end**
  Unquiesces the advanced replication context (subtree). If not specified, the default is to quiesce the replication context.

This example performs the **Quiesce or unquiesce context** extended operation to quiesce the replication context `o=acme,c=us`. After this extended operation is performed, updates to entries within the replication context are only allowed from certain users.

```
ldapexop -D adminDn -w adminPw -op quiesce -rc "o=acme,c=us"
```

**repltopology -rc contextDn [-timeout secs] [-ra agreementDn]**

Perform the **Replication topology** extended operation. This extended operation synchronizes all replication topology related entries under the specified context DN. See [Replication topology](#) for more information about the **Replication topology** extended operation.

- **-rc contextDN**
  Specifies the DN of the advanced replication context (subtree) on the supplier server for which
advanced replication topology related entries are synchronized. The extended operation is
cascaded through all forwarding and gateway servers.

- **timeout** *secs*
  Specifies the number of seconds that the extended operation has to successfully complete. If
  not present, or 0, the operation has an indefinite amount of time to complete.

- **ra** *agreementDn*
  Specifies the DN of the advanced replication agreement and allows the operation to be
  restricted to only one server and its consumer.

This example performs the **Replication topology** extended operation to synchronize replication
topology entries within the o=acme,c=us replication context with the consumer defined in the replication
agreement cn=server3,ibm-replicaSubentry=master1-id,ibm-replicaGroup=default,o=acme,c=us.
This extended operation does not cascade, it only modifies the consumer defined in the replication
agreement before returning.

```
ldapexop -D adminDn -w adminPw -op repltopology -rc "o=acme,c=us"
-ra "cn=server3,ibm-replicaSubentry=master1-id,ibm-replicaGroup=default,o=acme,c=us"
```

**Usage**

1. You must be bound as the LDAP administrator to successfully perform these extended operations
   against the z/OS LDAP server.
2. The LDAP_DEBUG environment variable can be used to set the debug level. For more information on
   specifying the debug level using keywords, decimal, hexadecimal, and plus and minus syntax, see
   [Table 20](#table_20) for a list of specific debug levels.
3. If you are attempting a CRAM-MD5 authentication bind to a non-z/OS version of IBM Tivoli Directory
   Server, see [CRAM-MD5 authentication support](#cram-md5-authentication-support) for more information.
4. You can specify an LDAP URL for host on the **-h** parameter. See [ldap_init()](#ldap_init) in [IBM Tivoli Directory
   Server Client Programming](#server-client-programming) for more information.

**Diagnosis**

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

See [Appendix D, “Supported extended operations”](#appendix_d) for additional diagnostic information about each of the
extended operations.
ldapexop
Chapter 12. Internationalization support

This topic discusses translated messages and UTF-8 support.

Translated messages

The LANG and NLSPATH environment variables are set for the z/OS LDAP server and utility programs in the LDAP environment variables file. The default name and location for this file is:

/etc/ldap/ds.envvars

A sample ds.envvars file is shipped in /usr/lpp/ldap/etc. You can copy this file to /etc/ldap and modify its contents. Following is part of the sample file:

NLSPATH=/usr/lib/nls/msg/%L/%N
LANG=En_US.IBM-1047

There are no default values for these variables. Messages are also available in Japanese. The LANG variable should be set to LANG=Ja_JP or Ja_JP.IBM-939. These variables should also be set either in the environment variable file of the user or by exporting the variables in the shell for the user ID that runs the LDAP directory utilities.

There are symbolic links to the English language message catalogs in the following locations:

- /usr/lib/nls/msg/C
- /usr/lib/nls/msg/En_US
- /usr/lib/nls/msg/En_US.IBM-1047

It is possible to run with either LANG=En_US, LANG=C, or LANG=En_US.IBM-1047 and access the English language LDAP message catalogs.

There are also symbolic links to the following Japanese language message catalogs:

- /usr/lib/nls/msg/Ja_JP
- /usr/lib/nls/msg/Ja_JP.IBM-939

UTF-8 support

UTF stands for “UCS (Unicode) Transformation Format”. The UTF-8 encoding can be used to represent any Unicode character. Depending on a Unicode character’s numeric value, the corresponding UTF-8 character is a 1, 2, or 3 byte sequence. Table 29 shows the mapping between Unicode and UTF-8. See RFC 2279: UTF-8, a transformation format of ISO 10646 and RFC 2253: Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names for more information about UTF-8.

Table 29. Mapping between Unicode and UTF-8

<table>
<thead>
<tr>
<th>Unicode range (hexadecimal)</th>
<th>UTF-8 octet sequence (binary)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000-007F</td>
<td>0xxxxx</td>
</tr>
<tr>
<td>0080-07FF</td>
<td>110xxxxx 10xxxxxx</td>
</tr>
<tr>
<td>0800-FFFF</td>
<td>1110xxxx 10xxxxxx 10xxxxxx</td>
</tr>
</tbody>
</table>

The LDAP Version 3 protocol specifies that all data exchanged between LDAP clients and servers be UTF-8. The LDAP server supports UTF-8 data exchange as part of its Version 3 protocol support.
Note: For UTF-8 data stored in a LDAP server’s TDBM and GDBM (when DB2-based) backends, collation for single-byte UTF-8 characters is relative to the server’s locale. For multi-byte UTF-8 characters, collation is relative to the numeric value of the equivalent Unicode character.
Part 2. Use
Chapter 13. Data model

The LDAP data model is closely aligned with the X.500 data model. In this model, a directory service provides a hierarchically organized set of entries. Each of these entries is represented by an object class. The object class of the entry determines the set of attributes which are required to be present in the entry as well as the set of attributes that can optionally appear in the entry. An attribute is represented by an attribute type and one or more attribute values. In addition to the attribute type and values, each attribute has an associated syntax which describes the format of the attribute values. Examples of attribute syntaxes for LDAP directory include directory string and binary.

To summarize, the directory is made up of entries. Each entry contains a set of attributes. These attributes can be single or multi-valued (have one or more values associated with them). The object class of an entry determines the set of attributes that must exist and the set of attributes that may exist in the entry. Refer to [z/OS DCE Application Development Guide: Directory Services] for more information about the X.500 Directory Information Model.

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
```

The order of the component attribute=value pairs is important. The DN contains one component for each level of the directory hierarchy. LDAP directory DNs begin with the most specific attribute (typically some sort of name), and continue with progressively broader attributes, often ending with a country attribute.

Relative distinguished names

Each component of a DN is referred to as a relative distinguished name (RDN). It identifies an entry distinctly from any other entries which 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). The attribute=value pair or pairs making up the RDN for an entry must also be present as an attribute=value pair or pairs in the entry. This is not true of the other components of the DN. The TDBM, LDBM, and CDBM backends add the attribute=value pairs in the RDN to the entry if they are not already present.

RDNs can contain multiple attribute=value pairs. So-called multivalued RDNs use two or more attribute=value pairs from the directory entry to define the name of the entry relative to its parent. An example where this would be useful would be where a directory hierarchy of users was being defined for a large university. This hierarchy would be segmented by campus. A problem is encountered, however, when it is discovered that there is more than one John Smith at the downtown campus. The RDN cannot simply be the name of the user. What can be done, however, is to add a unique value to the RDN, therefore, ensuring its uniqueness across the campus. Typically universities hand out serial numbers to their students. Coupling the student number with the person’s name is one method of solving the problem of having a unique RDN under a parent in the directory hierarchy. The entry’s RDN might look something like:

```
cn=John Smith+studentNumber=123456.
```

The plus sign (+) is used to delimit separate attribute=value pairs within an RDN. The entry’s DN might look like:

```
cn=John Smith+studentNumber=123456, ou=downtown, o=Big University, c=US
```

Any attribute can be used to make up an RDN except:
- attributes with binary syntax, UTC time syntax, or generalized time syntax.
**Note:** The userPassword attribute is binary, therefore, it cannot appear in an RDN. Time stamp attributes use one of the time syntaxes, therefore, they cannot appear in an RDN.

- attributes that are marked NO-USER-MODIFICATION in the schema, because these attributes cannot be added to an entry by a user.
- the aclEntry, aclPropagate, entryOwner, and ownerPropagate attributes.

### Distinguished name syntax

The Distinguished Name (DN) syntax supported by this server is based on IETF RFC 2253 LDAP (v3): UTF-8 String Representation of Distinguished Names. A semicolon (;) character may be used to separate RDNs in a distinguished name, although the comma (,) character is the typical notation. A plus sign (+) is used to separate attribute=value pairs in an RDN.

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

In addition, space characters may be present between an attribute=value pair and a plus sign (+), between an attribute type and an equal sign (=), and between an equal sign (=) and an attribute value. These space characters are ignored when parsing.

A value may be surrounded by quotation marks, which are not part of the value. Inside the quoted value, the following characters can occur without any escaping:

- A space or pound sign (#) character occurring at the beginning of the string
- A space character occurring at the end of the string
- One of the characters
  - apostrophe (‘)
  - equal sign (=)
  - plus sign (+)
  - backslash (\)
  - less than sign (<)
  - greater than sign (>)
  - semicolon (;)

Alternatively, a single character to be escaped may be prefixed by a backslash (\). This method may be used to escape any of the characters listed above, plus the quotation mark. Pound signs (#) and space characters that do not occur at the beginning of a string can also be escaped, but this is not required.

This notation is designed to be convenient for common forms of name. This section gives a few examples of distinguished names written using this notation. First is a name containing three components:

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

This example shows a method of escaping a comma in an organization name:

```
CN=R. Smith,O=Big Company\, Inc.,C=US
```

### Domain component naming

Domain component naming as specified by RFC 2247 is also supported in the LDAP server. For example, the domain name ibm.com could be specified as an entry in the LDAP server with the following distinguished name:

```
dc=ibm,dc=com
```

### RACF-style distinguished names

If you are using SDBM (the RACF database backend of the LDAP server), the format of the DNs is restricted in order to match the schema of the underlying RACF data.
A RACF-style DN for a user or group contains two required attributes plus a suffix:

racfid  Specifies the user ID or group ID.

profiletype
  Specifies user or group.

suffix  Specifies the SDBM suffix.

A RACF-style DN for a user’s connection to a group contains three required attributes plus a suffix:

racuserid+racgroupid
  Specifies the user and the group.

profiletype
  Specifies connect.

suffix  Specifies the SDBM suffix.

A RACF-style DN for a general resource profile contains two required attributes plus a suffix:

profilename
  Specifies the name of the resource profile. The case of the name is important if the class containing the resource profile supports mixed case names.

profiletype
  The name of the class containing the resource profile.

suffix  Specifies the SDBM suffix.

The suffix for SDBM may contain additional attributes. For example, if the suffix has been specified as:

suffix cn=myRACF,c=US

in the LDAP configuration file, any RACF-style DN would end with:

cn=myRACF,c=US

Following is DN format and a sample DN for a user:

racfid=userid,profiletype=user,suffix
racfid=ID1,profiletype=user,cn=myRACF,c=US

Following is the DN format and a sample DN for a connection:

racuserid=userid+racgroupid=groupid,profiletype=connect,suffix
racuserid=ID1+racgroupid=GRP1,profiletype=connect,cn=myRACF,c=US

Following is the DN format and a sample DN for a resource profile:

profilename=resourcename,profiletype=classname,suffix
profilename=ABC.KEN,profiletype=FACILITY,cn=myRACF,c=US
Chapter 14. LDAP directory schema

The LDAP Version 3 (V3) protocol, as defined in IETF RFC 2252 Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions and IETF RFC 2256 A Summary of the X.500(96) User Schema for use with LDAPv3, describes schema publication and update. Schema publication provides the ability to query the active directory schema through the use of the LDAP search function. Schema update is the ability to change the schema while the directory server is running.

Note:

- The z/OS LDAP server implements both schema publication and update. The schema is stored as an entry in the database and search (publication) and modify (update) operations may be performed on this entry. The distinguished name of the schema entry is `cn=schema`.
  - The `schemaPath` option in the LDAP server configuration file defines the location where the LDAP server will save the schema entry. The default is `/var/ldap/schema`. This directory should be backed up as part of the normal system backup procedure since the loss of the schema directory will invalidate all existing directory entries.
  - When the z/OS LDAP server is first started, the server supplies an initial schema. This initial schema is sufficient for usage of the SDBM (without RACF custom fields), CDBM (with configuration related entries), and GDBM backends, but needs to be updated for usage of LDBM, TDBM, SDBM with RACF custom fields, and CDBM with user defined entries. The initial schema elements cannot be deleted and can only be modified in limited ways. See Appendix A, “Initial LDAP server schema” for the contents of the initial schema.
  - Access to the schema entry is controlled by an access control list (ACL), even if the LDAP server is in maintenance mode. All requests to access the schema entry except those from the LDAP administrator are subject to ACL checking. In particular for a basic replica server, requests from the `masterServerDN` or `peerServerDN` are subject to access control. The default ACL allows all users to display the schema but only the LDAP administrator can update the schema. This ACL can be modified. See Chapter 22, “Using access control” for more information.

Setting up the schema for LDBM, TDBM, and CDBM - new users

The LDAP server is shipped with two predefined schema files representing schema definitions which the user may want to load into the LDAP server schema when using LDBM, TDBM, or CDBM. These files are `schema.user.ldif` and `schema.IBM.ldif` and are located in the `/usr/lpp/ldap/etc` directory. The `schema.IBM.ldif` schema definitions require that the definitions contained in `schema.user.ldif` are loaded prior to loading `schema.IBM.ldif`. Determine which of these schema files would be used to represent the data to be stored in the LDBM, TDBM, or CDBM directory, or locate or create other schema files to use.

Use the `ldapmodify` command to load the schema. For example, the commands to load the `schema.user.ldif` and `schema.IBM.ldif` schema files would be similar to:

```
ldapmodify -h ldaphost -p ldapport -D adminDN -w passwd -f /usr/lpp/ldap/etc/schema.user.ldif
ldapmodify -h ldaphost -p ldapport -D adminDN -w passwd -f /usr/lpp/ldap/etc/schema.IBM.ldif
```

See IBM Tivoli Directory Server Client Programming for z/OS for more information about `ldapmodify`.

Upgrading schema for LDBM, TDBM, and CDBM

The schema files that are shipped with the z/OS LDAP server are based on industry and product defined schemas. As such, they should not be modified since existing products and applications use the schema elements as defined.

If you are migrating TDBM from a z/OS Integrated Security Services LDAP server, see Migrating schema from an ISS LDAP server to z/OS IBM TDS for information concerning the migration of the schema.
Occasionally, schema updates are required during the life of an LDAP release. These updates are applied to and shipped with the `schema.user.ldif` and `schema.IBM.ldif` files found in `/usr/lpp/ldap/etc` directory. When moving to a new release you must reapply both of these files if you previously applied these schema files. Future schema service will depend on those updates being applied to your schema.

If you are using the `schema.user.ldif` and `schema.IBM.ldif` schema files and either the files are updated in the service stream or you are moving to a new release, the LDAP Administrator should update the LDAP server schema through the `ldapmodify` utility. Run the following `ldapmodify` commands to load the schema:

```
ldapmodify -h ldaphost -p ldapport -D adminDN -w passwd -f /usr/lpp/ldap/etc/schema.user.ldif
ldapmodify -h ldaphost -p ldapport -D adminDN -w passwd -f /usr/lpp/ldap/etc/schema.IBM.ldif
```

See [IBM Tivoli Directory Server Client Programming for z/OS](https://www.ibm.com) for more information about `ldapmodify`.

**Notes:**

1. Check that `schemaReplaceByValue off` is not specified in the global section of the LDAP server configuration file or send the `IBMSchemaReplaceByValueControl` control with a value of `TRUE` on the modify request. This control can be sent by specifying the `-u` option on the `ldapmodify` utility. Refer to [Chapter 8, “Customizing the LDAP server configuration”](https://www.ibm.com) for more information about the `schemaReplaceByValue` configuration option and to [Appendix C, “Supported server controls”](https://www.ibm.com) for more information about the `IBMSchemaReplaceByValueControl` control.

2. When the LDAP schema is modified using the `schema.user.ldif` and `schema.IBM.ldif` files, each attribute and object class definition in the file replaces the existing definition in the schema. Any changes previously made in the schema to these attributes and object classes needs to be made again. This includes any changes that are allowed to attributes and object classes in the initial LDAP schema.

### Schema introduction

Entries in the directory are made up of attributes which consist of an attribute type and one or more attribute values. These are referred to as `attribute=value` pairs. Every entry contains one or more `objectclass=value pairs` that identify what type of information the entry contains. The object classes associated with the entry determine the set of attributes which must or may be present in the entry.

The z/OS LDAP server has a single schema for the entire server. This schema is stored as an entry whose distinguished name is `cn=schema`. Following is a portion of the schema entry.

```
cn=SCHEMA
subtreespecification=NULL
objectclass=TOP
objectclass=SUBSCHEMA
objectclass=SUBENTRY
objectclass=IBMSUBSCHEMA
... attributetypes= ( 2.5.4.3 NAME ( 'cn' 'commonName' ) SUP name )
... ibmattributetypes = ( 2.5.4.3 ACCESS-CLASS normal )
... objectclasses = ( 2.5.6.0 NAME 'top' ABSTRACT MUST objectclass )
... ldapsyntaxes = ( 1.3.6.1.4.1.1466.115.121.1.15 DESC 'directory string' )
... matchingrules = ( 2.5.13.5 NAME 'caseExactMatch' SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
...```
The `objectClass` values specified for the schema entry are `top`, `subEntry`, `subSchema`, and `ibmSubschema`. This set of object classes result in the `objectClass`, `cn`, and `subtreeSpecification` attributes being required for a schema entry and the `attributeTypes`, `objectClasses`, `ldapSyntaxes`, `matchingRules`, and `IBMAttributeTypes` attributes being allowed in a schema entry.

**Note:** The `ditContentRules`, `ditStructureRules`, `nameforms`, and `matchingRuleUse` attributes are allowed in a schema entry, but usage of these directives is not implemented by the z/OS LDAP server.

Every entry in the directory including the schema entry contains the `schemasubentry` attribute. The value shown for this attribute is the DN of the schema entry, `cn=schema`. Therefore, a search operation requesting the `schemasubentry` for an entry always returns:

```
schemasubentry=cn=schema
```

Attribute types, object classes, LDAP syntaxes, and matching rules have assigned unique numeric object identifiers. These numeric object identifiers are in dotted decimal format, for example, 2.5.6.6. Attribute types, object classes, and matching rules are also identified by a textual name, for example, `person` or `names`. The numeric object identifier and the textual names may be used interchangeably when an attribute type or object class definition specifies an object identifier. Most schema definitions use the textual name as the object identifier for these definitions.

**Note:** Non-numeric object identifiers, for example `myattr-oid`, can be used instead of numeric object identifiers. However, non-numeric identifiers are not supported in the z/OS Integrated Security Services LDAP server on earlier releases. Do not use non-numeric identifiers if you intend to share a TDBM database with a z/OS Integrated Security Services LDAP server on earlier releases.

The attributes that comprise a directory schema include attribute types, IBM attribute types, object classes, LDAP syntaxes, and matching rules. There is a fixed set of LDAP syntaxes and matching rules supported by the z/OS LDAP server. These are listed in Table 33, Table 34, and Table 35. Each of the schema attributes are described below:

- **Attribute types**
  Attribute types define the characteristics of the data values stored in the directory. Each attribute type defined in a schema must contain a unique numeric object identifier and optionally contain a textual name, zero or more alias names, and a description of the attribute type. The characteristics defined for each attribute type include the syntax, number of values, and matching rules.

  The `SYNTAX` defines the format of the data stored for the attribute type. The server checks the attribute values that are to be added to the directory by comparing the values against the set of allowed characters based on the syntax. For example, if the syntax of an attribute type is Boolean (where the acceptable values are `TRUE` or `FALSE`) and the attribute value specified is `yes`, the update will fail. The syntaxes supported by the z/OS LDAP server are shown in Table 33 and Table 34.

  Matching rules may be specified for `EQUALITY`, `ORDERING`, and `SUBSTR` (substring matching). The matching rule determines how comparisons between values are done. The `EQUALITY` matching rule determines if two values are equal. Examples of `EQUALITY` matching rules are `caseIgnoreMatch`, `caseExactMatch`, and `telephoneNumberMatch`. The `ORDERING` matching rule determines how two values are ordered (`greaterThanOrEqual`, `lessThanOrEqual`). Examples of `ORDERING` matching rules are `caseIgnoreOrderingMatch` and `generalizedTimeOrderingMatch`. The `SUBSTR` matching rule determines if the presented value is a substring of an attribute value from the directory. Examples of `SUBSTR` matching rules are `caseIgnoreSubstringsMatch` and `telephoneNumberSubstringsMatch`.

  If `EQUALITY`, `ORDERING`, or `SUBSTR` matching rules are not specified in the definition of an attribute type or through the inheritance hierarchy, the z/OS LDAP server will perform evaluations to the best of
its ability, but the results may not be as expected. The z/OS LDAP server uses the matching rules shown in the following table based on attribute type syntax to evaluate **EQUALITY**, **ORDERING**, and **SUBSTR** if those matching rules are not specified.

### Table 30. Syntax and default EQUALITY, ORDERING, and SUBSTR matching rules

<table>
<thead>
<tr>
<th>Syntax</th>
<th>EQUALITY</th>
<th>ORDERING</th>
<th>SUBSTR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute Type Description</td>
<td>objectIdentifierFirstComponentMatch</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Binary</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Boolean</td>
<td>booleanMatch</td>
<td>caselgnoreOrderingMatch</td>
<td>caselgnoreSubstringsMatch</td>
</tr>
<tr>
<td>Directory String</td>
<td>caselgnoreMatch</td>
<td>caselgnoreOrderingMatch</td>
<td>caselgnoreSubstringsMatch</td>
</tr>
<tr>
<td>DIT Content Rule Description</td>
<td>objectIdentifierFirstComponentMatch</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>DIT Structure Rule Description</td>
<td>integerFirstComponentMatch</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Distinguished Name</td>
<td>distinguishedNameMatch</td>
<td>distinguishedNameOrderingMatch</td>
<td>-</td>
</tr>
<tr>
<td>Generalized Time</td>
<td>generalizedTimeMatch</td>
<td>generalizedTimeOrderingMatch</td>
<td>-</td>
</tr>
<tr>
<td>IA5 String</td>
<td>caselgnoreIA5Match</td>
<td>caselignoreOrderingMatch</td>
<td>caselgnoreSubstringsMatch</td>
</tr>
<tr>
<td>IBM Attribute Type</td>
<td>objectIdentifierFirstComponentMatch</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>IBM Entry UUID</td>
<td>IBM-EntryUUIDMatch</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Integer</td>
<td>integerMatch</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Matching Rule Description</td>
<td>objectIdentifierFirstComponentMatch</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Matching Rule Use Description</td>
<td>objectIdentifierFirstComponentMatch</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Name Form Description</td>
<td>objectIdentifierFirstComponentMatch</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Object Class Description</td>
<td>objectIdentifierFirstComponentMatch</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Object Identifier</td>
<td>objectIdentifierMatch</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Octet String</td>
<td>octetStringMatch</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Substring Assertion</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Telephone Number</td>
<td>telephoneNumberMatch</td>
<td>-</td>
<td>telephoneNumberSubstringsMatch</td>
</tr>
<tr>
<td>UTC Time</td>
<td>utcTimeMatch</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

The z/OS LDAP server also verifies that the matching rules specified for **EQUALITY**, **ORDERING**, and **SUBSTR** are consistent with the specified **SYNTAX** [Table 31](#) shows acceptable values **EQUALITY**, **ORDERING**, and **SUBSTR**.

### Table 31. Syntax and acceptable matching rules (EQUALITY, ORDERING, and SUBSTR)

<table>
<thead>
<tr>
<th>Syntax</th>
<th>EQUALITY</th>
<th>ORDERING</th>
<th>SUBSTR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute Type Description</td>
<td>objectIdentifierFirstComponentMatch</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Binary</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Boolean</td>
<td>booleanMatch</td>
<td>caselgnoreMatch</td>
<td>caselgnoreSubstringsMatch</td>
</tr>
<tr>
<td>Directory String</td>
<td>caseExactMatch</td>
<td>caselgnoreOrderingMatch</td>
<td>caselgnoreSubstringsMatch</td>
</tr>
<tr>
<td>DIT Content Rule Description</td>
<td>objectIdentifierFirstComponentMatch</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>DIT Structure Rule Description</td>
<td>integerFirstComponentMatch</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Distinguished Name</td>
<td>distinguishedNameMatch</td>
<td>distinguishedNameOrderingMatch</td>
<td>-</td>
</tr>
<tr>
<td>Generalized Time</td>
<td>generalizedTimeMatch</td>
<td>generalizedTimeOrderingMatch</td>
<td>-</td>
</tr>
<tr>
<td>IA5</td>
<td>caselgnoreMatch</td>
<td>caselgnoreOrderingMatch</td>
<td>caselgnoreSubstringsMatch</td>
</tr>
<tr>
<td></td>
<td>caselgnoreIA5Match</td>
<td>caselgnoreOrderingMatch</td>
<td>caselgnoreSubstringsMatch</td>
</tr>
<tr>
<td></td>
<td>caseExactMatch</td>
<td>caselgnoreOrderingMatch</td>
<td>caselgnoreSubstringsMatch</td>
</tr>
<tr>
<td></td>
<td>caseExactIA5Match</td>
<td>caselgnoreOrderingMatch</td>
<td>caselgnoreSubstringsMatch</td>
</tr>
</tbody>
</table>

---

230 z/OS V1R11.0 IBM Tivoli Directory Server Administration and Use for z/OS
Table 31. Syntax and acceptable matching rules (EQUALITY, ORDERING, and SUBSTR) (continued)

<table>
<thead>
<tr>
<th>Syntax</th>
<th>EQUALITY</th>
<th>ORDERING</th>
<th>SUBSTR</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM Attribute Type Description</td>
<td>objectIdentifierFirstComponent Match</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>IBM Entry UUID</td>
<td>IBM-EntryUUIDMatch</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Integer</td>
<td>integerMatch</td>
<td>integerFirstComponentMatch</td>
<td>-</td>
</tr>
<tr>
<td>LDAP Syntax Description</td>
<td>objectIdentifierFirstComponentMatch</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Matching Rule Description</td>
<td>objectIdentifierFirstComponentMatch</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Matching Rule Use Description</td>
<td>objectIdentifierFirstComponentMatch</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Name Form Description</td>
<td>objectIdentifierFirstComponentMatch</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Object Class Description</td>
<td>objectIdentifierFirstComponentMatch</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Object Identifier</td>
<td>objectIdentifierMatch</td>
<td>objectIdentifierFirstComponentMatch</td>
<td>-</td>
</tr>
<tr>
<td>Octet String</td>
<td>octetStringMatch</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Substring Assertion</td>
<td>telephoneNumberMatch</td>
<td>telephoneNumberSubstringsMatch</td>
<td>-</td>
</tr>
<tr>
<td>UTC Time</td>
<td>utcTimeMatch</td>
<td>generalizedTimeOrderingMatch</td>
<td>-</td>
</tr>
</tbody>
</table>

The syntax or matching rule values may be inherited by specifying a superior attribute type. This is done by specifying the keyword `SUP`, followed by the object identifier of the superior attribute type. This is known as an attribute type hierarchy and referred to as inheritance. A superior hierarchy may be created with multiple levels of inheritance. In the following partial example, ePersonName and personName would inherit their `SYNTAX` from name.

ePersonName SUP personName
personName SUP name
name SYNTAX 1.3.6.1.4.1.1466.115.121.1.15

When the `SYNTAX`, `EQUALITY`, `ORDERING`, or `SUBSTR` values are not specified for an attribute type, the attribute type hierarchy are used to determine these values. The `SYNTAX` must be specified on the attribute type or through inheritance.

The number of values that may be stored in each entry for an attribute type is limited to one value if the keyword `SINGLE-VALUE` is specified. Otherwise, any number of attribute values may exist in the entry.

The `OBSOLETE` keyword indicates that the attribute type cannot be used to add data to existing entries or to store data in new entries. Modifications to entries which contain data values of an attribute type which has been made obsolete will fail unless all data values for all obsolete attribute types are removed during the modification. Searches specifying the obsolete attribute type will return the entries containing the attribute type. If an obsolete attribute type is referred to in a superior hierarchy, the inherited values will continue to be resolved.

Example 1:

attributeTypes: ( 1.2.3.4 NAME 'obsattr1' SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 OBSOLETE )
attributeTypes: ( 5.6.7.8 NAME 'validattr1' SUP obsattr1 )

would be the same as

attributeTypes: ( 5.6.7.8 NAME 'validattr' SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

Example 2:

attributeTypes: ( 10.20.30.40 NAME 'obsattr2' SUP obsattr3 )
attributeTypes: ( 50.60.70.80 NAME 'obsattr3' EQUALITY caseIgnoreMatch SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )
attributeTypes: ( 90.100.110.120 NAME 'validattr2' SUP obsattr2 )

would be the same as

attributeTypes: ( 90.100.110.120 NAME 'validattr2' EQUALITY caseIgnoreMatch SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )
The USAGE keyword’s valid values are userApplications or one of three operational values (directoryOperation, distributedOperation, or dSAOperation). An attribute type which has an operational USAGE value is called an operational attribute. Operational attributes are treated differently than non-operational attributes. In particular, the value of an operational attribute type in an entry is only returned by a search operation if the attribute type is specified in the list of attributes to be returned. Also, operational attribute types do not have to belong to an object class. The default for USAGE is userApplications.

The z/OS LDAP server restricts users from modifying data values specified for an attribute type when NO-USER-MODIFICATION is specified on the definition of the attribute type. In general, NO-USER-MODIFICATION should only be specified for attribute types that are set by the server because they cannot be assigned a value by the user. Attribute types which are NO-USER-MODIFICATION can be modified during replication processing and when the LDAP server is in maintenance mode. See Chapter 23, “Basic replication” for more information.

Note: The LDAP V3 protocol also defines a COLLECTIVE key word for attribute types. The LDAP server does not support this key word. All attribute types are assumed to be not COLLECTIVE.

• IBM attribute types

Additional information required by IBM LDAP servers for each attribute type defined in the schema is specified using the IBMAttributeTypes schema attribute. The IBMAttributeTypes schema attribute is an extension of the attributeTypes schema attribute. If the attributeTypes value is not defined, then the corresponding IBMAttributeTypes value cannot be defined. For the z/OS LDAP server, the additional information defined using this attribute is the ACCESS-CLASS and the RACFFIELD of the associated attribute type.

ACCESS-CLASS specifies the level of access users have to data values of this attribute type. The levels that may be specified for user-defined attribute types are normal, sensitive, and critical. The system and restricted keywords are for LDAP server use and are specified for some of the attribute types controlled by the server. See “Attribute access classes” on page 343 for the definition of access classes.

RACFFIELD specifies the information needed to associate this attribute type with a RACF custom field. The presence of RACFFIELD indicates that this attribute type is used to represent a RACF custom field in a user or group profile.

Note: Other LDAP servers from IBM use the DBNAME and LENGTH characteristics to specify additional information for their implementations. These may be specified in the schema but are not used by the z/OS LDAP server.

• Object classes

Object classes define the characteristics of individual directory entries. The object classes listed in a directory entry determine the set of required and optional attributes for the entry. Each object class defined in a schema must contain a unique numeric object identifier and optionally contain a textual name, zero or more alias names, a description of the object class, and lists of required (MUST) or optional (MAY) attribute types.

Required and optional attribute types for an object class may be inherited by specifying one or more superior object classes in an object class definition. This is done by specifying the keyword SUP followed by the object identifiers of the superior object classes. This is known as an object class hierarchy and referred to as multiple inheritance. A superior hierarchy may be created with multiple levels of inheritance.

Each object class is defined as one of three types: STRUCTURAL, ABSTRACT, or AUXILIARY. The type can be specified when the object class is defined. If the type is not specified, it defaults to STRUCTURAL.

The structural object class defines the characteristics of a directory entry. Each entry must specify exactly one base structural object class. A base structural object class is defined as the most
subordinate object class in an object class hierarchy. **The structural object class of an entry cannot be changed.** Once an entry is defined in the directory, it must be deleted and recreated to change the structural object class.

Abstract and auxiliary object classes are used to provide common characteristics to entries with different structural object classes. Abstract object classes are used to derive additional object classes. Abstract object classes must be referred to in a structural or auxiliary superior hierarchy. Auxiliary object classes are used to extend the set of required or optional attribute types of an entry.

When using the keyword `SUP` to create an object class hierarchy, an auxiliary class should only specify superior object classes that are either auxiliary or abstract object classes. Similarly, a structural object class should only specify superior object classes that are either structural or abstract object classes. If these rules are not followed, the z/OS LDAP server might not be able to determine the base structural object class of the entry, resulting in the rejection of the entry.

An example of the relationship between structural, abstract, and auxiliary object classes is the schema entry shown in Figure 7. The schema entry specifies `top`, `subEntry`, `subSchema`, and `ibmSubschema` as object classes. The object classes form the following hierarchy:

```
  top (abstract)
    subEntry (structural)
    SubSchema (auxiliary)
      ibmSubSchema (auxiliary)
```

**Figure 8. Object class hierarchy example**

In this example, the `subEntry` object class is the base structural object class.

The `OBSOLETE` keyword indicates that the object class cannot be used to define entries in the directory. When an object class is made obsolete, new entries specifying the obsoleted object class cannot be added to the directory and existing entries cannot be modified unless the obsolete object class is removed from the entries' object class list. When the obsolete object class is removed from the entry, any attributes in the entry that are associated only with that object class must also be removed. These changes must be made through the same modify operation. If an obsolete object class is specified in a superior hierarchy for a new entry, then attempts to add the entry to the LDAP directory will fail.

- **LDAP syntaxes**
  Each attribute type definition includes the LDAP syntax which applies to the values for the attribute. The LDAP syntax defines the set of characters which are allowed when entering data into the directory.

  The z/OS LDAP server is shipped with predefined supported syntaxes. See Table 33 and Table 34 for the list of syntaxes supported by the z/OS LDAP server. The set of syntaxes cannot be changed, added to, or deleted by users.

- **Matching rules**
Matching rules allow entries to be selected from the database based on the evaluation of the matching rule assertion. Matching rule assertions are propositions which may evaluate to true, false, or undefined concerning the presence of the attribute value or values in an entry.

The z/OS LDAP server is shipped with predefined supported matching rules. See Table 35 for the list of matching rules supported by the z/OS LDAP server. The set of matching rules cannot be changed, added to, obsoleted, or deleted by users.

**Schema attribute syntax**

The attributes which are used in the schema entry use specific character representations in their values. These character representations are described in Table 32. The terms shown in this table are used in the schema attribute definitions in the next section.

Table 32. Character representations

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>noidlen</td>
<td>Represented as:</td>
</tr>
<tr>
<td></td>
<td>numericoid[length]</td>
</tr>
<tr>
<td></td>
<td>where length is a numeric string representing the maximum length of values of this attribute type.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>1.3.6.1.4.1.1466.115.121.1.7{5}</td>
</tr>
<tr>
<td></td>
<td>Implementation note: The z/OS LDAP server allows values to be any length, regardless of the specification of a length in the attribute type definition. User installations that want to limit the length of values need to handle this during data input.</td>
</tr>
<tr>
<td>numericoid</td>
<td>A dotted decimal string.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>2.5.13.72</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> A non-numeric object identifier, for example myattr-oid, can be used instead of a numeric object identifier. However, non-numeric identifiers are not supported in the z/OS Integrated Security Services LDAP server on earlier releases. Do not use non-numeric identifiers if you intend to share a TDBM database with a z/OS Integrated Security Services LDAP server on earlier releases.</td>
</tr>
<tr>
<td>oid</td>
<td>A single object identifier. This may be specified either as a name or as a numeric object identifier.</td>
</tr>
<tr>
<td></td>
<td>Examples:</td>
</tr>
<tr>
<td></td>
<td>name</td>
</tr>
<tr>
<td></td>
<td>2.5.4.41</td>
</tr>
<tr>
<td>oidlist</td>
<td>A list of object identifiers specified as names or numeric object identifiers separated by dollar signs ($) within parentheses.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>( cn $ sn $ postaladdress $ 2.5.4.6 )</td>
</tr>
<tr>
<td>oids</td>
<td>Either an oid or oidlist.</td>
</tr>
</tbody>
</table>
Table 32. Character representations (continued)

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>qdescrs</td>
<td>A quoted description shown as ’descr’ for one and as (’descr’ ’descr’) for more than one. The description (descr) must have an alphabetic character as the first character, followed by any combination of alphabetic or numeric characters, the dash character (-), or the semicolon character (;). Each value must be in single quotation marks (’). If there is more than one value, they must be enclosed in parentheses. Examples: ’x121address’ (’cn’ ’commonName’) ’userCertificate;binary’</td>
</tr>
<tr>
<td>qdstring</td>
<td>A quoted descriptive string shown as ’dstring’. The descriptive string (dstring) is composed of one or more UTF-8 characters. Example: ’This is an example of a quoted descriptive string.’</td>
</tr>
</tbody>
</table>

LDAP schema attributes

The five attributes used to define an LDAP schema are discussed below. For these schema attributes, the numericoid must be the first item in the definition. All other keywords and values may be in any order.

LDAP syntaxes

The set of syntaxes which are supported by the z/OS LDAP server cannot be modified, added to, or deleted by users. The descriptive material included here is for information about.

The format of the LDAP syntaxes attribute in a dynamic schema is:

```
ldapSyntaxes: ( numericoid [DESC qdstring] )
```

- `numericoid`: The unique, assigned numeric object identifier.
- `DESC qdstring`: Text description of the LDAP syntax

**Note:** LDAP syntaxes do not have a textual name. They are identified only by the numeric object identifier.

Following is an example of the definition of an LDAP syntax:

```
ldapSyntaxes: ( 1.3.6.1.4.1.1466.115.121.1.7 DESC 'Boolean' )
```

The LDAP syntaxes supported by the z/OS LDAP server fall into two categories. The first set, as shown in Table 33, would be used when defining attribute types that are used for directory data.

Table 33. Supported LDAP syntaxes - general use

<table>
<thead>
<tr>
<th>Numeric object identifier</th>
<th>Description</th>
<th>Valid values</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.3.6.1.4.1.1466.115.121.1.5</td>
<td>Binary</td>
<td>Binary data</td>
</tr>
<tr>
<td>1.3.6.1.4.1.1466.115.121.1.7</td>
<td>Boolean</td>
<td>TRUE, FALSE</td>
</tr>
<tr>
<td>1.3.6.1.4.1.1466.115.121.1.15</td>
<td>Directory String</td>
<td>UTF-8 characters</td>
</tr>
</tbody>
</table>
Table 33. Supported LDAP syntaxes - general use (continued)

<table>
<thead>
<tr>
<th>Numeric object identifier</th>
<th>Description</th>
<th>Valid values</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.3.6.1.4.1.1466.115.121.1.12</td>
<td>Distinguished Name</td>
<td>Sequence of attribute type and value pairs</td>
</tr>
<tr>
<td>1.3.6.1.4.1.1466.115.121.1.24</td>
<td>Generalized Time</td>
<td>yyyymmddhhmss.ffffff (local time)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>yyyymmddhhmss.ffffffZ (GMT)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>yyyymmddhhmss.ffffff-hhmm (Time zone west)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>yyyymmddhhmss.ffffff+hhmm (Time zone east)</td>
</tr>
<tr>
<td></td>
<td>Note:</td>
<td>The effective time zone for the LDAP server is assumed when calculating GMT from local time.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.3.6.1.4.1.1466.115.121.1.26</td>
<td>IA5 String</td>
<td>IA5 characters (commonly known as 7-bit ASCII)</td>
</tr>
<tr>
<td>1.3.6.1.4.1.1466.115.121.1.27</td>
<td>Integer</td>
<td>+/- 62 digit integer</td>
</tr>
<tr>
<td>1.3.6.1.4.1.1466.115.121.1.38</td>
<td>Object Identifier</td>
<td>Name or numeric object identifier</td>
</tr>
<tr>
<td>1.3.6.1.4.1.1466.115.121.1.40</td>
<td>Octet String</td>
<td>Octet data</td>
</tr>
<tr>
<td>1.3.6.1.4.1.1466.115.121.1.50</td>
<td>Telephone Number</td>
<td>printable string (alphanumeric, decimal, &quot;&quot;, ( ), +, -, ., :, ?, and space)</td>
</tr>
<tr>
<td>1.3.6.1.4.1.1466.115.121.1.53</td>
<td>UTC Time</td>
<td>See Generalized Time above for details</td>
</tr>
</tbody>
</table>

Values defined using the binary and octet string syntaxes are transferred in binary and do not consist of UTF-8 characters.

The second set of syntaxes defined by the z/OS LDAP server are used in the definition of the LDAP schema. These would not typically be used in user schema attribute type definitions. They are listed here for reference.

Table 34. Supported LDAP syntaxes - server use

<table>
<thead>
<tr>
<th>Numeric object identifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.3.6.1.4.1.1466.115.121.1.3</td>
<td>Attribute Type Description</td>
</tr>
<tr>
<td>1.3.6.1.4.1.1466.115.121.1.16</td>
<td>DIT Content Rule Description</td>
</tr>
<tr>
<td>1.3.6.1.4.1.1466.115.121.1.17</td>
<td>DIT Structure Rule Description</td>
</tr>
<tr>
<td>1.3.18.0.2.8.1</td>
<td>IBM Attribute Type Description</td>
</tr>
<tr>
<td>1.3.18.0.2.8.3</td>
<td>IBM Entry UUID Description</td>
</tr>
<tr>
<td>1.3.6.1.4.1.1466.115.121.1.54</td>
<td>LDAP Syntax Description</td>
</tr>
<tr>
<td>1.3.6.1.4.1.1466.115.121.1.30</td>
<td>Matching Rule Description</td>
</tr>
<tr>
<td>1.3.6.1.4.1.1466.115.121.1.31</td>
<td>Matching Rule Use Description</td>
</tr>
<tr>
<td>1.3.6.1.4.1.1466.115.121.1.35</td>
<td>Name Form Description</td>
</tr>
<tr>
<td>1.3.6.1.4.1.1466.115.121.1.37</td>
<td>Object Class Description</td>
</tr>
<tr>
<td>1.3.6.1.4.1.1466.115.121.1.58</td>
<td>Substring Assertion</td>
</tr>
</tbody>
</table>
Matching rules

The set of matching rules which are supported by the z/OS LDAP server cannot be modified, added to, obsoleted, or deleted by users. The descriptive material included here is for information aboutly.

The format of the matching rules attribute in a dynamic schema is:

```
matchingRules: ( numericoid [NAME qdescrs] [DESC qdstring] [OBSOLETE] SYNTAX numericoid )
```

- **numericoid**
  The unique, assigned numeric object identifier.

- **NAME qdescrs**
  The name by which this matching rule is known.

- **DESC qdstring**
  Text description of the matching rule.

- **OBSOLETE**
  Indicates that the matching rule is obsolete.

- **SYNTAX numericoid**
  Specifies the numeric object identifier of the syntax for this matching rule.

Following is an example of the definition of a matching rule:

```
matchingRules: ( 2.5.13.5 NAME 'caseExactMatch' SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

The matching rules supported by the z/OS LDAP server is a fixed set as listed in the following table.

<table>
<thead>
<tr>
<th>Name</th>
<th>Numeric object identifier</th>
<th>Assertion syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>booleanMatch</td>
<td>2.5.13.13</td>
<td>Boolean. Both values are either TRUE or FALSE. Case is ignored.</td>
</tr>
<tr>
<td>caseExactIA5Match</td>
<td>1.3.6.1.4.1.1466.109.114.1</td>
<td>IA5 String. Leading and trailing whitespace is ignored. Embedded whitespace is replaced by a single blank. Case must be the same.</td>
</tr>
<tr>
<td>caseExactMatch</td>
<td>2.5.13.5</td>
<td>Directory String. Leading and trailing whitespace is ignored. Embedded whitespace is replaced by a single blank. Case must be the same.</td>
</tr>
<tr>
<td>caseExactOrderingMatch</td>
<td>2.5.13.6</td>
<td>Directory String. Leading and trailing whitespace is ignored. Embedded whitespace is replaced by a single blank. Case must be the same. Collating sequence is based on the UTF-8 representation.</td>
</tr>
<tr>
<td>caseExactSubstringsMatch</td>
<td>2.5.13.7</td>
<td>Directory String. Leading and trailing whitespace is ignored. Embedded whitespace is replaced by a single blank. Case must be the same.</td>
</tr>
<tr>
<td>caseIgnoreIA5Match</td>
<td>1.3.6.1.4.1.1466.109.114.2</td>
<td>IA5 String. Leading and trailing whitespace is ignored. Embedded whitespace is replaced by a single blank. Case is ignored.</td>
</tr>
<tr>
<td>caseIgnoreMatch</td>
<td>2.5.13.2</td>
<td>Directory String. Leading and trailing whitespace is ignored. Embedded whitespace is replaced by a single blank. Case is ignored.</td>
</tr>
</tbody>
</table>
Table 35. Supported matching rules (continued)

<table>
<thead>
<tr>
<th>Name</th>
<th>Numeric object identifier</th>
<th>Assertion syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>caseIgnoreOrderingMatch</td>
<td>2.5.13.3</td>
<td>Directory String. Leading and trailing whitespace is ignored. Embedded whitespace is replaced by a single blank. Case is ignored. Collating sequence is based on the UTF-8 representation.</td>
</tr>
<tr>
<td>caseIgnoreSubstringsMatch</td>
<td>2.5.13.4</td>
<td>Directory String. Leading and trailing whitespace is ignored. Embedded whitespace is replaced by a single blank. Case is ignored.</td>
</tr>
<tr>
<td>distinguishedNameMatch</td>
<td>2.5.13.1</td>
<td>Distinguished Name. Each name must have the same number of RDN components and each attribute within each RDN must match using the EQUALITY rule for that attribute type.</td>
</tr>
<tr>
<td>distinguishedNameOrderingMatch</td>
<td>1.3.18.0.2.4.405</td>
<td>Distinguished Name. The normalized string representation of each name is compared. The collating sequence is based on the UTF-8 representation.</td>
</tr>
<tr>
<td>generalizedTimeMatch</td>
<td>2.5.13.27</td>
<td>Generalized Time. The value will be normalized as yyyyymmddhhmmss.ffffffZ.</td>
</tr>
<tr>
<td>generalizedTimeOrderingMatch</td>
<td>2.5.13.28</td>
<td>Generalized Time. The value will be normalized as yyyyymmddhhmmss.ffffffZ.</td>
</tr>
<tr>
<td>IBM-EntryUUIDMatch</td>
<td>1.3.18.0.2.22.2</td>
<td>IBM Entry UUID. Hyphens are removed and a case-insensitive string comparison is performed.</td>
</tr>
<tr>
<td>integerFirstComponentMatch</td>
<td>2.5.13.29</td>
<td>Integer.</td>
</tr>
<tr>
<td>integerMatch</td>
<td>2.5.13.14</td>
<td>Integer.</td>
</tr>
<tr>
<td>objectIdentifierMatch</td>
<td>2.5.13.0</td>
<td>Object Identifier. The value will be normalized as an attribute descriptor.</td>
</tr>
<tr>
<td>objectIdentifierFirstComponentMatch</td>
<td>2.5.13.30</td>
<td>Object Identifier. The value will be normalized as an attribute descriptor.</td>
</tr>
<tr>
<td>octetStringMatch</td>
<td>2.5.13.17</td>
<td>Octet String. Both values must contain the same number of octets and each octet must have the same value.</td>
</tr>
<tr>
<td>telephoneNumberMatch</td>
<td>2.5.13.20</td>
<td>Telephone Number</td>
</tr>
<tr>
<td>telephoneNumberSubstringsMatch</td>
<td>2.5.13.21</td>
<td>Telephone Number. The value will be normalized using the telephoneNumberMatch rule.</td>
</tr>
<tr>
<td>utcTimeMatch</td>
<td>2.5.13.25</td>
<td>UTC Time. The value will be normalized as yyyyymmddhhmmss.ffffffZ.</td>
</tr>
</tbody>
</table>

Notes on matching rules:
1. An undefined attribute type within a distinguished name uses the directory string matching rules.
2. The aclEntry and entryOwner attribute types use the distinguished name matching rules. The assertion value is just the DN portion of the attribute value.
3. Attribute types with a binary transfer syntax cannot be used in a search filter but can be used in a compare operation.

4. The ibm-allGroups and ibm-allMembers attribute types cannot be used in a search filter. These are read-only operational attributes and will result in a FALSE match status when used in a search filter.

5. The TDBM, LDBM, and CDBM backends ignore the ORDERING and SUBSTR matching rules and always uses the EQUALITY matching rule when processing a search filter.

Attribute types
The format of the attribute types attribute in a dynamic schema is:

```
attributeTypes: ( numericoid [ NAME qdescrs ] [ DESC qdstring ] [ OBSOLETE ] [ SUP oid ]
[ EQUALITY oid ] [ ORDERING oid ] [ SUBSTR oid ] [ SYNTAX noidlen ] [ SINGLE-VALUE ]
[ NO-USER-MODIFICATION ] [ USAGE attributeUsage ] )
```

<table>
<thead>
<tr>
<th>numericoid</th>
<th>The unique, assigned numeric object identifier.</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME qdescrs</td>
<td>The name and alias names by which this attribute type is known. This is also known as the object identifier. The first name in the list is used as the base name and the other names are referred to as alias names. It is suggested the shortest name be listed first. If a name is not specified, the numeric object identifier is used to refer to the attribute type.</td>
</tr>
<tr>
<td>DESC qdstring</td>
<td>Text description of the attribute type.</td>
</tr>
<tr>
<td>OBSOLETE</td>
<td>Indicates that the attribute type is obsolete.</td>
</tr>
<tr>
<td>SUP oid</td>
<td>Specifies the superior attribute type. When a superior attribute type is defined, the EQUALITY, ORDERING, SUBSTR, and SYNTAX values may be inherited from the superior attribute type. The referenced superior attribute type must also be defined in the schema. When the SYNTAX, EQUALITY, ORDERING, or SUBSTR values are not specified for an attribute type, the attribute type hierarchy is used to determine these values. The SYNTAX must be specified on the attribute type or through inheritance.</td>
</tr>
<tr>
<td>EQUALITY oid</td>
<td>Specifies the object identifier of the matching rule which is used to determine the equality of values.</td>
</tr>
<tr>
<td>ORDERING oid</td>
<td>Specifies the object identifier of the matching rule which is used to determine the order of values.</td>
</tr>
<tr>
<td>SUBSTR oid</td>
<td>Specifies the object identifier of the matching rule which is used to determine substring matches of values.</td>
</tr>
<tr>
<td>SYNTAX noidlen</td>
<td>The syntax defines the format of the data stored for this attribute type. It is specified using the numeric object identifier of the LDAP syntax and, optionally, the maximum length of data stored for this attribute type.</td>
</tr>
</tbody>
</table>

Implementation note: The z/OS LDAP server allows values to be any length, regardless of the specification of a length in the attribute type definition. User installations that want to manage the lengths of values need to handle this when values are put into the directory.

| SINGLE-VALUE         | Limits entries to only one value for this attribute type. |
| NO-USER-MODIFICATION | When specified, users may not modify values of this attribute type. |
**USAGE** attributeUsage

Specify **userApplications** for *attributeUsage*. If **USAGE** is not specified, the default is *userApplications*.

The **directoryOperation**, **distributedOperation**, and **DSAOperation** keywords are used to create operational attributes. Operational attributes are treated differently than non-operational attributes. In particular, the value of an operational attribute type in an entry is only returned by a search operation if the attribute type is specified in the list of attributes to be returned. Also, operational attribute types do not have to belong to an object class.

Following are examples of the definition of attribute types:

```plaintext
attributeTypes: ( 2.5.4.6 NAME 'c' SUP name SINGLE-VALUE )
attributeTypes: ( 2.5.4.41 NAME 'name' EQUALITY caseIgnoreMatch SUBSTR
caseIgnoreSubstringsMatch SYNTAX 1.3.6.1.4.1.1466.115.121.1.15[32768] )
```

**IBM attribute types**

The format of the IBM attribute types attribute in a dynamic schema is:

```plaintext
IBMAttributeTypes: ( numericoid [ACCESS-CLASS ibmAcessClass] [RACFFIELD qdescrs] )
```

**numericoid**

The unique, assigned numeric object identifier of the associated attribute type.

**ACCESS-CLASS ibmAcessClass**

The level of sensitivity of the data values for this attribute type. The acceptable values are **normal**, **sensitive**, and **critical**. See "Attribute access classes" on page 343 for the definition of these values. The default attribute access class for an attribute is **normal**.

**RACFFIELD qdescrs**

The information needed to associate this attribute with a RACF custom field in a user or group profile.

The format of *qdescrs* is either a value in single quotation marks:

```plaintext
RACFFIELD 'racfFieldName'
```

or two values in parentheses, each in single quotation marks and separated by a blank:

```plaintext
RACFFIELD ('racfFieldName' 'racfFieldType')
```

*racfFieldName* format must be:

```plaintext
USER-CSDATA-name
```

or

```plaintext
GROUP-CSDATA-name
```

where *name* is the name of the associated RACF custom field.

*racfFieldType* is the type of custom field. The acceptable values are:

- **char**, **flag**, **hex**, **num**, and **qchar**

This value defaults to **char** if the *racfFieldType* is not specified.

**Note:** When specifying **RACFFIELD**:

1. The syntax of the attribute type must be IA5 String (1.3.6.1.4.1.1466.115.121.1.26).
2. The same *racfFieldName* value cannot be specified in more than one IBMAttributeTypes schema definition.
3. The *racfFieldType* value can affect the processing of the attribute values.
4. See "Associating LDAP attributes to RACF custom fields" for more information about the LDAP server support for RACF custom fields.
The `IBMAttributeTypes` schema element is an extension of the `attributeTypes` schema element. If the `attributeTypes` value is not defined, then the corresponding `IBMAttributeTypes` value cannot be defined.

Some schema elements are shipped with `ACCESS-CLASS` set to `restricted` or `system`. These values are used by the LDAP server. Other IBM LDAP servers may also specify `DBNAME`, `LENGTH`, and other keywords and values. These keywords are not used by the z/OS LDAP server and do not need to be specified when creating schemas. If they are specified in a schema used by the z/OS LDAP server, they are ignored.

Following is an example of the definition of an IBM attribute type:

```
IBMAttributeTypes: (2.5.4.6 ACCESS-CLASS normal)
```

**Object classes**

The format of the object classes attribute in a dynamic schema is:

```
objectClasses: ( numericoid [NAME qdescrs] [DESC qdstring]
[OBSOLETE] [SUP oids] [ABSTRACT|STRUCTURAL|AUXILIARY] [MUST oids] [MAY oids] )
```

**numericoid**
The unique, assigned numeric object identifier.

**NAME qdescrs**
The name and alias names by which this object class is known. This is also known as the object identifier. The first name in the list is used as the base name. If name is not specified, the numeric object identifier is used to refer to the object class.

**DESC qdstring**
Text description of the object class.

**OBSOLETE**
Indicates that the object class is obsolete.

**SUP oids**
List of one or more superior object classes. When a superior object class is defined, entries specifying the object class must adhere to the superset of `MUST` and `MAY` values. The superset of `MUST` and `MAY` values include all `MUST` and `MAY` values specified in the object class definition and all `MUST` and `MAY` values specified in the object class’s superior hierarchy. When an attribute type is specified as a `MUST` in an object class in the hierarchy and a `MAY` in another object class in the hierarchy, the attribute type is treated as a `MUST`. Referenced superior object classes must be defined in the schema.

**ABSTRACT | STRUCTURAL | AUXILIARY**
Indicates the type of object class. `STRUCTURAL` is the default.

**MUST oids**
List of one or more mandatory attribute types. Attribute types which are mandatory must be specified when adding or modifying a directory entry.

**MAY oids**
List of one or more optional attribute types. Attribute types which are optional may be specified when adding or modifying a directory entry.

The `extensibleObject` object class is an `AUXILIARY` object class which allows an entry to optionally hold any attribute type. The `extensibleObject` object class is supported by the z/OS LDAP server. This allows any attribute type that is known by the schema to be specified in an entry which includes `extensibleObject` in its list of object classes.

The `top` object class is an abstract object class used as a superclass of all structural object classes. For each structural object class, `top` must appear in the `SUP` list of this object class or of an object class in the superior hierarchy of this object class.
Following is an example of the definition of an object class:

```ldif
objectClasses: ( 2.5.6.0 NAME 'top' ABSTRACT MUST objectclass )
objectClasses: ( 2.5.6.6 NAME 'person' SUP top STRUCTURAL MUST ( cn $ sn )
            MAY ( userpassword $ telephonenumber $ seealso $ description ) )
objectClasses: ( 5.6.7.8 NAME 'company' SUP top MUST ( department $ telephoneNumber ) MAY ( postalAddress $ street ) )
objectClasses: ( 1.2.3.4 NAME 'companyPerson' SUP ( company $ person ) )
```

### Defining new schema elements

You can define new schema elements for use by applications that you develop to use the directory. You can add new object classes and attribute types to the schema. To define a new object class or attribute type, create an LDIF file containing the new schema information, and perform an LDAP modify operation on the schema entry. Object classes and attribute types must be defined using the formats described in the previous section, and must include unique numeric object identifiers and names. Ensuring that the numeric object identifier and names are unique is essential to the correct operation of the directory when using your newly defined schema elements.

Numeric object identifiers (OIDs) are strings of numbers, separated by periods. OID “ranges” or “arcs” are allocated by naming authorities. If you are going to define new schema elements, you should obtain an “OID arc” from a naming authority. One such location to get an “OID arc” assigned is managed by Internet Assigned Numbers Authority (IANA) and, can be found at:

[http://www.iana.org](http://www.iana.org)

Select the “Application Forms” link and then the “Private Enterprise Number” link to apply for a Private Enterprise number.

Once you have obtained an “OID arc” you can begin assigning OIDs to object classes and attribute types that you define.

For the example below, assume that we have been assigned OID arc 1.3.18.0.2.1000.100. (Note: Do not use this OID arc for defining your own schema elements. This arc is assigned to IBM for its use.) The following example adds a new object class that refers to two new attribute types. As you can see, the object class and attribute types can be added to the schema using a single LDAP modify operation. The changes to the schema are represented in LDIF mode input below:

```ldif
dn: cn=schema
changetype: modify
add: attributetypes
attributetypes: ( 1.3.18.0.2.1000.100.4.1 NAME 'YourCompanyDeptNo'
                DESC 'A users department number.'
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 EQUALITY caseIgnoreMatch
                USAGE userApplications
            )
ibmattributetypes: ( 1.3.18.0.2.1000.100.4.1 ACCESS-CLASS normal )
attributetypes: ( 1.3.18.0.2.1000.100.4.2 NAME 'YourCompanyEmployeeID'
                DESC 'A user employee ID.'
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 EQUALITY caseIgnoreMatch
                USAGE userApplications
            )
ibmattributetypes: ( 1.3.18.0.2.1000.100.4.2 ACCESS-CLASS sensitive )
- add: objectclasses
objectclasses: ( 1.3.18.0.2.1000.100.6.1 NAME 'YourCompanyPerson'
                DESC 'Attached to inetOrgPerson to add more attributes.'
                SUP top
                AUXILIARY
                MAY ( YourCompanyDeptNo $ YourCompanyEmployeeID )
            )
-```

This short description has described how to update the schema with new schema elements. Defining new schema elements is a complex undertaking and requires a thorough understanding of schema.
Updating the schema

Attention

Updating the schema, if not done properly, can result in being unable to access data. Read this section thoroughly to avoid this situation.

When the z/OS LDAP server is first started, the server supplies an initial schema. This initial schema is sufficient for usage of the SDBM (without RACF custom fields), CDBM (with configuration related entries), and GDBM backends, but needs to be updated for usage of LDBM, TDBM, SDBM with RACF custom fields, and CDBM with user defined entries. The schema files shipped with the LDAP server, schema.user.ldif and schema.IBM.ldif, may be sufficient for your usage of LDBM, CDBM, or TDBM. See Setting up the schema for LDBM, TDBM, and CDBM - new users for more information about adding these files to the schema. If they are not sufficient, you can change the schema as needed. The schema entry is required and cannot be deleted. When in a sysplex, changes to the schema made at one LDAP server are broadcast to all the LDAP servers in the group. When deleting an attribute type or object class definition, you need to provide just the object identifier enclosed in parentheses. Any additional fields that are specified are checked for proper syntax but are not used.

The operations supported include adding, modifying, or deleting any object class, attribute type, or IBM attribute type that is not part of the initial schema definition required by the LDAP server. Changes to the initial schema are very restricted. See Changing the initial schema for more information. The modifications (additions, changes, and deletions) specified by the LDAP modify function are applied to the schema entry. The resulting schema entry becomes the active schema and is used by all backends to verify that directory changes adhere to it.

Updates to the schema must be performed such that the schema fully resolves. This includes:

- All attribute types referred to in object classes must exist in the schema.
- All superior attribute types or object classes must exist.
- Only the syntaxes and matching rules supported by the schema may be specified in attribute type definitions.
- All attribute types referred to in IBM attribute type definitions must also be defined as attribute types.
- All structural object classes must include the top object class in their object class hierarchy.

Modifications to the schema are rejected if they would possibly make existing entries no longer valid. If there is an entry in a TDBM, CDBM, or LDBM backend that is using an attribute or object class:

- The attribute or object class cannot be deleted. Instead, "delete" the schema element by modifying it to mark it as OBSOLETE rather than deleting its definition from the schema entry. Therefore, no new entries can be created using the schema element and the existing entries which do use the schema element are still accessible. An existing entry that uses the OBSOLETE schema element must be modified to use only non-OBSOLETE schema elements during the next modification of the entry in order for the modification to succeed.
- The attribute or object class cannot be modified in a way that could affect the data in the entry. For example, the syntax of an attribute cannot be changed when that attribute is in use. You must modify the entries first so they do not use the object class or attribute, then change the schema.

The following fields in an attribute type definition are the only fields that can be modified if the attribute type is in use by an entry:

- DESC
- OBSOLETE
- SINGLE-VALUE (can be removed but not added)
- NO-USER-MODIFICATION
- USAGE
The following fields in an IBM attribute type definition can be modified:

- ACCESS-CLASS
- RACFFIELD

The following fields in an object class definition can be modified when the object class is in use by an entry:

- DESC
- OBSOLETE
- MUST (can only move an attribute to MAY)
- MAY (can only add an attribute)

## Changing the initial schema

The initial schema contains the `ldapSyntaxes`, `matchingRules`, `attributeTypes`, `IBMAttributeTypes`, and `objectClasses` needed by the LDAP server. See Appendix A, “Initial LDAP server schema” for the contents of the initial schema.

The syntaxes, matching rules, attribute types, and IBM attribute types in the initial schema cannot be deleted or modified. The object classes in the initial schema cannot be deleted or modified, with the following exceptions:

1. `groupOfNames`
2. `groupOfUniqueNames`

These object classes allow the following fields to be modified:

- DESC
- MUST
- MAY

The `MUST` and `MAY` lists can be modified in any way if no directory entries are using this object class. If there is a directory entry using this object class, the only `MUST` and `MAY` changes allowed are to move an attribute from the `MUST` list to the `MAY` list and to add an attribute to the `MAY` list.

Any part of a schema modification that attempts to add LDAP syntaxes or matching rules to the schema or to modify the initial schema except as described above is ignored, with no message issued to indicate this. The rest of the schema modification is performed and the result of those changes is returned to the client.

## Replacing individual schema values

It is often necessary to apply an updated schema file to an existing schema. Optimally, this would replace changed values in the existing schema with their updated values from the file and add new values from the file to the existing schema, leaving all other values in the existing schema unchanged. However, this is not the way the RFC 2251 definition for such a modify with replace operation works: the RFC requires that ALL the existing values in the schema be replaced by the values specified in the schema file. Therefore, the schema file would have to contain all the unchanged values from the schema in addition to the updated and new values so that no unchanged existing values are lost.

To address this problem, the LDAP server supports two different behaviors when using a modify with replace operation on the schema entry:

1. Standard RFC behavior, in which all the existing values for an attribute are replaced by the ones specified in the modify operation. In order for the modification to succeed, the replacement values must include definitions for all schema definitions that are in use by existing directory entries and the replacement values must conform to the rules described above about what fields can be modified in an active schema entry.
2. Schema-replace-by-value behavior, in which each replace value in the modify operation either replaces the existing value (if one exists) in the schema or is added to the schema (if an existing value does not exist). All other values in the schema remain as they are. A replace value replaces a schema value if the schema value and replace value have the same numeric object identifier (NOID). Otherwise, the replace value is considered a new value and is added to the existing values in the schema.

In all cases, the values of the attribute that are in the initial LDAP server schema cannot be deleted and can only be modified in limited ways as described in Changing the initial schema.

The behavior used by the LDAP server is selected in one of two ways:

1. Specify the `schemaReplaceByValue` option in the global section of the LDAP server configuration file to set the behavior for all modify with replace operations of the schema. Specifying `on` activates the schema-replace-by-value behavior; `off` activates the standard RFC behavior. Refer to Chapter 8, "Customizing the LDAP server configuration" for more information.

2. Specify the `IBMschemaReplaceByValueControl` control on the modify with replace operation to set the behavior for just that specific modify operation, overriding the `schemaReplaceByValue` configuration option. Specifying `TRUE` in the control activates the schema-replace-by-value behavior; `FALSE` activates the standard RFC behavior. Refer to Appendix C, "Supported server controls" for more information.

If neither the `schemaReplaceByValue` configuration option nor the `IBMschemaReplaceByValueControl` control is specified, the default behavior is schema-replace-by-value.

Example: assume that the objectclasses attribute for `cn=schema` contains the following values:

```
objectclasses: ( 1.130.255 NAME 'oldObjectclass1' DESC 'old description 1' ... )
objectclasses: ( 1.130.256 NAME 'oldObjectclass2' DESC 'old description 2' ... )
objectclasses: ( 1.130.257 NAME 'oldObjectclass3' DESC 'old description 3' ... )
```

We would like to replace 'oldObjectclass1' and add a value for 'newObjectclass4'.

This is the update file for the modify operation:

```
dn: cn=schema
changetype: modify
replace: objectclasses
objectclasses: ( 1.130.255 NAME 'newObjectClass1' DESC 'new description 1' ... )
objectclasses: ( 1.3.5.9 NAME 'newObjectClass4' DESC 'description 4' ... )
```

After the modify operation with schema-replace-by-value behavior, the objectclasses attribute in the schema would have the following values:

```
objectclasses: ( 1.130.255 NAME 'newObjectClass1' DESC 'new description 1' ... )
objectclasses: ( 1.130.256 NAME 'oldObjectclass2' DESC 'old description 2' ... )
objectclasses: ( 1.130.257 NAME 'oldObjectclass3' DESC 'old description 3' ... )
objectclasses: ( 1.3.5.9 NAME 'newObjectClass4' DESC 'description 4' ... )
```

If the modify operation with traditional RFC behavior is performed instead, the objectclasses attribute in the schema would end up with the following values:

```
objectclasses: ( 1.130.255 NAME 'newObjectClass1' DESC 'new description 1' ... )
objectclasses: ( 1.3.5.9 NAME 'newObjectClass4' DESC 'description 4' ... )
```

IBM attribute types are extensions to the attribute type definition. The IBM attribute type is deleted when the corresponding attribute type is deleted. IBM attribute types are always replaced by value even when `schemaReplaceByValue off` is specified in the LDAP server configuration file. This ensures that access class protection isn't inadvertently removed from an existing attribute type.
Updating a numeric object identifier (NOID)

It may become necessary to update the numeric object identifier (NOID) of an attribute type or object class in the schema. This NOID change can be accomplished by a special modify operation. The modify operation must consist only of a value to delete followed by a value to add. The value to delete must specify the current NOID of the attribute type or object class whose NOID is to be changed; the value to add must specify the new NOID for the attribute type or object class, along with all the other parts of the attribute type or object class definition. For an attribute type, the NAME, SUP, EQUALITY, ORDERING, SUBSTR, and SYNTAX must be identical in the existing definition and the value to add. SINGLE-VALUE can be removed but not added. For an object class, NAME, SUP, MUST, MAY, and type (ABSTRACT, STRUCTURAL, or AUXILIARY) must be identical in the existing definition and the value to add. The entire attribute type or object class definition is replaced by the contents of the add. Note that the object identifier assigned to an attribute type or object class cannot be changed if there are any directory entries using the attribute type or object class. Also, the object identifier of an attribute type or object class in the initial LDAP schema cannot be changed.

Example: suppose we want to change the NOID of the xyz attribute type from 1.3.5.7 to 2.4.6.8. The update file for the modify operation to accomplish this would look like:

```
cn=schema
-attributetypes=( 1.3.5.7 NAME 'xyz' DESC 'xyz attribute added for application abc' \ 
SYNTAX 1.3.6.1.4.1.1466.115.121.1.5 USAGE userApplications )
+attributetypes=( 2.4.6.8 NAME 'xyz' DESC 'xyz attribute added for application abc' \ 
SYNTAX 1.3.6.1.4.1.1466.115.121.1.5 USAGE userApplications )
```

Changing a NOID should not need to be done as part of normal LDAP server operations. It is intended to be used as an error recovery device for when an incorrect NOID has been added to the schema.

Analyzing schema errors

Following is some information about the possible cause of some schema errors that may be encountered when updating schema:

- For enhanced readability, type:value pairs in LDIF files may be split across multiple lines. The indicator to LDIF that the subsequent lines are continuations is that the first character on the subsequent line is a space. This character is ignored by parsers and it is assumed that the next character immediately follows the previous line. Therefore, if a space is needed between the last value on one line and the first value on the subsequent line, a second space needs to exist on the subsequent LDIF line. Various reason codes related to unrecognized values may be issued.
- Only limited changes are allowed to the initial schema, as described in Changing the initial schema. All other changes to the initial schema are ignored by the LDAP server with no error returned.
- The IBM attribute type schema attribute is an extension to the associated attribute type in the schema. If the schema update contains an IBM attribute type value for which an attribute type value is not defined, the schema update will fail. For example,

  IBMAttributeTypes: ( 1.2.3.4 ACCESS-CLASS normal )

  cannot be specified unless

  attributeTypes: ( 1.2.3.4 NAME 'sample' ... )

  is also defined.
- While the UTC Time syntax is supported, usage of the Generalized Time syntax is recommended. For UTC Time syntax, year values between 70 and 99 assume 1970 to 1999 and values between 00 and 69 assume 2000 to 2069.
- When searching attribute type values of GMT or UTC Time syntax, use GMT syntax in the search filter rather than local time. All time values are stored in the data store as GMT times.
Retrieving the schema

The following sections describe how you can display the schema entry and also find the subschemaSubentry DN.

Displaying the schema entry

The following command shows how to search for the schema entry. Note that the scope must be base in the search request to display the schema.

```
ldapsearch -h ldaphost -p ldapport -s base -b "cn=schema" "objectclass=subschema"
```

Immediately after the server is started for the first time, this command produces the results shown in Appendix A, “Initial LDAP server schema.” After the schema has been updated by the administrator, the search results will show the full schema as the union of the initial schema and the added schema elements.

The search results will contain these attributes:

- `cn=SCHEMA`
- `cn=schema`
- `subtreespecification=NULL`
- `objectclass=TOP`
- `objectclass=SUBSCHEMA`
- `objectclass=SUBENTRY`
- `objectclass=IBMSUBSCHEMA`

```
... attributetypes = ( 2.5.4.3 NAME ( 'cn' 'commonName' ) SUP name )
... ibmattributetypes = ( 2.5.4.3 ACCESS-CLASS normal )
... objectclasses = ( 2.5.6.0 NAME 'top' ABSTRACT MUST objectclass )
... ldapsyntaxes = ( 1.3.6.1.4.1.1466.115.121.1.15 DESC 'directory string' )
... matchingrules = ( 2.5.13.5 NAME 'caseExactMatch' SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
...```

Finding the subschemaSubentry DN

The subschemaSubentry attribute in each directory entry contains the DN of the LDAP server schema entry. To find the value of the subschemaSubentry attribute, specify subschemaSubentry as an attribute to be returned on an LDAP search of the entry.

```
ldapsearch -h ldaphost -p ldapport -s base -b "o=Acme Company, c=UK" "objectclass=*"
   subschemasubentry
```

```
o=Acme Company, c=UK
subschemasubentry=cn=schema
```
Chapter 15. Modify DN operations

The Modify DN Operation allows a client to change the leftmost (least significant) component of the name of an entry in the directory, or to move a subtree of entries to a new location in the directory. This topic explains the function of the Modify DN operation and the options supported to influence the scope and duration of the operation. In addition, it instructs on the techniques necessary to achieve certain forms of directory renames and movement, and it advises on issues which may result in unintentional or unwanted results.

In LDAP, modify DN operations are only supported in the TDBM (DB2-based), LDBM (file-based), and CDBM (file-based) backends.

Modify DN operation syntax

The z/OS implementation of the Modify DN operation supports all required and optional parameters described for the operation in RFC 2251. Specifically, these parameters are required:

- **entryDN**: This is the Distinguished Name (DN) of the entry whose name will be changed. This entry may or may not have subordinate entries. This parameter may not be a zero-length string.

- **newRdn**: The Relative Distinguished Name (RDN) that will form the leftmost component of the new name of the entry. This parameter may not be a zero-length string. If the intent of the Modify DN operation is to move the target entry to a new superior without changing its RDN, the old RDN value must be supplied in the `newRdn` parameter. The attributes and values in the `newRdn` parameter are added to the entry if they are not already present in the entry.

- **deleteoldrdn**: A boolean parameter that controls whether the old RDN attribute values are to be retained attributes of the entry or whether they will be deleted from the entry.

The following parameter to the Modify DN operation is optional:

- **newSuperior**: The Distinguished Name (DN) of the entry which will become the immediate superior of the renamed entry (identified by the `entryDN` parameter). If this parameter is present, it may consist of a zero-length string or a non-zero-length string. See [Modify DN operations related to suffix DNs](#) for more information on the use of a zero-length string for this parameter. A zero-length string value for this parameter (""") will signify that the new superior entry is the root DN.

This operation also supports **optional values, or controls, to influence the behavior of the operation. Two controls are supported (see [Appendix C, “Supported server controls”](#))**: 

- **IBMModifyDNTimelimitControl**: This control causes the Modify DN operation to be abandoned if its duration exceeds the time limit represented by the control value expressed in seconds. No changes are made if the operation is abandoned. This control is honored even if it is set by the admin DN for the server. When this control is present, it will not be propagated to the replica servers. (See [Modify DN operations and replication](#) for more information about replication of Modify DN operations.)

- **IBMModifyDNRealignDNAttributesControl**: This control causes the server to search for all attributes whose attribute type is based on a DN syntax (designated by OID 1.3.6.1.4.1.1466.115.121.1.12) and whose values match any of the old DN values being renamed as part of the Modify DN operation, and to modify the old DN values to reflect the corresponding renamed DN attribute values. This includes modifications to two other attribute types which have constructed DN-type attribute values (those whose attribute syntax is not distinguished name but which may be used to store DN values). They are `aciEntry` and ownership `entryOwner` attributes. Updates to constructed DN types will be limited to these two attributes defined by the LDAP Server. No changes will be made to any user constructed types.

This control is an all-or-none operation in which the server attempts to realign all appropriately-matched DN attribute values in the LDBM, TDBM, or CDBM backend. Users cannot limit the scope of values which should be realigned. If a failure arises during the realignment operation, it realigns none of the values, and the Modify DN operation fails. No changes are made if the operation is abandoned. It
should be noted that even if the control is designated as non-critical, the server will still try to honor the intent of the control and if this attempt fails, the entire Modify DN operation will fail.

When IBMModifyDNRealignDNAttributesControl is present on a request to a master server on which replication of Modify DN operations is enabled, it will be propagated to the replica servers. (See Modify DN operations and replication for more information about replication of Modify DN operations.)

A few simple examples of the use of the Modify DN operation follow. Each request will be expressed in the format of the ModifyDNRequest defined in RFC 2251, as well as in the corresponding invocation command for the z/OS client utility program ldapmodrdn. Refer to the IBM Tivoli Directory Server Client Programming for z/OS for more information on the ldapmodrdn utility.

Example 1: Simple Modify DN of leaf node

```
ModifyDNRequest ::= {
  entry cn=Kevin Heard, o=Athletics, o=Human Resources, o=Deltawing, c=AU
  newrdn   cn=Kevin T. Heard
  deleteoldrdn TRUE
}
```

```
ldapmodrdn -h ldaphost -p ldapport -D binddn -w passwd -r "cn=Kevin Heard, o=Athletics, o=Human Resources, o=Deltawing, c=AU" "cn=Kevin T. Heard"
```

Figure 9. Before Modify DN operation
Note: The -r parameter specifies that the old RDN attribute value (cn=Kevin Heard) will be deleted from the target entry after this operation.

Example 2: Simple Modify DN of non-leaf node

```
ModifyDNRequest ::= {
  entry o=Athletics, o=Human Resources, o=Deltawing, c=AU
  newrdn ou=College Athletics Dept.,
  deleteoldrdn FALSE
}
```

```
ldapmodrdn -h ldaphost - p ldapport -D binddn -w passwd "o=Athletics, o=Human Resources, o=Deltawing, c=AU" * "ou=College Athletics Dept."
```
Note: The absence of the -r parameter specifies that the old RDN attribute value (o=Athletics) will be preserved in the target entry after this operation.

Example 3: Modify DN of non-leaf node with relocation (newSuperior)

ModifyDNRequest ::= {
  entry          o=Athletics, o=Human Resources, o=Deltawing, c=AU
  newrdn         o=Adult Athletics
  deleteoldrdn   FALSE,
  newSuperior    ou=Sport, ou=Vision On Demand, o=Deltawing, o=AU
}

ldapmodrdn -h ldaphost -p ldapport -D binddn -w passwd -s "ou=Sport, ou=Vision On Demand, o=Deltawing, c=AU" "o=Athletics,o=Human Resources, o=Deltawing, c=AU" "o=Adult Athletics"
Note: The absence of the -r parameter specifies that the old RDN attribute value (o=Athletics) will be preserved in the target entry after this operation. The target entry and descendants in its subtree will be relocated in the directory hierarchy.

Considerations in the use of Modify DN operations

As this operation has the potential to significantly change directory data and how it can be accessed, it is important that the user fully understand the data before using the Modify DN operation. Specifically, the user needs to know that:

- The ability of this operation to move directory subtrees has the potential for affecting many entries in the directory in a single operation.
• Certain options may result in modification of additional directory entries which are outside the scope of the directory subtrees being moved. This topic will explain and give examples of how that can occur.
• Because the changes performed to the directory as a result of the operation are committed as a single transaction (or reversed if an error occurs), it may result in a long-running transaction, which may reduce concurrency of other LDAP operations targeted for the same directory entries. See Concurrency considerations between Modify DN operations and other LDAP operations for more information.
• The scope of the changes may result in unanticipated effects in the directory and may affect user access to these entries. See Access control changes for more information.
• There are limitations to which directory entries are eligible for the Modify DN operation. See Eligibility of entries for rename for more information.
• In case the directory needs to be returned to a state before a Modify DN operation, the directory should be backed up by using the ds2ldif utility program or, for a TDBM backend, by using DB2 utilities to generate a DB2 image copy of the underlying tablespaces. See Chapter 11, “Running and using the LDAP server utilities” for more information about the ds2ldif utility program, and DB2 Utility Guide and Reference for more information about the DB2 image copy.
• There are considerations if the data to be modified by this operation is being replicated. See Modify DN operations and replication for more information.

Eligibility of entries for rename

Entries in the directory which are targeted to be renamed in a single Modify DN operation are subject to these constraints:

1. All entries to be renamed must be located in the same TDBM, LDBM, or CDBM backend targeted by the Modify DN operation. The Modify DN operation with newSuperior option will move subtree entries within the same TDBM, LDBM, or CDBM backend, and will not permit movement of subtree entries from one backend to another. The entry to be renamed must exist in the backend, and the new DN for the entry must not already exist in the backend.

2. Referral entries may be renamed as part of a Modify DN operation. If a referral entry is renamed as part of a Modify DN operation, its corresponding entry in the referral server must be manually updated to reflect the name changes; no automatic updates are propagated to those backends from the target backend. Referrals which exist in other directory servers which refer to any of the entries whose DNs were modified in the local directory by a Modify DN operation will need to be manually updated to reflect the changes; no automatic updates are propagated to those servers from the local one.

3. The LDAP server schema entry can not be renamed.

4. Entries renamed by a Modify DN operation must conform to the LDAP server schema. As such, the RDN attribute type must be consistent with the schema rules for the object classes of the entry: a Modify DN operation fails if the attribute type of newRdn is not in the MUST or MAY list for the entry’s object classes.

5. If a new superior entry is specified, it must be in the same backend as the entry to be renamed but may be under a different suffix managed by that backend. If the IBMModifyDNRealignDNAttributesControl is specified, only entries within the same backend as the renamed entry will be processed.

6. When IBMModifyDNRealignDNAttributesControl is present on a Modify DN request, the operation looks for occurrences of each renamed DN (this can be multiple DNs if renaming a subtree) in certain attributes within all the entries in the backend and replaces each renamed DN with its new DN. The affected attributes are:
   a. Any attribute whose syntax is DN syntax (OID 1.3.6.1.4.1.1466.115.121.1.12).
   b. The aclEntry and entryOwner attributes (these contain DNs in a structured format).

7. If newRdn is specified on a Modify DN operation, each attribute in the newRdn value is added to the entry when it is moved. If a newRdn attribute already has a different value in the entry and the attribute is defined as SINGLE-VALUE in the schema, the Modify DN operation fails. For example,
suppose an entry with DN of dept=AAA,ou=mydivision,o=MyCompany,c=us is to be renamed with the newRdn sector=northeast and that the entry already contains the SINGLE-VALUE attribute sector with a value of northwest. This rename fails because it attempts to add a second value (northeast) to the sector attribute.

If the newRdn attribute is contained in the current RDN, then the deleteoldrdn parameter can be added to the Modify DN operation to allow it to succeed. In this case, the current attribute value is removed so that the attribute only contains the one value from newRdn in the renamed entry. For example, suppose an entry with DN of sector=northwest,ou=mydivision,o=MyCompany,c=us is to be renamed with the newRdn sector=northeast and deleteoldrdn is specified on the Modify DN operation. This rename succeeds because northwest is replaced by northeast as the single value of the sector attribute in the renamed entry.

8. Entries may be renamed only if all access control requirements are satisfied for the bound user, as determined by the effective ACL and ownership permissions for those entries and attributes. See Access control and ownership for detailed explanation and examples of this effect.

9. Alias entries (entries containing the aliasedObjectName attribute and either the alias or aliasObject object class) can be renamed as part of a modify DN operation as long as this does not result in an aliasedObjectName value that is a DN equal to the DN of the renamed alias entry.

10. When advanced replication is configured, a Modify DN operation from one replication context to a different replication context is not supported. The Modify DN operation must occur within the same replication context.

Concurrent considerations between Modify DN operations and other LDAP operations

The ability of the Modify DN operation to rename non-leaf nodes in the directory (which causes all entries which are hierarchical subordinates of the target entry to be renamed) and the ability to move directory subtrees have the potential for affecting many entries in the directory in a single operation. Use of IBMModifyDNRealignDNAttributesControl with this operation may further result in modification of additional directory entries which are outside the scope of the directory subtrees being renamed or moved.

Changes to all entries affected by the operation are committed at the same time. While modified entries are awaiting the transaction commit point, database locks are held which prevent other concurrent operations from sharing and modifying the data. If many entries undergo modification with this operation, it may result in a long-running transaction which has potential for reducing concurrency of other operations targeted for the same directory entries.

Although the LDAP server is capable of processing concurrent LDAP operations targeted at a given TDBM, LDBM, or CDBM backend while the Modify DN operation is in progress, the extent to which such concurrency is possible will depend on what data in the directory may be needed and locked by the competing operations. In addition, if the number of entries being affected by the a TDBM Modify DN operation is large or if the database is small, the underlying DB2 locking mechanism may escalate locking levels, which would result in more entries being excluded from use by other concurrent operations than just those which are modified by the Modify DN operation. It may be advisable to submit such a request during a low-activity period when demand for the same resources by multiple concurrent operations is relatively low.

For example, the Modify DN operation that is shown at 253 and 253 would potentially be susceptible to lock contention when:

- there are concurrent update operations under the new parent "ou=Sport, ou=Vision On Demand, o=Deltawing, c=AU"
- or there are concurrent update operations under the old parent "o=Human Resources, o=Deltawing, c=AU"
- or DB2 locking chooses an access path that results in lock escalation for entries under "o=Deltawing, c=AU"
For more information on DB2 lock escalation, see IBM redbook, *Locking in DB2 for MVS/ESA™ Environment (SG24–4725)*, at http://www.redbooks.ibm.com/ or *DB2 Application Programming and SQL Guide*.

---

### Access control and ownership

For all entries being renamed, the caller must have **w**(rite) permissions for the attribute values that will have to change in all affected entries. In addition, if the `newSuperior` parameter is present on the Modify DN request, the caller must have permissions of **object:a** on the `newSuperior` entry and **object:d** on the target entry at the top of the subtree of entries being moved. If the caller lacks one or more of these permissions, the operation is denied. No access control checking is done against any of the target entry’s subordinates even though their DN is changed. It should be noted that if the caller is an effective owner of any of the entries being renamed, the permissions are automatically satisfied for those entries.

In addition, if the `IBMModifyDNRealignDNAtributesControl` accompanies a Modify DN request, then the bound DN must have **w**(rite) permission to all of the attributes that are changed as a result of realignment of the DN values.

**Example:**

Assume our sample directory contains the following entry which will be the target of a Modify DN operation, and which contains explicit ACL information:

```
dn: o=Athletics, o=Human Resources, ou=Delta Home Media Ltd., o=Deltawing, c=AU
aclEntry: access-id: cn=Mark Crawford, ou=Human Resources, ou=Delta Home Media Ltd.,
o=Deltawing,c=AU:normal:rswc:sensitive:rsc:object:d
```

(other attributes not shown)

The directory also contains an entry with DN `ou=Production, ou=Vision On Demand,o=Deltawing, c=AU` which will be the new Superior of the Modify DN operation. This entry inherits the following ACL information (propagated from a superior entry):

```
acEntry: access-id: dn: cn=Mark Crawford, ou=Human Resources, ou=Delta Home Media Ltd.,
o=Deltawing,c=AU:normal:rwsc:sensitive:rsc:object:a
```

In addition, there are several entries containing attributes of DN syntax. For this example, assume that these attribute types and their respective attribute access classes are as follows:

- **attribute:** reportingOrganization
  - **access-class:** sensitive
- **attribute:** workingOrganization
  - **access-class:** normal

The LDIF format representation of the entries containing `reportingOrganization` or `workingOrganization` attributes are:

```
dn: cn=Lisa Fare, ou=Human Resources, ou=Delta Home Media Ltd., o=Deltawing, c=AU
cn: Lisa Fare
objectclass: organizationalPerson
objectclass: person
objectclass: TOP
aclEntry: access-id: cn=Mark Crawford, ou=Human Resources, ou=Delta Home Media Ltd.,
o=Deltawing,c=AU:normal:rswc:sensitive:rsc
sn: Fare
title: Occupational Health and Safety Administrator
telephonenumber: (07) 635 1432
manager: cn=John Gardner, ou=Human Resources Group, ou=Deltawing InfoSystems,
o=Deltawing, c=AU
secretary: cn=Ian Campbell, o=Deltawing, c=AU
reportingOrganization: o=Athletics, o=Human Resources, ou=Delta Home Media Ltd.,
o=Deltawing, c=AU
```
Relocating an entry

User "cn=Mark Crawford, ou=Human Resources, ou=Delta Home Media Ltd., o=Deltawing, c=AU" submits the following Modify DN operation request to the server to relocate the target entry:

```
ldapmodrdn -h ldaphost -p ldappart -D "cn=Mark Crawford, ou=Human Resources, ou=Delta Home Media Ltd., o=Deltawing, c=AU" -w passwd -s "ou=Production, ou=Vision On Demand, o=Deltawing, c=AU" "o=Athletics, o=Human Resources, ou=Delta Home Media Ltd., o=Deltawing, c=AU" "o=Athletics Division"
```

The -s parameter specifying newSuperior is present on this operation request, so in addition to the access permissions needed for all Modify DN operations (w on affected attributes), the user also needs object:d on the target entry and object:a on the newSuperior entry. The bound user is in the aclEntry for the target entry as well as in the aclEntry for the newSuperior entry, and has all required access permissions (can write attributes and delete the target entry, and can add objects under the newSuperior entry), so the operation is permitted.

Relocating an entry with DN realignment requested

If the same user submits a Modify DN operation request to the server to relocate the same target entry under the same newSuperior entry, but with the addition of the control requesting realignment of DN attribute values (-a parameter):

```
ldapmodrdn -h ldaphost -p ldappart -D "cn=Mark Crawford, ou=Human Resources, ou=Delta Home Media Ltd., o=Deltawing, c=AU" -w passwd -a -s "ou=Production, ou=Vision On Demand, o=Deltawing, c=AU" "o=Athletics, o=Human Resources, ou=Delta Home Media Ltd., o=Deltawing, c=AU" "o=Athletics Division"
```

In addition to the permissions required on the previous example, this operation requires additional permissions to be checked on entries containing values which qualify for realignment. The DN being modified ("o=Athletics, o=Human Resources, ou=Delta Home Media Ltd., o=Deltawing, c=AU") is found in DN-syntax attributes of two entries: The entry with DN "cn=Laurie Wood, ou=Human Resources Group, ou=Deltawing Automotive Ltd., o=Deltawing, c=AU" contains this value in the workingOrganization attribute, and the entry with DN "cn=Lisa Fare, ou=Human Resources, ou=Delta Home Media Ltd., o=Deltawing, c=AU" contains this value in the reportingOrganization attribute.

The bound user is in the aclEntry for "cn=Laurie Wood, ou=Human Resources Group, ou=Deltawing Automotive Ltd., o=Deltawing, c=AU". The workingOrganization attribute is in the access-class of normal, and the bound user is granted w access to this class of attributes, so the realignment of the DN value would be permitted in this entry.

The bound user is also in the aclEntry for "cn=Lisa Fare, ou=Human Resources, ou=Delta Home Media Ltd., o=Deltawing, c=AU". The reportingOrganization attribute is in the access-class of sensitive, and the bound user is granted only rs permissions on sensitive attributes in the entry, so the realignment of this value would be denied. Even though the bound user had adequate permissions to perform the realignment of the DN value, the lack of rs permissions on reportingOrganization attribute in the aclEntry for "cn=Lisa Fare" prevents the realignment from occurring.
relocation of the target entry and had adequate permissions to perform realignment of the DN value in one of the two entries containing a matching DN, the operation would fail because the bound user does not have the necessary permissions on everything needed to complete the operation.

**Access control changes**

If a Modify DN operation is accompanied by the `newSuperior` parameter, changes in effective ACLs and in effective ownership of the relocated entries may result. Regardless of the effective ACLs which applied to the moved subtree in its old location, the moved subtree inherits any propagating ACLs applying to the `newSuperior` entry. As a consequence, entries to which a user had access before the request may no longer be accessible by that user, and entries to which access was denied for a given user before the request is accessible by that user.

Explicit ACLs in the entry or subtree override propagating ACLs. All explicit ACLs which were in the moved subtree at its original location move along with the entries.

When renaming a DN, it is possible that ACLs and entryOwners containing the renamed DN will be modified. Therefore, before such a move or rename users should carefully consider how ownership and accessibility to entries protected by these attributes may change after the move, and what ACL and ownership changes may be desired, if any.

The following is an example of how a Modify DN operation might affect access controls:

```
ModifyDNRequest ::= {
  entry   o=Athletics, o=Human Resources, o=Deltawing, c=AU
  newsrdn o=Adult Athletics
  deleteoldrdn FALSE,
  newSuperior ou=Sport, ou=Vision On Demand, o=Deltawing, c=AU
}
```

```
ldapmodrdn -h ldaphost -p ldapport -D binddn -w passwd -s "ou=Sport, ou=Vision On Demand, o=Deltawing, c=AU" "o=Athletics, o=Human Resources, o=Deltawing, c=AU" "ou=Adult Athletics"
```

![Figure 15. Before Modify Dn operation](image-url)
Assume that the entry with DN `o=Human Resources, o=Deltawing, c=AU` has an explicit propagating ACL containing the following `aclEntry`:

```
aclEntry: access-id: cn=Mark Edmondson, ou=Vision On Demand, ou=Deltawing, o=Deltawing, c=AU:
```

Also, assume that the entry with DN `ou=Sport, ou=Vision On Demand, o=Deltawing, c=AU` has an explicit propagating ACL containing the following `aclEntry`:

```
aclEntry: access-id: cn=Mark Edmondson, ou=Vision On Demand, ou=Deltawing, o=Deltawing, c=AU:
```

If the user bound as DN `cn=Mark Edmondson, ou=Vision On Demand, ou=Deltawing, c=AU` performs the example Modify DN operation, there are at least two consequences which should be noted:

- While this DN previously had `rwcs` permissions on sensitive attributes in the entry `o=Athletics, o=Deltawing, c=AU` and `rws` permissions on critical attributes in the same entry, this DN has only `r` access on both sensitive and critical attributes in the entry after the relocation. It might be expected that a given DN will have the same accessibility to specific entries and data in the directory after a Modify DN operation as it had to those entries and data before the operation, but this example demonstrates that such an expectation is not valid.

- If, after completion of the Modify DN operation, the bound user decides that they want to return the moved entry (and its subordinates) back to their original location in the directory hierarchy, this will not be possible with the access controls currently in place. The bound DN has only `object:d` permission on the old superior node ("o=Human Resources, o=Deltawing, c=AU") where `object:a` is needed to effect the move of an entry or subtree under the superior node, and the bound DN has only `object:a` permission on the moved entry ("o=Adult Athletics, ou=Sport, ou=Vision On Demand, o=Deltawing, c=AU") where `object:d` is needed to move the entry. Therefore, while it may be expected that a given DN can reverse a Modify DN operation under all circumstances, this example demonstrates that such an expectation is not valid.
Ownership changes

When the newSuperior parameter accompanies the Modify DN request, any entries in a relocated subtree which had explicit owners before the relocation will preserve that explicit ownership after the relocation has been performed. Any entries in the relocated subtree which inherited ownership before relocation will continue to inherit ownership following relocation. If the owning entry before relocation was a node superior to the relocated entry, the owning entry will be the new superior entry. If the owning entry was an entry within the relocated subtree, the owning entry is preserved following the relocation.

Any entries in the relocated subtree which propagated ownership to subordinates before relocation continue to propagate ownership to subordinates after the relocation.

Refer to the example in Access control changes.

Assume that the entry with DN o=Human Resources, o=Deltawing, c=AU has an explicit propagating owner of cn=Mark Crawford, ou=Human Resources, ou=Delta Home Media Ltd., o=Deltawing, c=AU.

Also, assume that the entry with DN ou=Sport, ou=Vision On Demand, o=Deltawing, c=AU has an explicit propagating owner of cn=Neville McAuliffe, ou=Human Resources Group, ou=Deltawing Infosystems, o=Deltawing, c=AU.

Before the Modify DN operation, the effective owner of the renamed entry is cn=Mark Crawford, ou=Human Resources, ou=Delta Home Media Ltd., o=Deltawing, c=AU; after completion of the operation, the effective owner of the renamed entry is now cn=Neville McAuliffe, ou=Human Resources Group, ou=Deltawing Infosystems, o=Deltawing, c=AU. Therefore, the act of relocating an entry may change the effective owner of that entry and of its subordinates.

Modify DN operations related to suffix DNs

The Modify DN operation may be used to modify the DNs of any and all entries in a TDBM or LDBM backend. In addition to renaming leaf entries (directory entries with no subordinate entries) and mid-hierarchy entries (directory entries which have both superior entries and subordinate entries), suffix entries may also be renamed. Suffix entries may be renamed to become non-suffix entries and suffix entries may be renamed such that they continue to be suffix entries. In addition, non-suffix entries may be renamed to become suffix entries. This section provides example scenarios for rename operations which involve suffix entries. It summarizes constraints which have been adopted for the LDAP directory implementation which are not defined in the protocol behavior prescribed by RFC 2251 for the Modify DN operation. Examples are provided on how various renaming scenarios may be accomplished, and factors to be considered when performing these operations are discussed.

Note: It is strongly recommended that the cn=ibmpolicies and cn=configuration CDBM suffix entries not be renamed. Renaming these suffixes can cause configuration related problems.

Scenario constraints

Several constraints will apply which are not defined by RFC 2251 in the description of the protocol behavior:

1. If an entry being renamed will become (or remain) a suffix, the new DN must be designated in the server’s configuration file as a suffix for the backend, otherwise the operation will not be permitted.
2. The newRdn parameter of the Modify DN request must contain a non-null value, otherwise the operation request will be treated as an error.
3. If the newSuperior parameter is present, it may contain a zero-length string signifying that the new entry does not have a superior entry, therefore is a suffix entry.
In the directory hierarchy diagrams which follow, a circle outlined with a dashed line represents a component of a suffix DN. Circles containing gray fill represent DNs for which an entry exists in the directory.

Example scenarios

The following are example scenarios:

1. Rename a suffix RDN with no accompanying newSuperior, and the new DN remains a suffix after the rename is completed.

   For example:

   Suffixes defined in the server configuration file:
   
   suffix: ou=End_GPL, o=MyCompany, c=US
   suffix: ou=Endicott, o=MyCompany, c=US

   Rename operation is to rename suffix entry
   ou=End_GPL, o=MyCompany, c=US
   to suffix entry
   ou=Endicott, o=MyCompany, c=US

   The following figure shows an example of this operation:

   ![Diagram showing suffix rename with no new superior](image)

   Figure 17. Suffix rename with no new superior

   The new DN must be already designated as a suffix for this backend, otherwise this operation will fail.

   The operation is performed the same as a rename of any other RDN in the directory

   a. Send Modify DN operation request with
   target=ou=End_GPL, o=MyCompany, c=US

   newRdn=ou=Endicott

   This results in renaming ou=End_GPL, o=MyCompany, c=US to ou=Endicott, o=MyCompany, c=US and in renaming subordinate entries accordingly.

2. Rename of suffix DN with an accompanying newSuperior, and the new DN remains a suffix after the rename is completed. For example:

   Suffix defined in the server configuration file:
   
   suffix: ou=Endicott, o=MyCompany, c=us

Rename operation is to rename suffix entry
ou=Endicott, o=MyCompany, c=us
to suffix entry
o=MyCompany, c=us

The following figure shows an example of this operation:

This scenario, which involves renaming an existing suffix to an overlapping new suffix, must be performed in several steps, since the product does not permit designation in the server configuration file of overlapping suffixes. The definition of overlapping suffixes is when two suffixes with differing numbers of naming components are equal to the extent of the shorter of the two suffixes. For example, ou=Endicott, o=MyCompany, c=US and o=MyCompany, c=US are considered to be overlapping suffixes, while ou=Endicott, o=MyCompany, c=US and ou=Raleigh, o=MyCompany, c=US are not considered to be overlapping suffixes.

This rename can be accomplished by having a temporary suffix pre-defined for the backend (for example, o=OurTemporarySuffix), renaming the target entry to become the temporary suffix, stopping the server and deleting the suffix ou=Endicott, o=MyCompany, c=us and adding the suffix o=MyCompany, c=us, and restarting the server. The temporary suffix would later be deleted from the list of suffixes for the backend.

a. Send a Modify DN operation request with
   target= ou=Endicott, o=MyCompany, c=us
   newRdn= o=OurTemporarySuffix
   newSuperior= ** (present in request with zero-length string)
   This results in renaming ou=Endicott, o=MyCompany, c=us to o=OurTemporarySuffix. Note that the server treats newRdn as an error if it contains a zero-length string, but zero-length strings are permitted in the newSuperior argument to signify that the superior entry is the root DN.

b. Stop server, remove suffix ou=Endicott, o=MyCompany, c=us from the server configuration file, add suffix o=MyCompany, c=us, and restart server.
   This results in adding the desired target suffix without a resulting conflict from overlapping suffixes.

c. Send a Modify DN operation request with
   target= o=OurTemporarySuffix
   newRdn= o=MyCompany
   newSuperior= c=us
This step results in renaming the temporary suffix `o=OurTemporarySuffix` to the desired suffix `o=MyCompany, c=us`, thereby accomplishing the rename from `ou=Endicott, o=MyCompany, c=us` to `o=MyCompany, c=us`. In the process, subordinate entries would be renamed accordingly.

3. This example shows the renaming of a suffix to another overlapping suffix higher in the directory hierarchy. A similar scenario could also be performed involving the rename of a suffix to another overlapping suffix, where the new name is a suffix lower in the directory hierarchy. For example:

**Suffix defined in the server configuration file suffix:**

```plaintext
ou=Endicott, o=MyCompany, c=us
```

**Rename operation is to rename suffix entry:**

```plaintext
ou=Endicott, o=MyCompany, c=us
```

to suffix entry:

```plaintext
div=S390, ou=Endicott, o=MyCompany, c=us
```

The following figure shows an example of this operation:

![Figure 19. Overlapping suffix rename A](image)

This rename can be accomplished by having a temporary suffix pre-defined for this backend in the server configuration file (for example, `o=OurTemporarySuffix`), renaming the target entry to become the temporary suffix, stopping the server and deleting the suffix `ou=Endicott, o=MyCompany, c=us` and adding the suffix `div=S390, ou=Endicott, o=MyCompany, c=us`, and restarting the server. The temporary suffix would later be deleted from the list of suffixes for the backend. This scenario would be done as follows:

a. Send a Modify DN operation request with
   ```plaintext
target= ou=Endicott, o=MyCompany, c=us
newRdn= o=OurTemporarySuffix
newSuperior= ** (present in request with zero-length string)
```

b. Stop server, remove suffix `ou=Endicott, o=MyCompany, c=us`, add suffix `div=S390, ou=Endicott, o=MyCompany, c=us`, and restart server.

c. Send a Modify DN operation request with
   ```plaintext
target= o=OurTemporarySuffix
newRdn= div=S390
newSuperior= ou=Endicott, o=MyCompany, c=us
```
If basic replication is configured, it should be noted that if these operational scenarios are to be replicated from a master server to one or more replica servers, there is a procedure this must be followed to permit this. Advanced replication does not support a Modify DN operation from one replication context to another replication context. Therefore, the following procedure only works with basic replication.

a. Stop the replica server(s), add the temporary suffix (o=OurTemporarySuffix in our examples), restart the replica server(s).

b. On the master server, perform the previous Steps 3a and 3b from the examples above. This will result in the intermediate rename to be performed on the master server and the results to be propagated to the replica server(s).

c. Stop the replica server(s), delete the original suffix (ou=Endicott, o=MyCompany, c=us in both examples above), add the new suffix (o=MyCompany, c=us in the first example above, div=S390, ou=Endicott, o=MyCompany, c=us in the second example above), and restart the replica server(s).

d. On the master server, perform the previous Step 3c from the examples above. This will result in the rename of entries to the final destination on the master server and in the results being propagated to the replica server(s).

4. Rename of suffix DN (some component other than RDN), and the new DN remains a suffix after the rename is completed. For example:

Suffixes defined in the server configuration file:

```
suffix: ou=Endicott, o=MyCompany, c=us
suffix: ou=Endicott, o=MyCompany_ny, c=us
```

Rename operation is to rename suffix entry:

```
ou=Endicott, o=MyCompany, c=us
```

to suffix entry

```
ou=Endicott, o=MyCompany_ny, c=us
```

The following figure shows an example of this operation:

![Figure 20. Overlapping suffix rename B](image)

The new DN must be already designated as a suffix for this backend, otherwise this operation will fail. The operation is performed the same as a rename of any other DN in the directory. The product will permit the rename to occur in one step, even if an entry for newSuperior does not already exist, since the newly-named entry will become a suffix entry.
5. Rename of suffix DN (including some component other than RDN), with an accompanying newSuperior, but the new DN is no longer a suffix. For example:

Suffixes defined in the server configuration file:

- suffix: ou=End, o=MyCompany, c=us
- suffix: ou=End, ou=MyCompany_na, o=MyCompany, c=us

Rename operation is to rename suffix entry ou=End, o=MyCompany, c=us to non-suffix entry ou=GPL, ou=End, ou=MyCompany_na, o=MyCompany, c=us

The following figure shows an example of this operation:

![Figure 21. Suffix rename to non-suffix entry]

The newSuperior entry must already exist before this operation will be permitted.

6. Rename of a non-suffix DN (including some component other than RDN), with an accompanying newSuperior, and the new DN is now a suffix. For example:
Suffixes defined in the server configuration file:
  suffix: ou=End, o=MyCompany, c=us
  suffix: o=Lotus, c=us

Rename operation is to rename non-suffix
div=Lotus, ou=End, o=MyCompany, c=us
to suffix o=Lotus, c=us

The following figure shows an example of this operation:
The new DN must be already designated as a suffix for this backend, otherwise this operation will fail.

Figure 22. Rename non-suffix entry to suffix entry

a. Send a Modify DN operation request with
  target= div=Lotus, ou=Endicott, o=MyCompany, c=us
  newRdn= o=Lotus
  newSuperior= c=us
  This step results in renaming div=Lotus, ou=Endicott, o=MyCompany, c=us to o=Lotus, c=us and
  in renaming subordinate entries accordingly.

Modify DN operations and replication

Modify DN operations may be classified into two categories:
1. Simple Modify DN operations are those that rename a leaf node, and that are not accompanied by the
   newSuperior parameter or the IBMModifyDNRealignDNAttributesControl control or the
   IBMModifyDNTimelimitControl control.
2. Complex Modify DN operations are those that either rename a mid-tree (non-leaf) node, or that are
   accompanied by the newSuperior parameter, or that are accompanied by either the
   IBMModifyDNRealignDNAttributesControl control or the IBMModifyDNTimelimitControl control.

If basic replication is configured, simple Modify DN operations are always accepted by the master server,
and are replicated if replica entries are present in the TDBM, LDBM, or CDBM backend where a Modify
DN operation is applied.

If advanced replication is configured, simple Modify DN operations are only accepted by the supplier
server when the operation occurs within the same replication context.
A compatible server version is one known to support for Modify DN operations all features and controls implemented by the z/OS LDAP server including:

- the IBMModifyDNRealignDNAttributesControl control
- the IBMModifyDNTimelimitControl control
- the newSuperior parameter
- rename of non-leaf entries (complex Modify DN operations).

If one or more of these features or controls is not supported by a replica or consumer server, all complex Modify DN operations are refused at the master server. If advanced replication is configured, complex Modify DN operations are only allowed within the same replication context.

**Initial validation of compatible server versions in consumer and replica servers**

Checks are made of consumer or replica servers by the supplier or master server which are intended to increase the likelihood that complex Modify DN operations will be successfully replicated.

The LDAP server must be able to establish a connection to each of the consumer or replica servers represented by replication agreement entries or replica entries in a TDBM or LDBM backend. When the connection is established to a given consumer or replica server, the supplier or master server determines if the consumer or replica server is at a compatible server version based on a query of the root DSE on that server. If a connection cannot be established to a consumer or replica server, it is assumed that the server does not provide the requisite support for replication of Modify DN operations, and complex Modify DN operations are refused on the consumer or master server. If a connection is established to a consumer or replica server and it is determined that the consumer or replica is not at a compatible server version, complex Modify DN operations are refused at the consumer or master server. In a basic replication environment, the replication of simple Modify DN operations is always permitted, and such operations are always performed at the master server. In an advanced replication environment, simple and complex Modify DN operations must occur within the same replication context, otherwise they are not allowed.

**Periodic validation of compatible server versions in basic replication replicas**

The following periodic replica checks are only performed in a basic replication environment and not in an advanced replication environment.

The master server may enable or disable processing of complex Modify DN operations, depending on dynamically changing states of replica servers and of replica entries within the master server’s TDBM or LDBM backend. It is possible for the server to refuse complex Modify DN operations after having accepted them for some period of time, and it is possible for the server to accept complex Modify DN operations after having refused them for some period of time. Such a change can be triggered by several events.

Each replication cycle tests connections to all replica servers defined by replica entries in the TDBM or LDBM backend, and if a connection can no longer be established to any of the replica servers (even if it had been established to the same replica on the previous replication cycle), the master server begins refusing complex Modify DN operations. If all connections succeed but it is determined that one or more of the replica servers is not at a compatible server version (such as might happen, for example, when the replica server has been stopped when running one version of the LDAP server code and subsequently restarted using a different version of the LDAP server code), the master server begins refusing complex Modify DN operations. Only if connections may be established successfully to all replica servers and if they are determined to be running a compatible server version will the master server resume accepting complex Modify DN operations.

Other possible events which may influence whether the master server accepts or refuses complex Modify DN operations involve:

- the addition of new replica entries
• deletion of existing replica entries
• modification of existing replica entries in the TDBM or LDBM backend.

Each of these causes the master server to temporarily suspend processing of complex Modify DN operations, until the check of replica servers at the start of the next replication cycle, at which point the replica server version levels will be used to determine whether the master server resumes accepting complex Modify DN operations.

To determine whether a replica server is at a compatible version level, submit a root DSE search to that server, similar to the following. The -D and -w options only need to be specified if the replica server does not support anonymous binds.

```
ldapsearch -h ldaphost -p ldapport -D binddn -w passwd
-s base -b "" objectclass=* ibm-enabledCapabilities
```

where ldaphost represents the hostname on which the replica server runs, ldapport is the port number on which the replica server is listening, and binddn and passwd are the distinguished name and password of a user on the replica server.

If the ibm-enabledCapabilities attribute is returned on the root DSE search and its values contain 1.3.18.0.2.32.33 (subtree move) or 1.3.18.0.2.32.34 (subtree rename), then the replica server is capable of supporting those operations.

### Loss of basic replication synchronization because of incompatible replica server versions

The LDAP server basic replication model runs periodically, rather than continuously, and the state of the replica is not checked until the start of each replication cycle. A complex Modify DN operation could be accepted or rejected based on inaccurate information about the state of a replica server between the start of two replication cycles. As a consequence, the basic replication process could stall and the synchronization between the master server and its replicas could be lost.

#### Attention

It is highly recommended that the LDAP server administrator ensure that each replica server is at a compatible server version level before starting a master server which may be the recipient of complex Modify DN operations.

### Loss of basic replication synchronization because of incompatible replica server versions - recovery

If at some point a master server accepts a complex Modify DN operation which cannot be replicated, there are several means of recovering from this situation. The best method of recovering from this situation is to ensure that all replica servers are reachable from the master server, and that all replica servers are running at a compatible version level (this may entail stopping some replica servers and restarting them at a compatible version level). Once this state has been reached, queued changes awaiting propagation to replica servers will drain from the queue at the master server and the replication process will resume normal operation.

An alternative is to delete the replica entry from the master server corresponding to the replica server which is currently unreachable or which is running at an incompatible server level. Note that this will result in loss of synchronization with that replica server, and if you want to later restart the offending replica (such as, after it has been brought up to a compatible server version) it will be necessary to take a backup of the master server contents and restore those contents to the replica server before restarting it, to ensure the two directories are synchronized.
Chapter 16. Accessing RACF information

RACF provides definitions of users, groups, classes, and general resources, and access control for resources. The LDAP server can provide LDAP access to this information stored in RACF.

Using SDBM, the RACF database backend of the LDAP server, you can:

- Add, modify, and delete RACF users, groups, and general resources. Note that dataset resources are not supported.
- Add, modify, and delete user connections to groups
- Add and remove users and groups in general resource access lists
- Modify SETROPTS options that affect classes (for example, RACLIST)
- Retrieve RACF information for users, groups, connections, general resources, and class options
- Retrieve RACF user password and password phrase envelopes

The SDBM backend of the LDAP server implements portions of the adduser, addgroup, rdefine, altuser, altgroup, ralter, permit, setropts, deluser, deigroup, rdelete, connect, remove, and search RACF commands. SDBM makes use of the R_admin "run command" interface to invoke these RACF commands. As a result, this support is subject to the restrictions of the R_admin interface. See z/OS Security Server RACF Callable Services for more information regarding these restrictions. One restriction in particular affects return of search results obtained using the RACF search command. See RACF restriction on amount of output for more details.

SDBM uses the R_admin profile extract functions to retrieve user, group, connection, and resource information. It uses the R_admin setropts extract function to retrieve class options information. These interfaces are not subject to any restrictions on the amount of data returned.

Note that the SDBM backend only updates the default RACF on a given system. That is, the AT and ONLYAT clauses of the RACF commands, used to redirect RACF commands, are not exploited by SDBM.

See z/OS Security Server RACF Command Language Reference for more information about the supported RACF commands.

See Setting up for SDBM for information about getting your LDAP server configured with SDBM.

SDBM authorization

SDBM operations can be performed after several different types of binds to the LDAP server. In each of these binds, the LDAP server associates a RACF user ID with the bound user. SDBM invokes RACF commands under the context of this RACF user ID, and RACF uses its normal authorization processing to determine what this RACF user ID can do.

The supported bind mechanisms are:

- Simple bind to SDBM: The RACF user ID is specified in the bind DN. See Binding using a RACF user ID and password or password phrase on page 270 for more information.
- LDBM, TDBM, or CDBM native authentication bind: The RACF user ID specified in the native authentication entry is used. See Chapter 18, "Native authentication," on page 311 for more information.
- Kerberos bind: The RACF user ID is mapped by SDBM from the Kerberos identity. See "SDBM mapping" on page 306 for more information.
- Certificate bind: The RACF user ID associated with the certificate is used. See "Support of certificate bind" on page 63 for more information.
Binding using a RACF user ID and password or password phrase

The SDBM backend allows for directory authentication (or bind) using the RACF user ID and password or password phrase. The RACF user ID must have an OMVS segment defined and an OMVS UID present. The RACF user and group information that make up an identity can be used to establish access control on other LDAP directory entities. This expands use of the RACF identity to the rest of the LDAP-managed namespace. Note the following when using RACF access:

- An LDAP simple bind to a z/OS LDAP server using RACF access support but having a non-RACF security manager succeeds if the __passwd() call made by the LDAP server is successful. However, no group membership information will be available for the bound distinguished name if the security manager is not RACF.

- An LDAP simple bind made to a z/OS LDAP server using RACF access support provides a successful or unsuccessful LDAP return code. In addition, if the LDAP return code is LDAP_INVALID_CREDENTIALS, additional information is provided in the “message” portion of the LDAP result. The additional information is an LDAP-unique reason code and reason code text in the following format:

  \[ Rnnnnnnn text \]

  The following errno values returned by __passwd() has an LDAP reason code defined for them:

  \begin{table}[h]
  \centering
  \begin{tabular}{|l|l|l|}
  \hline
  \textbf{errno value} & \textbf{Reason} & \textbf{Text} \\
  \hline
  EACCES & R000104 & The password is not correct or the user id is not completely defined (missing password or uid) \\
  \hline
  EINVAL & R000105 & A bind argument is not valid \\
  \hline
  EMVSERR & R004107 & The __passwd function failed; not loaded from a program controlled library \\
  \hline
  EMVSEXPIRE & R000100 & The password has expired \\
  \hline
  EMSPASSWORD & R000101 & The new password is not valid \\
  \hline
  EMVSSAFEXTRERR & R000102 & The user id has been revoked \\
  \hline
  EMVSSAF2ERR & R000104 & The password is not correct or the user id is not completely defined (missing password or uid) \\
  \hline
  EMVSSAF2ERR (system problem) & R004176 & The __passwd() function failed with error error_code \\
  \hline
  EMVSSAF2ERR (userid problem) & R000104 & The password is not correct or the user id is not completely defined (missing password or uid) \\
  \hline
  ESRCH & R000104 & The password is not correct or the user id is not completely defined (missing password or uid) \\
  \hline
  \end{tabular}
  \end{table}

  \textbf{Note:} The same reason codes are issued when binding using a password or a password phrase.

  The return code returned by LDAP is LDAP_OPERATIONS_ERROR when the errno value is EMVSERR or EMVSSAF2ERR (system problem). For the other errno values, the return code is LDAP_INVALID_CREDENTIALS.

  \textbf{Note:} The use of RACF passtickets is supported by the z/OS LDAP server when binding through SDBM. The job name associated with the LDAP Server started task should be used as the application name when generating RACF passtickets. See \textit{z/OS Security Server RACF Macros and Interfaces} for more information about RACF passtickets.
SDBM group gathering

After successfully authenticating to the LDAP server, a list is created of the groups to which the authenticated RACF user ID belongs. Only groups in which the user ID's membership is active (has not been revoked) are included in the list. This group membership list is used in authorization checking when trying to access entries in directories on the LDAP server.

If the SDBM backend is to be used for authentication purposes only and group membership is not needed, consider having your clients use the `authenticateOnly` server control, to streamline bind processing. This control overrides any extended group membership searching and default group membership gathering and is supported for Version 3 clients. See Appendix C, "Supported server controls" for more information.

Note the `authenticateOnly` control is not necessary if there is no TDBM, LDBM, GDBM, or CDBM backend configured. In this case, SDBM does not do any group gathering.

Associating LDAP attributes to RACF fields

Each RACF field in a user, group, connection, and resource profile and in the RACF class options must be associated with an LDAP attribute. The LDAP attribute is used to set the RACF field value in LDAP add and modify operations and to represent the RACF field in LDAP search output. There are two types of RACF fields:
- The fixed fields are defined by RACF. For each profile, these fields make up all the segments supported by SDBM (including the base segment) except the CSDATA segment.
- The custom fields are defined by customers. These fields make up the CSDATA segment in the profile.

Each of these types is associated to an LDAP attribute in a different way.

Associating LDAP attributes to RACF fixed fields

The fields defined by RACF for user, group, connection, and resource profiles, and for class options (setropts) are mapped to predefined attributes in the LDAP schema. These LDAP attributes cannot be deleted or modified and the attribute names cannot be changed. The following tables show the RACF fixed field names and the associated LDAP attribute names for user profiles (Table 37), group profiles (Table 38), connection profiles (Table 39), resource profiles (Table 40) and setropts (Table 41). The RACF names in the table are the keywords used to set the field in RACF commands or used by RACF in display output (for display-only fields). Not all names apply to all versions of LDAP and RACF.

Table 37. Mapping of LDAP attribute names to RACF fixed fields (user)

<table>
<thead>
<tr>
<th>RACF segment name</th>
<th>RACF keyword in altuser/adduser/listuser</th>
<th>LDAP attribute name</th>
</tr>
</thead>
<tbody>
<tr>
<td>User base</td>
<td>ADDCATEGORY</td>
<td>racfSecurityCategoryList</td>
</tr>
<tr>
<td>User base</td>
<td>Multi-value: ADSP, SPECIAL, OPERATIONS, GRPACC, AUDITOR, OIDCARD, UAUDIT, or any other one-word values, such as NOEXPIRED and NOOMVS</td>
<td>racfAttributes</td>
</tr>
<tr>
<td>User base</td>
<td>AUTH not displayed by LDAP</td>
<td>racfConnectGroupAuthority</td>
</tr>
<tr>
<td>User base</td>
<td>CLAUTH</td>
<td>racfClassName</td>
</tr>
<tr>
<td>User base</td>
<td>DFLTGRP</td>
<td>racfDefaultGroup</td>
</tr>
<tr>
<td>User base</td>
<td>GROUP</td>
<td>racfConnectGroupName</td>
</tr>
<tr>
<td>User base</td>
<td>Not modifiable - displayed as LAST-ACCESS</td>
<td>racfLastAccess</td>
</tr>
<tr>
<td>User base</td>
<td>NAME</td>
<td>racfProgrammerName</td>
</tr>
</tbody>
</table>
Table 37. Mapping of LDAP attribute names to RACF fixed fields (user) (continued)

<table>
<thead>
<tr>
<th>RACF segment name</th>
<th>RACF keyword in altuser/adduser/listuser</th>
<th>LDAP attribute name</th>
</tr>
</thead>
<tbody>
<tr>
<td>User base</td>
<td>Not modifiable - displayed as PASSDATE</td>
<td>racfPasswordChangeDate</td>
</tr>
<tr>
<td>User base</td>
<td>Not modifiable - displayed as PASS-INTERVAL</td>
<td>racfPasswordInterval</td>
</tr>
<tr>
<td>User base</td>
<td>PASSWORD</td>
<td>racfPassword</td>
</tr>
<tr>
<td>User base</td>
<td>password envelope - not modifiable</td>
<td>racfPasswordEnvelope</td>
</tr>
<tr>
<td>User base</td>
<td>Not modifiable - displayed as PASSWORD ENVELOPED</td>
<td>racfHavePasswordEnvelope</td>
</tr>
<tr>
<td>User base</td>
<td>password phrase envelope - not modifiable</td>
<td>racfPassPhraseEnvelope</td>
</tr>
<tr>
<td>User base</td>
<td>PHRASE</td>
<td>racfPassPhrase</td>
</tr>
<tr>
<td>User base</td>
<td>Not modifiable - displayed as PHRASEDATE</td>
<td>racfPassPhraseChangeDate</td>
</tr>
<tr>
<td>User base</td>
<td>Not modifiable - displayed as PHRASE ENVELOPED</td>
<td>racfHavePassPhraseEnvelope</td>
</tr>
<tr>
<td>User base</td>
<td>RESUME</td>
<td>racfResumeDate</td>
</tr>
<tr>
<td>User base</td>
<td>REVOKE</td>
<td>racfRevokeDate</td>
</tr>
<tr>
<td>User base</td>
<td>SECLABEL</td>
<td>racfSecurityLabel</td>
</tr>
<tr>
<td>User base</td>
<td>SECLEVEL</td>
<td>racfSecurityLevel</td>
</tr>
<tr>
<td>User base</td>
<td>UACC - value is not displayed by LDAP</td>
<td>racfConnectGroupUACC</td>
</tr>
<tr>
<td>User base</td>
<td>WHEN(DAYS())</td>
<td>racfLogonDays</td>
</tr>
<tr>
<td>User base</td>
<td>WHEN(TIME())</td>
<td>racfLogonTime</td>
</tr>
<tr>
<td>User base or Group base</td>
<td>Not modifiable - displayed as CREATED</td>
<td>racfAuthorizationDate</td>
</tr>
<tr>
<td>User base or Group base</td>
<td>DATA</td>
<td>racfInstallationData</td>
</tr>
<tr>
<td>User base or Group base</td>
<td>MODEL</td>
<td>racfDatasetModel</td>
</tr>
<tr>
<td>User base or Group base</td>
<td>OWNER</td>
<td>racfOwner</td>
</tr>
<tr>
<td>CICS® segment</td>
<td>OPCLASS</td>
<td>racfOperatorClass</td>
</tr>
<tr>
<td>CICS segment</td>
<td>OPIDENT</td>
<td>racfOperatorIdentification</td>
</tr>
<tr>
<td>CICS segment</td>
<td>OPPRTY</td>
<td>racfOperatorPriority</td>
</tr>
<tr>
<td>CICS segment</td>
<td>RSLKEY</td>
<td>racfRslKey</td>
</tr>
<tr>
<td>CICS segment</td>
<td>TIMEOUT</td>
<td>racfTerminalTimeout</td>
</tr>
<tr>
<td>CICS segment</td>
<td>TSLKEY</td>
<td>racfTslKey</td>
</tr>
<tr>
<td>CICS segment</td>
<td>XRFSOFF</td>
<td>racfOperatorReSignon</td>
</tr>
<tr>
<td>DCE segment</td>
<td>AUTOLOGIN</td>
<td>racfDCEAutoLogin</td>
</tr>
<tr>
<td>DCE segment</td>
<td>DCENAME</td>
<td>racfDCEPrincipal</td>
</tr>
<tr>
<td>DCE segment</td>
<td>HOMECELL</td>
<td>racfDCEHomeCell</td>
</tr>
<tr>
<td>DCE segment</td>
<td>HOMEUUID</td>
<td>racfDCEHomeCellUUID</td>
</tr>
<tr>
<td>DCE segment</td>
<td>UID</td>
<td>racfDCEUUID</td>
</tr>
</tbody>
</table>
Table 37. Mapping of LDAP attribute names to RACF fixed fields (user) (continued)

<table>
<thead>
<tr>
<th>RACF segment name</th>
<th>RACF keyword in altuser/adduser/listuser</th>
<th>LDAP attribute name</th>
</tr>
</thead>
<tbody>
<tr>
<td>DFP segment - common to group or user</td>
<td>DATAAPPL</td>
<td>SAFDfpDataApplication</td>
</tr>
<tr>
<td>DFP segment - common to group or user</td>
<td>DATACLAS</td>
<td>SAFDfpDataClass</td>
</tr>
<tr>
<td>DFP segment - common to group or user</td>
<td>MGMTCLAS</td>
<td>SAFDfpmgmtClass</td>
</tr>
<tr>
<td>DFP segment - common to group or user</td>
<td>STORCLAS</td>
<td>SAFDfpStorageClass</td>
</tr>
<tr>
<td>EIM segment</td>
<td>LDAPPROF</td>
<td>racfLDAPProf</td>
</tr>
<tr>
<td>KERB segment</td>
<td>ENCRYPT</td>
<td>racfEncryptType</td>
</tr>
<tr>
<td>KERB segment</td>
<td>KERBNAME</td>
<td>krbPrincipalName</td>
</tr>
<tr>
<td>KERB segment</td>
<td>Not modifiable - displayed as KEY FROM</td>
<td>racfKerbKeyFrom</td>
</tr>
<tr>
<td>KERB segment</td>
<td>Not modifiable - displayed as KEY VERSION</td>
<td>racfCurKeyVersion</td>
</tr>
<tr>
<td>KERB segment</td>
<td>MAXTTLIFE</td>
<td>maxTicketAge</td>
</tr>
<tr>
<td>LANGUAGE segment</td>
<td>PRIMARY</td>
<td>racfPrimaryLanguage</td>
</tr>
<tr>
<td>LANGUAGE segment</td>
<td>SECONDARY</td>
<td>racfSecondaryLanguage</td>
</tr>
<tr>
<td>LNOTES segment</td>
<td>SNAME</td>
<td>racfLNotesShortName</td>
</tr>
<tr>
<td>NDS segment</td>
<td>UNAME</td>
<td>racfNDSUserName</td>
</tr>
<tr>
<td>NETVIEW segment</td>
<td>CONSNAME</td>
<td>racfDefaultConsoleName</td>
</tr>
<tr>
<td>NETVIEW segment</td>
<td>CTL</td>
<td>racfCTLKeyword</td>
</tr>
<tr>
<td>NETVIEW segment</td>
<td>DOMAINS</td>
<td>racfDomains</td>
</tr>
<tr>
<td>NETVIEW segment</td>
<td>IC</td>
<td>racfNetviewInitialCommand</td>
</tr>
<tr>
<td>NETVIEW segment</td>
<td>MSGRECVR</td>
<td>racfMSGRCVRKeyword</td>
</tr>
<tr>
<td>NETVIEW segment</td>
<td>NGMFADMN</td>
<td>racfNGMFADMKeyword</td>
</tr>
<tr>
<td>NETVIEW segment</td>
<td>NGMFVSPN</td>
<td>racfNGMFVSPNKeyword</td>
</tr>
<tr>
<td>NETVIEW segment</td>
<td>OPCLASS</td>
<td>racfNetviewOperatorClass</td>
</tr>
<tr>
<td>User OMVS segment</td>
<td>ASSIZEMAX</td>
<td>racfOmvsMaximumAddressSpaceSize</td>
</tr>
<tr>
<td>User OMVS segment</td>
<td>CPUTIMEMAX</td>
<td>racfOmvsMaximumCPUTime</td>
</tr>
<tr>
<td>User OMVS segment</td>
<td>FILEPROCMA</td>
<td>racfOmvsMaximumFilesPerProcess</td>
</tr>
<tr>
<td>User OMVS segment</td>
<td>HOME</td>
<td>racfOmvsHome</td>
</tr>
<tr>
<td>User OMVS segment</td>
<td>MEMLIMIT</td>
<td>racfOmvsMemoryLimit</td>
</tr>
<tr>
<td>User OMVS segment</td>
<td>MMAPAREAMAX</td>
<td>racfOmvsMaximumMemoryMapArea</td>
</tr>
<tr>
<td>User OMVS segment</td>
<td>PROCUSERMA</td>
<td>racfOmvsMaximumProcessesPerUID</td>
</tr>
<tr>
<td>User OMVS segment</td>
<td>PROGRAM</td>
<td>racfOmvsInitialProgram</td>
</tr>
<tr>
<td>User OMVS segment</td>
<td>SHARED, AUTOUID</td>
<td>racfOmvsUidKeyword</td>
</tr>
<tr>
<td>User OMVS segment</td>
<td>SHMEMMAX</td>
<td>racfOmvsSharedMemoryMaximum</td>
</tr>
<tr>
<td>User OMVS segment</td>
<td>THREADSMAX</td>
<td>racfOmvsMaximumThreadsPerProcess</td>
</tr>
<tr>
<td>User OMVS segment</td>
<td>UID</td>
<td>racfOmvsUid</td>
</tr>
<tr>
<td>OPERPARM segment</td>
<td>ALTGR</td>
<td>racfAltGroupKeyword</td>
</tr>
</tbody>
</table>
### Table 37. Mapping of LDAP attribute names to RACF fixed fields (user) (continued)

<table>
<thead>
<tr>
<th>RACF segment name</th>
<th>RACF keyword in altuser/adduser/listuser</th>
<th>LDAP attribute name</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPERPARM segment</td>
<td>AUTH</td>
<td>racfAuthKeyword</td>
</tr>
<tr>
<td>OPERPARM segment</td>
<td>AUTO</td>
<td>racfAutoKeyword</td>
</tr>
<tr>
<td>OPERPARM segment</td>
<td>CMDSYS</td>
<td>racfCMDSYSKeyword</td>
</tr>
<tr>
<td>OPERPARM segment</td>
<td>DOM</td>
<td>racfDOMKeyword</td>
</tr>
<tr>
<td>OPERPARM segment</td>
<td>HC</td>
<td>racfHcKeyword</td>
</tr>
<tr>
<td>OPERPARM segment</td>
<td>INTIDS</td>
<td>racfIntidsKeyword</td>
</tr>
<tr>
<td>OPERPARM segment</td>
<td>KEY</td>
<td>racfKEYKeyword</td>
</tr>
<tr>
<td>OPERPARM segment</td>
<td>LEVEL</td>
<td>racfLevelKeyword</td>
</tr>
<tr>
<td>OPERPARM segment</td>
<td>LOGCMDRESP</td>
<td>racfLogCommandResponseKeyword</td>
</tr>
<tr>
<td>OPERPARM segment</td>
<td>MFORM</td>
<td>racfMformKeyword</td>
</tr>
<tr>
<td>OPERPARM segment</td>
<td>MIGID</td>
<td>racfMGIDKeyword</td>
</tr>
<tr>
<td>OPERPARM segment</td>
<td>MONITOR</td>
<td>racfMonitorKeyword</td>
</tr>
<tr>
<td>OPERPARM segment</td>
<td>MSCOPE</td>
<td>racfMscopeSystems</td>
</tr>
<tr>
<td>OPERPARM segment</td>
<td>ROUTCODE</td>
<td>racfRoutcodeKeyword</td>
</tr>
<tr>
<td>OPERPARM segment</td>
<td>STORAGE</td>
<td>racfStorageKeyword</td>
</tr>
<tr>
<td>OPERPARM segment</td>
<td>UD</td>
<td>racfUDKeyword</td>
</tr>
<tr>
<td>OPERPARM segment</td>
<td>UNKNIDS</td>
<td>racfUnknidsKeyword</td>
</tr>
<tr>
<td>User OVM segment</td>
<td>FSROOT</td>
<td>racfOvmFileSystemRoot</td>
</tr>
<tr>
<td>User OVM segment</td>
<td>HOME</td>
<td>racfOvmHome</td>
</tr>
<tr>
<td>User OVM segment</td>
<td>PROGRAM</td>
<td>racfOvmInitialProgram</td>
</tr>
<tr>
<td>User OVM segment</td>
<td>UID</td>
<td>racfOvmUid</td>
</tr>
<tr>
<td>PROXY segment</td>
<td>BINDDN</td>
<td>racfLDAPBindDN</td>
</tr>
<tr>
<td>PROXY segment</td>
<td>BINDPW - value is not displayed by LDAP</td>
<td>racfLDAPBindPw</td>
</tr>
<tr>
<td>PROXY segment</td>
<td>LDAHOST</td>
<td>racfLDAPHost</td>
</tr>
<tr>
<td>TSO segment</td>
<td>ACCTNUM</td>
<td>SAFAccountNumber</td>
</tr>
<tr>
<td>TSO segment</td>
<td>COMMAND</td>
<td>SAFDefaultCommand</td>
</tr>
<tr>
<td>TSO segment</td>
<td>DEST</td>
<td>SAFDestination</td>
</tr>
<tr>
<td>TSO segment</td>
<td>HOLDCLASS</td>
<td>SAFHoldClass</td>
</tr>
<tr>
<td>TSO segment</td>
<td>JOBCLASS</td>
<td>SAFJobClass</td>
</tr>
<tr>
<td>TSO segment</td>
<td>MAXSIZE</td>
<td>SAFMaximumRegionSize</td>
</tr>
<tr>
<td>TSO segment</td>
<td>MSGCLASS</td>
<td>SAFMessageClass</td>
</tr>
<tr>
<td>TSO segment</td>
<td>PROC</td>
<td>SAFDefaultLoginProc</td>
</tr>
<tr>
<td>TSO segment</td>
<td>SECLABEL</td>
<td>SAFTsoSecurityLabel</td>
</tr>
<tr>
<td>TSO segment</td>
<td>SIZE</td>
<td>SAFLogonSize</td>
</tr>
<tr>
<td>TSO segment</td>
<td>SYSOUTCLASS</td>
<td>SAFDefaultSysoutClass</td>
</tr>
<tr>
<td>TSO segment</td>
<td>UNIT</td>
<td>SAFDefaultUnit</td>
</tr>
<tr>
<td>TSO segment</td>
<td>USERDATA</td>
<td>SAFUserdata</td>
</tr>
<tr>
<td>WORKATTR segment</td>
<td>WAACCNT</td>
<td>racfWorkAttrAccountNumber</td>
</tr>
</tbody>
</table>
Table 37. Mapping of LDAP attribute names to RACF fixed fields (user) (continued)

<table>
<thead>
<tr>
<th>RACF segment name</th>
<th>RACF keyword in altuser/adduser/listuser</th>
<th>LDAP attribute name</th>
</tr>
</thead>
<tbody>
<tr>
<td>WORKATTR segment</td>
<td>WAADDR1</td>
<td>racfAddressLine1</td>
</tr>
<tr>
<td>WORKATTR segment</td>
<td>WAADDR2</td>
<td>racfAddressLine2</td>
</tr>
<tr>
<td>WORKATTR segment</td>
<td>WAADDR3</td>
<td>racfAddressLine3</td>
</tr>
<tr>
<td>WORKATTR segment</td>
<td>WAADDR4</td>
<td>racfAddressLine4</td>
</tr>
<tr>
<td>WORKATTR segment</td>
<td>WABLDG</td>
<td>racfBuilding</td>
</tr>
<tr>
<td>WORKATTR segment</td>
<td>WADEPT</td>
<td>racfDepartment</td>
</tr>
<tr>
<td>WORKATTR segment</td>
<td>WANAME</td>
<td>racfWorkAttrUserName</td>
</tr>
<tr>
<td>WORKATTR segment</td>
<td>WAROOM</td>
<td>racfRoom</td>
</tr>
</tbody>
</table>

Table 38. Mapping of LDAP attribute names to RACF fixed fields (group)

<table>
<thead>
<tr>
<th>RACF segment name</th>
<th>RACF keyword in altgroup/addgroup/listgrp</th>
<th>LDAP attribute name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group base</td>
<td>SUPGROUP</td>
<td>racfSuperiorGroup</td>
</tr>
<tr>
<td>Group base</td>
<td>Not modifiable - displayed as SUBGROUP(S)</td>
<td>racfSubGroupName</td>
</tr>
<tr>
<td>Group base</td>
<td>TERMUACC</td>
<td>racfGroupNoTermUAC</td>
</tr>
<tr>
<td>Group base</td>
<td>UNIVERSAL</td>
<td>racfGroupUniversal</td>
</tr>
<tr>
<td>Group base</td>
<td>Not modifiable - displayed as USER(S)</td>
<td>racfGroupUserids</td>
</tr>
<tr>
<td>User base or Group base</td>
<td>Not modifiable - displayed as CREATED</td>
<td>racfAuthorizationDate</td>
</tr>
<tr>
<td>User base or Group base</td>
<td>DATA</td>
<td>racfInstallationData</td>
</tr>
<tr>
<td>User base or Group base</td>
<td>MODEL</td>
<td>racfDatasetModel</td>
</tr>
<tr>
<td>User base or Group base</td>
<td>OWNER</td>
<td>racfOwner</td>
</tr>
<tr>
<td>DFP segment - common to group or user</td>
<td>DATAAPPL</td>
<td>SAFDfpDataApplication</td>
</tr>
<tr>
<td>DFP segment - common to group or user</td>
<td>DATACLAS</td>
<td>SAFDfpDataClass</td>
</tr>
<tr>
<td>DFP segment - common to group or user</td>
<td>MGMTCLAS</td>
<td>SAFDfpManagementClass</td>
</tr>
<tr>
<td>DFP segment - common to group or user</td>
<td>STORCLAS</td>
<td>SAFDfpStorageClass</td>
</tr>
<tr>
<td>Group OMVS segment</td>
<td>GId</td>
<td>racfOmvsGroupId</td>
</tr>
<tr>
<td>Group OMVS segment</td>
<td>SHARED, AUTOGID</td>
<td>racfOmvsGroupIdKeyword</td>
</tr>
<tr>
<td>Group OVM segment</td>
<td>GId</td>
<td>racfOvmGroupId</td>
</tr>
</tbody>
</table>

Table 39. Mapping of LDAP attribute names to RACF fixed fields (connection)

<table>
<thead>
<tr>
<th>RACF segment name</th>
<th>RACF keyword in connect</th>
<th>LDAP attribute name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection base</td>
<td>Multi-value: ADSP, AUDITOR GRPACC, OPERATIONS, SPECIAL</td>
<td>racfConnectAttributes</td>
</tr>
<tr>
<td>Connection base</td>
<td>AUTHORITY</td>
<td>racfConnectGroupAuthority</td>
</tr>
<tr>
<td>Connection base</td>
<td>Not modifiable - displayed as CONNECT-DATE</td>
<td>racfConnectAuthDate</td>
</tr>
<tr>
<td>Connection base</td>
<td>Not modifiable - displayed as CONNECTS</td>
<td>racfConnectCount</td>
</tr>
<tr>
<td>Connection base</td>
<td>Not modifiable - displayed as LAST-CONNECT</td>
<td>racfConnectLastConnect</td>
</tr>
</tbody>
</table>
### Table 39. Mapping of LDAP attribute names to RACF fixed fields (connection) (continued)

<table>
<thead>
<tr>
<th>RACF segment name</th>
<th>RACF keyword in connect</th>
<th>LDAP attribute name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection base</td>
<td>OWNER</td>
<td>racfConnectOwner</td>
</tr>
<tr>
<td>Connection base</td>
<td>RESUME</td>
<td>racfConnectResumeDate</td>
</tr>
<tr>
<td>Connection base</td>
<td>REVOKE</td>
<td>racfConnectRevokeDate</td>
</tr>
<tr>
<td>Connection base</td>
<td>UACC</td>
<td>racfConnectGroupUACC</td>
</tr>
</tbody>
</table>

### Table 40. Mapping of LDAP attribute names to RACF fixed fields (resource)

<table>
<thead>
<tr>
<th>RACF segment name</th>
<th>RACF keyword in rdefine/ralter/permit</th>
<th>LDAP attribute name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource base</td>
<td>Multi-value: SINGLEDSN, TVTOC, WARNING, or any other one-word values, such as NOKERB</td>
<td>racfResourceAttributes</td>
</tr>
<tr>
<td>Resource base</td>
<td>ADDCATEGORY</td>
<td>racfSecurityCategoryList</td>
</tr>
<tr>
<td>Resource base</td>
<td>ADDMEM</td>
<td>racfMemberList</td>
</tr>
<tr>
<td>Resource base</td>
<td>ADDVOL</td>
<td>racfVolumeList</td>
</tr>
<tr>
<td>Resource base</td>
<td>Not modifiable - displayed as ALTER COUNT</td>
<td>racfAlterAccessCount</td>
</tr>
<tr>
<td>Resource base</td>
<td>APPLDATA</td>
<td>racfApplData</td>
</tr>
<tr>
<td>Resource base</td>
<td>AUDIT</td>
<td>racfResourceAudit</td>
</tr>
<tr>
<td>Resource base</td>
<td>Not modifiable - displayed as AUTOMATIC</td>
<td>racfAutomatic</td>
</tr>
<tr>
<td>Resource base</td>
<td>Not modifiable - displayed as CONTROL COUNT</td>
<td>racfControlAccessCount</td>
</tr>
<tr>
<td>Resource base</td>
<td>Not modifiable - displayed as CREATION DATE</td>
<td>racfAuthorizationDate</td>
</tr>
<tr>
<td>Resource base</td>
<td>DATA</td>
<td>racfInstallationData</td>
</tr>
<tr>
<td>Resource base</td>
<td>FCLASS, FGENERIC, FROM, FVOLUME - value is not displayed by LDAP</td>
<td>racfCopyProfileFrom</td>
</tr>
<tr>
<td>Resource base</td>
<td>GLOBALAUDIT</td>
<td>racfResourceGlobalAudit</td>
</tr>
<tr>
<td>Resource base</td>
<td>Not modifiable - displayed as LAST CHANGE DATE</td>
<td>racfLastReferenceDate</td>
</tr>
<tr>
<td>Resource base</td>
<td>LEVEL</td>
<td>racfLevel</td>
</tr>
<tr>
<td>Resource base</td>
<td>NOTIFY</td>
<td>racfNotify</td>
</tr>
<tr>
<td>Resource base</td>
<td>OWNER</td>
<td>racfOwner</td>
</tr>
<tr>
<td>Resource base</td>
<td>Not modifiable - displayed as READ COUNT</td>
<td>racfReadAccessCount</td>
</tr>
<tr>
<td>Resource base</td>
<td>SECLABEL</td>
<td>racfSecurityLabel</td>
</tr>
<tr>
<td>Resource base</td>
<td>SECLEVEL</td>
<td>racfSecurityLevel</td>
</tr>
<tr>
<td>Resource base</td>
<td>TIMEZONE</td>
<td>racfTimeZone</td>
</tr>
<tr>
<td>Resource base</td>
<td>UACC</td>
<td>racfUacc</td>
</tr>
<tr>
<td>Resource base</td>
<td>Not modifiable - displayed as UPDATE COUNT</td>
<td>racfUpdateAccessCount</td>
</tr>
<tr>
<td>Resource base</td>
<td>WHEN(DAYS())</td>
<td>racfLogonDays</td>
</tr>
<tr>
<td>Resource base</td>
<td>WHEN(TIME())</td>
<td>racfLogonTime</td>
</tr>
<tr>
<td>Resource base</td>
<td>Any of these PERMIT command keywords: ACCESS, DELETE, FCLASS, FGENERIC, FROM, FVOLUME, ID, RESET, WHEN</td>
<td>racfAccessControl</td>
</tr>
<tr>
<td>RACF segment name</td>
<td>RACF keyword in redefine/ralter/permit</td>
<td>LDAP attribute name</td>
</tr>
<tr>
<td>------------------</td>
<td>----------------------------------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>CDTINFO segment</td>
<td>CASE</td>
<td>racfCdtinfoCase</td>
</tr>
<tr>
<td>CDTINFO segment</td>
<td>DEFAULTRC</td>
<td>racfCdtinfoDefaultRc</td>
</tr>
<tr>
<td>CDTINFO segment</td>
<td>DEFAULTUACC</td>
<td>racfCdtinfoDefaultUacc</td>
</tr>
<tr>
<td>CDTINFO segment</td>
<td>FIRST</td>
<td>racfCdtinfoFirst</td>
</tr>
<tr>
<td>CDTINFO segment</td>
<td>GENERIC</td>
<td>racfCdtinfoGeneric</td>
</tr>
<tr>
<td>CDTINFO segment</td>
<td>GENLIST</td>
<td>racfCdtinfoGenList</td>
</tr>
<tr>
<td>CDTINFO segment</td>
<td>GROUP</td>
<td>racfCdtinfoGroup</td>
</tr>
<tr>
<td>CDTINFO segment</td>
<td>KEYQUALIFIERS</td>
<td>racfCdtinfoKeyQualifiers</td>
</tr>
<tr>
<td>CDTINFO segment</td>
<td>MACPROCESSING</td>
<td>racfCdtinfoMacProcessing</td>
</tr>
<tr>
<td>CDTINFO segment</td>
<td>MAXLENGTH</td>
<td>racfCdtinfoMaxLength</td>
</tr>
<tr>
<td>CDTINFO segment</td>
<td>MAXLENX</td>
<td>racfCdtinfoMaxLengthX</td>
</tr>
<tr>
<td>CDTINFO segment</td>
<td>MEMBER</td>
<td>racfCdtinfoMember</td>
</tr>
<tr>
<td>CDTINFO segment</td>
<td>OPERATIONS</td>
<td>racfCdtinfoOperations</td>
</tr>
<tr>
<td>CDTINFO segment</td>
<td>OTHER</td>
<td>racfCdtinfoOther</td>
</tr>
<tr>
<td>CDTINFO segment</td>
<td>POSIT</td>
<td>racfCdtinfoPosit</td>
</tr>
<tr>
<td>CDTINFO segment</td>
<td>PROFILESALLOWED</td>
<td>racfCdtinfoProfilesAllowed</td>
</tr>
<tr>
<td>CDTINFO segment</td>
<td>RACLST</td>
<td>racfCdtinfoRacList</td>
</tr>
<tr>
<td>CDTINFO segment</td>
<td>SECLABELSREQUIRED</td>
<td>racfCdtinfoSecLabelsRequired</td>
</tr>
<tr>
<td>CDTINFO segment</td>
<td>SIGNAL</td>
<td>racfCdtinfoSignal</td>
</tr>
<tr>
<td>CFDEF segment</td>
<td>FIRST</td>
<td>racfCfdefFirst</td>
</tr>
<tr>
<td>CFDEF segment</td>
<td>HELP</td>
<td>racfCfdefHelp</td>
</tr>
<tr>
<td>CFDEF segment</td>
<td>LISTHEAD</td>
<td>racfCfdefListHead</td>
</tr>
<tr>
<td>CFDEF segment</td>
<td>MAXLENGTH</td>
<td>racfCfdefMaxLength</td>
</tr>
<tr>
<td>CFDEF segment</td>
<td>MAXVALUE</td>
<td>racfCfdefMaxValue</td>
</tr>
<tr>
<td>CFDEF segment</td>
<td>MINVALUE</td>
<td>racfCfdefMinValue</td>
</tr>
<tr>
<td>CFDEF segment</td>
<td>MIXED</td>
<td>racfCfdefMixed</td>
</tr>
<tr>
<td>CFDEF segment</td>
<td>OTHER</td>
<td>racfCfdefOther</td>
</tr>
<tr>
<td>CFDEF segment</td>
<td>TYPE</td>
<td>racfCfdefType</td>
</tr>
<tr>
<td>DLFDATA segment</td>
<td>JOBNAMES</td>
<td>racfDlfdataJobNames</td>
</tr>
<tr>
<td>DLFDATA segment</td>
<td>RETAIN</td>
<td>racfDlfdataRetain</td>
</tr>
<tr>
<td>EIM segment</td>
<td>DOMAINDN</td>
<td>racfEimDomainDn</td>
</tr>
<tr>
<td>EIM segment</td>
<td>KERBREGISTRY</td>
<td>racfEimKerbRegistry</td>
</tr>
<tr>
<td>EIM segment</td>
<td>LOCALREGISTRY</td>
<td>racfEimLocalRegistry</td>
</tr>
<tr>
<td>EIM segment</td>
<td>OPTIONS</td>
<td>racfEimOptions</td>
</tr>
<tr>
<td>EIM segment</td>
<td>X509REGISTRY</td>
<td>racfEimX509Registry</td>
</tr>
<tr>
<td>ICSF segment</td>
<td>ASYMUSAGE</td>
<td>racficsfAsymUsage</td>
</tr>
<tr>
<td>ICSF segment</td>
<td>SYMEXPORTABLE</td>
<td>racficsfSymExportable</td>
</tr>
<tr>
<td>ICSF segment</td>
<td>SYMEXPORTCERTS</td>
<td>racficsfSymExportCerts</td>
</tr>
<tr>
<td>ICSF segment</td>
<td>SYMEXPORTKEYS</td>
<td>racficsfSymExportKeys</td>
</tr>
<tr>
<td>ICTX segment</td>
<td>DOMAP</td>
<td>racfictxDoMap</td>
</tr>
</tbody>
</table>
Table 40. Mapping of LDAP attribute names to RACF fixed fields (resource)  (continued)

<table>
<thead>
<tr>
<th>RACF segment name</th>
<th>RACF keyword in rdefine/alter/permit</th>
<th>LDAP attribute name</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICTX segment</td>
<td>MAPPINGTIMEOUT</td>
<td>racfIctxMappingTimeOut</td>
</tr>
<tr>
<td>ICTX segment</td>
<td>MAPREQUIRED</td>
<td>racfIctxMapRequired</td>
</tr>
<tr>
<td>ICTX segment</td>
<td>USEMAP</td>
<td>racfIctxUseMap</td>
</tr>
<tr>
<td>KERB segment</td>
<td>DEFTKTLFE</td>
<td>racfKerbDefaultTicketLife</td>
</tr>
<tr>
<td>KERB segment</td>
<td>ENCRYPT</td>
<td>racfEncryptType</td>
</tr>
<tr>
<td>KERB segment</td>
<td>KERBNAME</td>
<td>krbPrincipalName</td>
</tr>
<tr>
<td>KERB segment</td>
<td>Not modifiable - displayed as KEY VERSION</td>
<td>racfCurKeyVersion</td>
</tr>
<tr>
<td>KERB segment</td>
<td>MAXTKTLFE</td>
<td>maxTicketAge</td>
</tr>
<tr>
<td>KERB segment</td>
<td>MINTKTLFE</td>
<td>racfKerbMinTicketLife</td>
</tr>
<tr>
<td>KERB segment</td>
<td>PASSWORD - value is not displayed by LDAP</td>
<td>racfKerbPassword</td>
</tr>
<tr>
<td>PROXY segment</td>
<td>BINDDN</td>
<td>racfLDAPBindDn</td>
</tr>
<tr>
<td>PROXY segment</td>
<td>BINDPW - value is not displayed by LDAP</td>
<td>racfLDAPBindPw</td>
</tr>
<tr>
<td>PROXY segment</td>
<td>LDAPHOST</td>
<td>racfLDAPHost</td>
</tr>
<tr>
<td>SESSION segment</td>
<td>CONVSEC</td>
<td>racfSessionConvSec</td>
</tr>
<tr>
<td>SESSION segment</td>
<td>INTERVAL</td>
<td>racfSessionInterval</td>
</tr>
<tr>
<td>SESSION segment</td>
<td>LOCK</td>
<td>racfSessionLock</td>
</tr>
<tr>
<td>SESSION segment</td>
<td>SESSKEY</td>
<td>racfSessionSessKey</td>
</tr>
<tr>
<td>SIGVER segment</td>
<td>FAILLOAD</td>
<td>racfSigverFailLoad</td>
</tr>
<tr>
<td>SIGVER segment</td>
<td>SIGAUDIT</td>
<td>racfSigverSigAudit</td>
</tr>
<tr>
<td>SIGVER segment</td>
<td>SIGREQUIRED</td>
<td>racfSigverSigRequired</td>
</tr>
<tr>
<td>SSIGNON segment</td>
<td>KEYENCRYPTED - value is not displayed by LDAP</td>
<td>racfSsignonKeyEncrypted</td>
</tr>
<tr>
<td>SSIGNON segment</td>
<td>KEYMASKED - value is not displayed by LDAP</td>
<td>racfSsignonKeyMasked</td>
</tr>
<tr>
<td>STDATA segment</td>
<td>GROUP</td>
<td>racfStdataGroup</td>
</tr>
<tr>
<td>STDATA segment</td>
<td>PRIVILEGED</td>
<td>racfStdataPrivileged</td>
</tr>
<tr>
<td>STDATA segment</td>
<td>TRACE</td>
<td>racfStdataTrace</td>
</tr>
<tr>
<td>STDATA segment</td>
<td>TRUSTED</td>
<td>racfStdataTrusted</td>
</tr>
<tr>
<td>STDATA segment</td>
<td>USER</td>
<td>racfStdataUser</td>
</tr>
</tbody>
</table>

Table 41. Mapping of LDAP attribute names to RACF fixed fields (setropts)

<table>
<thead>
<tr>
<th>RACF segment name</th>
<th>RACF keyword in setropts</th>
<th>LDAP attribute name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setropts base</td>
<td>Multi-value: REFRESH, WHEN(PROGRAM)</td>
<td>racfSetroptsAttributes</td>
</tr>
<tr>
<td>Setropts base</td>
<td>AUDIT</td>
<td>racfAudit</td>
</tr>
<tr>
<td>Setropts base</td>
<td>CLASSACT</td>
<td>racfClassAct</td>
</tr>
<tr>
<td>Setropts base</td>
<td>GENCMD</td>
<td>racfGenCmd</td>
</tr>
<tr>
<td>Setropts base</td>
<td>GENERIC</td>
<td>racfGeneric</td>
</tr>
<tr>
<td>Setropts base</td>
<td>GENLIST</td>
<td>racfGenList</td>
</tr>
<tr>
<td>Setropts base</td>
<td>GLOBAL</td>
<td>racfGlobal</td>
</tr>
</tbody>
</table>
Table 41. Mapping of LDAP attribute names to RACF fixed fields (setropts) (continued)

<table>
<thead>
<tr>
<th>RACF segment name</th>
<th>RACF keyword in setropts</th>
<th>LDAP attribute name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setropts base</td>
<td>LOGOPTIONS(ALWAYS)</td>
<td>racfLogOptionsAlways</td>
</tr>
<tr>
<td>Setropts base</td>
<td>LOGOPTIONS(DEFAULT)</td>
<td>racfLogOptionsDefault</td>
</tr>
<tr>
<td>Setropts base</td>
<td>LOGOPTIONS(FAILURES)</td>
<td>racfLogOptionsFailures</td>
</tr>
<tr>
<td>Setropts base</td>
<td>LOGOPTIONS(NEVER)</td>
<td>racfLogOptionsNever</td>
</tr>
<tr>
<td>Setropts base</td>
<td>LOGOPTIONS(SUCCESSES)</td>
<td>racfLogOptionsSuccesses</td>
</tr>
<tr>
<td>Setropts base</td>
<td>RACLIST</td>
<td>racfRacList</td>
</tr>
<tr>
<td>Setropts base</td>
<td>STATISTICS</td>
<td>racfStatistics</td>
</tr>
</tbody>
</table>

## Associating LDAP attributes to RACF custom fields

The user and group profile custom fields in the CSDATA segment are not predefined by RACF, but are defined in RACF by the user. If those fields are to be set and displayed using LDAP, then the user must add an attribute to the LDAP schema to represent each custom field. The attribute is defined in the LDAP schema using an `attributeTypes` value and an `IBMAttributeTypes` value. LDAP does not allow more than one attribute to be associated with the same RACF custom field.

- The `attributeTypes` value specifies the object identifier (OID), name, syntax, and equality rule of the attribute. The OID and name must not be in use in the schema. The attribute name can be the same as the RACF custom field name, or it can be different. For example, if the custom field name is `phone`, the attribute name could be `phone`, `workphone`, or anything else that is not already in use. See [Attribute types](#) for more information about `attributeTypes`.

- The `IBMAttributeTypes` value must include the `RACFFIELD` keyword to identify the RACF custom field associated with the attribute. The `RACFFIELD` value specifies the resource profile name that is used to define the custom field in RACF, with periods (.) changed to dashes (-). For example, if the RACF custom field is defined by the `USER.CSDATA.PHONE` resource profile, `RACFFIELD` contains `USER-CSDATA-PHONE`. `RACFFIELD` also optionally specifies the type of RACF custom field. The accepted values are `char`, `flag`, `hex`, `num`, and `qchar`. If the RACF custom field is defined with `TYPE(CHAR)` `FIRST(ANY)` `OTHER(ANY)`, then specify `qchar` in `RACFFIELD` to indicate that SDBM should put the attribute value in quotes when creating RACF commands. Otherwise, specify in `RACFFIELD` the same value as was used for `TYPE` when defining the custom field in RACF. If a type value is not specified in `RACFFIELD`, LDAP assumes the custom field type is `char`. See [IBMAttributeTypes](#) for more information about `IBMAttributeTypes`.

For example, if the `phone` custom field is defined in the RACF user profile with `TYPE(CHAR)`, the following attribute could be added to the LDAP schema to represent the custom field:

```plaintext
attributetypes: {
    phone-OID
    NAME 'phone',
    DESC 'Represents the PHONE field in the RACF user CSDATA segment'
    EQUALITY caseIgnoreMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
    SINGLE-VALUE
    USAGE userApplications
}
```

```plaintext
ibmattributetypes: {
    phone-OID
    ACCESS-CLASS sensitive
    RACFFIELD ('USER-CSDATA-PHONE' 'char')
}
```

## Notes:

1. A numeric OID can be used instead of the nonnumeric OID `phone-OID`. A numeric OID must be used if the LDAP server is sharing a TDBM database with an Integrated Security Services LDAP server on z/OS V1R10 or earlier releases.
2. If the RACF custom field is defined to be case-sensitive (using MIXED(YES)), change the EQUALITY rule to EQUALITY caseExactMatch. Otherwise, compare operations can fail if mixed case values are involved.

3. The SYNTAX must be IA5 String (1.3.6.1.4.1.1466.115.121.1.26).

4. All RACF custom fields have only a single value, therefore, SINGLE-VALUE is specified.

5. The ACCESS-CLASS is 'sensitive' for most RACF attributes, but can be changed to 'critical' if the field contains data to which access is more restrictive. SDBM does not use the ACCESS-CLASS value, but TDBM, LDBM, and CDBM do.

For completeness, add an object class to the LDAP schema to represent the CSDATA segment in each profile. SDBM always assumes that the object class names are racfUserCsdataSegment for the CSDATA segment in the user profile or racfGroupCsdataSegment for the CSDATA segment in the group profile. SDBM adds this object class to a user or group entry if the corresponding RACF profile contains the CSDATA segment.

For example, if the PHONE and SSN custom fields are defined in RACF for the user profile and the LDAP attributes phone and socialSecurityNumber are defined in the LDAP schema to represent the custom fields, the following object class should be added to the LDAP schema:

objectclasses: ( 
    racfUserCsdataSegment-OID
    NAME 'racfUserCsdataSegment'
    DESC 'Represents the CSDATA segment in a z/OS RACF USER profile'
    SUP top
    AUXILIARY
    MAY ( phone $ socialSecurityNumber )
)

Note: A numeric OID can be used instead of the nonnumeric OID racfUserCsdataSegment-OID. A numeric OID must be used if the LDAP server is sharing a TDBM database with an Integrated Security Services LDAP server on z/OS V1R10 or earlier releases.

Special usage of racfAttributes, racfConnectAttributes, racfResourceAttributes, and racfSetroptsAttributes

The racfAttributes attribute is a multi-valued attribute that can be used to specify any single-word keywords that can be specified on a RACF adduser or altuser command. For example, racfAttributes can be used to add a RACF user entry with 'ADSP GRPACC NOPASSWORD' or modify a RACF user entry with 'NOGRPACC SPECIAL NOEXPIRED RESUME NOOMVS'. Additional values, such as PASSWORD, can be returned in racfAttributes that are not returned by the listuser command.

Similarly, racfConnectAttributes can be used to specify any single-word keywords that are valid on a RACF connect command, as can racfResourceAttributes for the RACF define andalter commands. racfSetroptsAttributes can be used for the RACF setropts command, but only those values listed in Table 41 on page 278 can be specified.

RACF namespace entries

When the SDBM backend is used to make RACF information accessible over the LDAP protocol, SDBM creates a set of top entries to set up a hierarchical representation of RACF users, groups, connections, classes, resources, and class options. These top entries consist of the suffix, top user entry, top group entry, top connection entry, a top entry for each RACF class (except DATASET, which is not supported), and a setropts entry. For example, the top entries in Figure 23 and Figure 24 are:

- cn=RACFA,o=IBM,c=US (suffix entry)
- profileType=User,cn=RACFA,o=IBM,c=US (top user entry)
- profileType=Group,cn=RACFA,o=IBM,c=US (top group entry)
The top entries cannot be added or deleted. With the exception for the setropts entry, the top entries can only be compared and searched.

The setropts entry can be modified, compared, and searched.

The value used for the suffix entry DN is the value specified for the `suffix` option in the SDBM section of the LDAP server configuration file (see Setting up for SDBM).

Following is a high-level diagram of the RACF backend.

**Figure 23. RACF namespace hierarchy (Part 1 of 2)**

**Figure 24. RACF namespace hierarchy (Part 2 of 2)**
SDBM schema information
The attributes and object classes used by SDBM to represent RACF values are always in the LDAP server schema, with the exception of any attributes needed for RACF custom fields.

SDBM support for special characters
- An SDBM DN, including the SDBM suffix, can contain the following special characters:
  - A plus sign (+), double quote ("), or backslash (\) anywhere in a DN.
  - A pound sign (#) at the beginning of a value in a DN.
- When present in a DN, a special character must be escaped by preceding it with a single backslash (\).
- Note that the suffix in the LDAP server configuration file must use two back slashes (\) to escape a special character, but only a single backslash is used in a DN.

For example, if the SDBM suffix in the configuration file is

```
suffix cn=\#plex\#1
```

then the DN for the RACF resource profile a+b in class #x#y would be

```
profilename=a\+b,profiletype=\#x#y,cn=\#plex\#1
```

- Special characters in a DN returned by SDBM are always escaped by a single backslash. Pound signs that are not at the beginning of a value and equal signs (=) might be escaped, depending on the usage of the DN.

- When specifying a value containing a special character for an attribute within an add or modify request, escape the special character with a back slash if the attribute is part of a DN, otherwise, do not escape the special character. For instance, to add a user with the default group #dlgrp, specify either:

```
racfdefaultgroup: racfid=\#dlgrp,profiletype=group,cn=\#plex\#1
```

or

```
racfdefaultgroup: #dlgrp
```

within the entry.

- When specifying a value containing a special character for an attribute within a search filter, the special character can be escaped or not. For instance, to search for all RACF users starting with #user, use the search filter `racfid=#user*` or `racfid=\#user*`.

Control of access to RACF data
As explained above, SDBM operations result in issuing RACF commands. Table 42 and Table 43 indicate which commands are issued for various SDBM operations. The RACF commands are issued under the context of the RACF user ID that has bound to SDBM. RACF determines the results of the RACF commands based on the RACF authority of that user ID. If the RACF command fails, the SDBM operation fails and returns any error information issued by RACF.

In particular, the RACF search command can fail because of lack of authority, even if the bound user is able to extract RACF data from user IDs that match the RACF search. In this case, SDBM searches that result in issuing a RACF search command fail and return:

```
ldap_search: Unknown error
```

```
ldap_search: additional info: ICH31005I NO ENTRIES MEET SEARCH CRITERIA
```

SDBM operational behavior
Table 42 shows how SDBM behaves during different LDAP operations.
Table 42. RACF backend behavior

<table>
<thead>
<tr>
<th>Target DN</th>
<th>LDAP operation behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>suffixDN</td>
<td></td>
</tr>
<tr>
<td>Add</td>
<td>Error: Unwilling to perform</td>
</tr>
<tr>
<td>Modify</td>
<td>Error: Unwilling to perform</td>
</tr>
<tr>
<td>Delete</td>
<td>Error: Unwilling to perform</td>
</tr>
<tr>
<td>Modify DN</td>
<td>Error: Unwilling to perform</td>
</tr>
<tr>
<td>Compare</td>
<td>Compare attribute</td>
</tr>
<tr>
<td>Search base</td>
<td>Return requested attributes</td>
</tr>
<tr>
<td>Search one level</td>
<td>Perform a base search against each subordinate of this entry</td>
</tr>
<tr>
<td>Search subtree</td>
<td>See <a href="#">Searching the entire RACF database</a></td>
</tr>
<tr>
<td>Bind</td>
<td>Error: No credentials</td>
</tr>
<tr>
<td>profiletype=User.suffixDN</td>
<td></td>
</tr>
<tr>
<td>Add</td>
<td>Error: Unwilling to perform</td>
</tr>
<tr>
<td>Modify</td>
<td>Error: Unwilling to perform</td>
</tr>
<tr>
<td>Delete</td>
<td>Error: Unwilling to perform</td>
</tr>
<tr>
<td>Modify DN</td>
<td>Error: Unwilling to perform</td>
</tr>
<tr>
<td>Compare</td>
<td>Compare attribute</td>
</tr>
<tr>
<td>Search base</td>
<td>Return requested attributes</td>
</tr>
<tr>
<td>Search one level</td>
<td>See <a href="#">Searching the entire RACF database</a></td>
</tr>
<tr>
<td>Search subtree</td>
<td>See <a href="#">Searching the entire RACF database</a></td>
</tr>
<tr>
<td>Bind</td>
<td>Error: No credentials</td>
</tr>
<tr>
<td>profiletype=Group.suffixDN</td>
<td></td>
</tr>
<tr>
<td>Add</td>
<td>Error: Unwilling to perform</td>
</tr>
<tr>
<td>Modify</td>
<td>Error: Unwilling to perform</td>
</tr>
<tr>
<td>Delete</td>
<td>Error: Unwilling to perform</td>
</tr>
<tr>
<td>Modify DN</td>
<td>Error: Unwilling to perform</td>
</tr>
<tr>
<td>Compare</td>
<td>Compare attribute</td>
</tr>
<tr>
<td>Search base</td>
<td>Return requested attributes</td>
</tr>
<tr>
<td>Search one level</td>
<td>See <a href="#">Searching the entire RACF database</a></td>
</tr>
<tr>
<td>Search subtree</td>
<td>See <a href="#">Searching the entire RACF database</a></td>
</tr>
<tr>
<td>Bind</td>
<td>Error: No credentials</td>
</tr>
<tr>
<td>Target DN</td>
<td>LDAP operation behavior</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| profiletype=Facility,suffixDN | **Add** Error: Unwilling to perform  
Modify  Error: Unwilling to perform  
Delete  Error: Unwilling to perform  
Modify DN  Error: Unwilling to perform  
Compare  Compare attribute  
Search base  Return requested attributes  
Search one level  See [Searching the entire RACF database](#)  
Search subtree  See [Searching the entire RACF database](#)  
Bind  Error: No credentials |
| cn=setropts,suffixDN | **Add** Error: Unwilling to perform  
Modify  Perform a setropts RACF command  
Delete  Error: Unwilling to perform  
Modify DN  Error: Unwilling to perform  
Compare  Compare attribute  
Search base  Perform an R_admin setropts extract RACF command  
Search one level  Empty search results (this is a leaf node in the hierarchy)  
Search subtree  Perform an R_admin setropts extract RACF command  
Bind  Error: No credentials |
| profiletype=Connect,suffixDN | **Add** Error: Unwilling to perform  
Modify  Error: Unwilling to perform  
Delete  Error: Unwilling to perform  
Modify DN  Error: Unwilling to perform  
Compare  Compare attribute  
Search base  Return requested attributes  
Search one level  See [Searching the entire RACF database](#)  
Search subtree  See [Searching the entire RACF database](#)  
Bind  Error: No credentials |
<table>
<thead>
<tr>
<th>Target DN</th>
<th>LDAP operation behavior</th>
</tr>
</thead>
</table>
| racfid=XYZ111,profiletype=User, suffixDN | **Add**  Perform an **adduser** RACF command using USER=XYZ111  
**Modify**  Perform an **altuser** RACF command using USER=XYZ111  
**Delete**  Perform a **deluser** RACF command using USER=XYZ111  
**Modify DN**  
**Compare**  Compare requested attribute with data returned from an  
R_admin profile extract RACF command using USER=XYZ111  
**Search base**  Perform an R_admin profile extract RACF command using  
USER=XYZ111  
**Search one level**  
Empty search results (this is a leaf node in the hierarchy)  
**Search subtree**  Perform an R_admin profile extract RACF command using  
USER=XYZ111  
**Bind**  If bind type is not simple, error: Unwilling to perform  
else use **_passwd()** to verify the user ID and password or  
password phrase combination and then perform an R_admin  
profile extract RACF command using USER=XYZ111 if  
gathering group membership |

| racfid=GRP222,profiletype=Group, suffixDN | **Add**  Perform an **addgroup** RACF command using  
GROUP=GRP222  
**Modify**  Perform an **altgroup** RACF command using GROUP=GRP222  
**Delete**  Perform a **delgroup** RACF command using GROUP=GRP222  
**Modify DN**  
**Compare**  Compare requested attribute with data returned from an  
R_admin profile extract RACF command using  
GROUP=GRP222  
**Search base**  Perform an R_admin profile extract RACF command using  
GROUP=GRP222  
**Search one level**  
Empty search results (this is a leaf node in the hierarchy)  
**Search subtree**  Perform an R_admin profile extract RACF command using  
GROUP=GRP222  
**Bind**  Error: No credentials |
### Table 42. RACF backend behavior (continued)

<table>
<thead>
<tr>
<th>Target DN</th>
<th>LDAP operation behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>racuserid=XYZ111+racgroupid=GRP222, profiletype=Connect, suffixDN</td>
<td><strong>Add</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Modify</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Delete</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Modify DN</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Compare</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Search base</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Search one level</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Search subtree</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Bind</strong></td>
</tr>
<tr>
<td>profilename=ABC.RES, profiletype=Facility, suffixDN</td>
<td><strong>Add</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Modify</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Delete</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Modify DN</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Compare</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Search base</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Search one level</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Search subtree</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Bind</strong></td>
</tr>
</tbody>
</table>
If LDAP is running with an SDBM backend, the `ldap_modify` and `ldap_add` APIs can return `LDAP_OTHER` or `LDAP_SUCCESS` and have completed a partial update to an entry in RACF. The results will match what occurs if the update were done using the RACF `altuser`, `altgroup`, `connect`, `ralter`, and `permit` commands. If several RACF attributes are being updated and one of them is in error, RACF might still update the other attributes, without, in some cases, returning an error message. If there is a RACF message, LDAP always returns it in the result.

This is further complicated when adding or modifying a general resource profile because this can involve multiple RACF commands: a `rdefine` or `ralter` command followed by one or more `permit` commands. If one of the commands fails, processing ends but the resource profile is still updated with the results of the prior successful commands.

The RACF `connect` command is used to both add a user connection to a group and to modify a user's connection to a group. As a result, the SDBM add and modify support for connection entries is different than normal LDAP support:

- When adding a connection entry that already exists, the entry is modified using the specified attributes. There is no indication returned that the entry already existed.
- When modifying a connection entry that does not exist, the entry is added using the specified attributes. There is no indication returned that the entry did not exist.

**Notes about specifying attribute values:**

1. In LDAP, the format of the value of the Kerberos principal name attribute, `krbPrincipalName`, is `userid@realm`. In SDBM, the `userid` portion of the name is case-sensitive while the `realm` portion of the name is not. However, the entire attribute value is processed as case-sensitive in a compare operation. In addition, SDBM only operates on the RACF local realm. If the realm specified in the value is not the local realm, the operation fails. For example, when searching in SDBM for a user entry using the search filter `krbPrincipalName=krbuser1@myrealm.com`, the search fails if `myrealm.com` is not the RACF local realm.

To facilitate using the `krbPrincipalName` attribute, the attribute value can be specified without the `@realm` portion. In this case, the realm is assumed to be the RACF local realm. For example, when adding a user entry with `krbuser1` as the `userid` portion of that Kerberos principal name, the `krbPrincipalName` attribute can be specified as:

```
krbPrincipalName: krbuser1
```

or

```
krbPrincipalName: krbuser1@myrealm.com
```

where `myrealm.com` is the RACF local realm.

The `krbPrincipalName` value returned by SDBM from a search is always the complete principal name, `userid@realm`, where `realm` is the RACF local realm.

2. There are several SDBM attributes whose value is a RACF user, group, or class name. For convenience, this value can be specified either as just the RACF name or as the complete LDAP DN. For example, when adding a user with a default group of `grp222`, the `racfDefaultGroup` attribute can be specified as:

```
racfDefaultGroup: grp222
```

or

```
racfDefaultGroup: racfid=grp222,profiletype=group,syplex=myplex
```

where `syplex=myplex` is the SDBM suffix.

The value returned by SDBM from a search is always the complete LDAP DN.

3. For multi-value attributes, the RACF `altuser` and `ralter` commands do not always support the ability to both add a value and replace the existing value. As a result, SDBM does not always respect the type of modification (add versus replace) that is specified in a modify command.
Values for the following multi-value attributes are always added to the existing value (even if replace is specified): racfAttributes, racfAudit, racfClassAct, racfClassName, racfConnectAttributes, racfGenCmd, racfGeneric, racfGenList, racfGlobal, racfLevelKeyword, racfLogonDays, racfLogOptionsAlways, racfLogOptionsDefault, racfLogOptionsFailures, racfLogOptionsNever, racfLogOptionsSuccessесс, racfMemberList, racfMemberList, racfMformKeyword, racfMonitorKeyword, racfRacList, racfResourceAttributes, racfSecurityCategoryList, racfSetroptsAttributes, racfStatistics, racfVolumeList.

Values for the following multi-value attributes always replace the existing value (even if add is specified): racfCdtinfoFirst, racfCdtinfoOther, racfDlfdataJobNames, racfDomains, racfCtscSymExportCerts, racfCtscSymExportKeys, racfMsceSystems, racfNetviewOperatorClass, racfOperatorClass, racfResourceAudit, racfResourceGlobalAudit, racfRoutcodeKeyword, racfRsiKey, racfTsiKey.

Values for the following multi-value attributes either are added to the existing values or replace the existing values, depending on the new and existing values: racfAuthKeyword, racfAccessControl, and racfCtscAsymUsage.

For single-value attributes, there is no difference between using an add modification or a replace modification to set the value. For either type of modification, the value is added if the attribute value does not exist and the value replaces the existing attribute value, if there is one.

4. In order to update CICS-related attributes, CICS must be set up on your system; otherwise, errors result.

5. For modify, if a request is made to delete a specific attribute value for an attribute where specific values cannot be selectively deleted, an \texttt{LDAP\_UNWILLING\_TO\_PERFORM} error code is returned. Similarly, if a request is made to delete the entire attribute for an attribute where specific values to delete must be specified, an \texttt{LDAP\_UNWILLING\_TO\_PERFORM} error code is returned.

The following attributes require that specific values to delete be specified: racfAudit, racfClassAct, racfClassName, racfGenCmd, racfGeneric, racfGenList, racfGlobal, racfMemberList, racfRacList, racfStatistics, racfVolumeList.

The following attributes allow specifying specific values to delete but also support deleting the entire attribute: racfAccessControl, racfAttributes, racfConnectAttributes, racfResourceAttributes, racfSecurityCategoryList, racfSetroptsAttributes.

All other attributes that have a delete command in RACF only allow deleting the entire attribute. If an attempt is made to delete any attribute that has no corresponding delete command in RACF, an \texttt{LDAP\_UNWILLING\_TO\_PERFORM} error code is returned.

6. The \texttt{racfCopyProfileFrom} attribute is used to specify any combination of the RACF \texttt{rdefine} FCLASS, FGENERIC, FROM, and FVOLUME keywords and values to indicate a resource profile to use as a model when creating a new resource profile. The value specified for this attribute must be syntactically correct for an \texttt{rdefine} command and is inserted as is in the command. For example, the following uses the RES.MODE resource profile in the FACILITY class as a model:

\texttt{racfcopyprofilefrom: FROM(RES.MODE) FCLASS(FACILITY)}

7. The \texttt{racfAccessControl} attribute is used to manage the access control lists for a general resource profile. Each attribute value is used to create a separate RACF \texttt{permit} command. Each value must be a syntactically-correct RACF \texttt{permit} command, without the class and profile names. SDBM adds the class and profile names before issuing the RACF \texttt{permit} command. It also adds the \texttt{DELETE} keyword for a modify request to delete the value.

\textbf{Note}: When issuing multiple RACF \texttt{permit} commands for the same resource, the order of the \texttt{permit} commands can be critical. SDBM issues a \texttt{permit} command for each \texttt{racfAccessControl} value in the order that SDBM receives the values. Be aware that if you specify multiple add, replace, and delete changes to an attribute in a single modify operation, many \texttt{ldapmodify} utilities (including the z/OS client \texttt{ldapmodify} utility) may reorder the changes to put all the changes of the same type together. Therefore, the values as presented to SDBM might not be in the original order and the results of the \texttt{permit} commands might not be as desired. To avoid this, separate different \texttt{racfAccessControl} attribute changes into separate modify operations.
When LDAP returns the `racfAccessControl` value during a search operation, the value might contain the COUNT field if this is part of the RACF output, for example:

```
racfaccesscontrol: ID(X) ACCESS(READ) COUNT(5)
```

If SDBM finds the COUNT field in a `racfAccessControl` value during an add or modify operation, the field is removed from the value before the value is used to generate a RACF permit command. This allows LDAP search output to be used as add or modify input.

When using the `racfAccessControl` attribute in a compare operation, the comparison is done only on the value specified for the ID keyword within the attribute value. The rest of the attribute value is not used. If the ID value is contained in any access control list within the resource profile, the compare returns LDAP_COMPARE_TRUE. If the attribute value does not have the ID keyword, has more than one ID value, or the value is not contained in any access list within the resource profile, compare returns LDAP_COMPARE_FALSE. Basically, a `racfAccessControl` compare operation can be used to determine if a specific RACF user or group appears in the access control lists within a resource profile.

### SDBM search capabilities

SDBM supports a limited set of search filters. The following table describes each supported filter and indicates from what bases it is valid, what sort of entries it returns (a complete entry or entries that just contain the DN of the entry), and what RACF commands are issued to perform the search. Most searches can only be performed from one of these top entries: the `suffix` entry, the `profiletype=user,suffix` entry, the `profiletype=group,suffix` entry, the `profiletype=connect,suffix` entry, and the `profiletype=class,suffix` entries.

<table>
<thead>
<tr>
<th>Filter</th>
<th>Search behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>krbprincipalname=any_value</code></td>
<td><strong>Description:</strong> find user profile for the RACF user whose KERB KERBNAME value is <code>any_value</code></td>
</tr>
<tr>
<td></td>
<td><strong>Allowed base:</strong> <code>suffix</code></td>
</tr>
<tr>
<td></td>
<td><code>profiletype=group,suffix</code></td>
</tr>
<tr>
<td></td>
<td><strong>Returns:</strong> complete entry</td>
</tr>
<tr>
<td></td>
<td><strong>Commands:</strong></td>
</tr>
<tr>
<td></td>
<td>- <code>R_usermap</code></td>
</tr>
<tr>
<td></td>
<td>- followed by <code>R_admin</code> user profile extract</td>
</tr>
<tr>
<td>Filter</td>
<td>Search behavior</td>
</tr>
<tr>
<td>----------------</td>
<td>---------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>objectclass=*</td>
<td><strong>Description:</strong> match any user, group, connection, resource profile, and setropts</td>
</tr>
<tr>
<td></td>
<td><strong>Allowed base:</strong> any SDBM entry</td>
</tr>
<tr>
<td></td>
<td><strong>Returns:</strong></td>
</tr>
<tr>
<td></td>
<td>• DN-only entries if scope includes all users, groups, connections, resource profiles, or setropts</td>
</tr>
<tr>
<td></td>
<td>• Complete entry if scope includes a single entry</td>
</tr>
<tr>
<td></td>
<td><strong>Commands:</strong></td>
</tr>
<tr>
<td></td>
<td>• if scope includes all users:</td>
</tr>
<tr>
<td></td>
<td>search class(user) filter(*)</td>
</tr>
<tr>
<td></td>
<td>• if scope includes all groups:</td>
</tr>
<tr>
<td></td>
<td>search class(group) filter(*)</td>
</tr>
<tr>
<td></td>
<td>• if scope includes all connections:</td>
</tr>
<tr>
<td></td>
<td>search class(group) filter(*)</td>
</tr>
<tr>
<td></td>
<td>• if scope includes all classes:</td>
</tr>
<tr>
<td></td>
<td>RACROUTE STAT to retrieve all class names</td>
</tr>
<tr>
<td></td>
<td>followed by search class(className) filter(“”) for each class</td>
</tr>
<tr>
<td></td>
<td>• if scope includes a specific class:</td>
</tr>
<tr>
<td></td>
<td>RACROUTE STAT to determine if the class exists</td>
</tr>
<tr>
<td></td>
<td>followed by search class(className) filter(“”) for the class</td>
</tr>
<tr>
<td></td>
<td>• if scope includes a single user:</td>
</tr>
<tr>
<td></td>
<td>R_admin user profile extract</td>
</tr>
<tr>
<td></td>
<td>• if scope includes a single group:</td>
</tr>
<tr>
<td></td>
<td>R_admin group profile extract</td>
</tr>
<tr>
<td></td>
<td>• if scope includes a single connection:</td>
</tr>
<tr>
<td></td>
<td>R_admin connect profile extract</td>
</tr>
<tr>
<td></td>
<td>• if scope includes a single resource:</td>
</tr>
<tr>
<td></td>
<td>R_admin resource profile extract</td>
</tr>
<tr>
<td></td>
<td>• if scope includes just the cn=setropts entry:</td>
</tr>
<tr>
<td></td>
<td>R_admin setropts extract</td>
</tr>
</tbody>
</table>
### Table 43. SDBM search filters (continued)

<table>
<thead>
<tr>
<th>Filter</th>
<th>Search behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>profilename=any_value</code></td>
<td><strong>Description:</strong> find the RACF general resource profiles whose names match any_value (can contain wildcards)  &lt;br&gt; <strong>Note:</strong> RACF profile names might be case-sensitive, depending on the class.  &lt;br&gt; <strong>Allowed base:</strong> suffix  &lt;br&gt; <code>profiletype=className,suffix</code>  &lt;br&gt; <strong>Returns:</strong> DN-only entries  &lt;br&gt; <strong>Commands:</strong>  &lt;br&gt; • if scope includes all classes:  &lt;br&gt; - <code>RACROUTE STAT</code> to retrieve all class names  &lt;br&gt; - followed by <code>search class(className) filter(any_value)</code> for each class  &lt;br&gt; • if scope includes a single class:  &lt;br&gt; - <code>RACROUTE STAT</code> to determine if the class exists  &lt;br&gt; - followed by <code>search class(className) filter(any_value)</code> for the class</td>
</tr>
<tr>
<td><code>racfgroupid=any_value</code></td>
<td><strong>Description:</strong> find connection profiles for members of the RACF groups whose names match any_value (can contain wildcards)  &lt;br&gt; <strong>Allowed base:</strong> suffix  &lt;br&gt; <code>profiletype=connect,suffix</code>  &lt;br&gt; <strong>Returns:</strong> DN-only entries  &lt;br&gt; <strong>Commands:</strong>  &lt;br&gt; • if no wildcard in any_value:  &lt;br&gt;   - <code>R_admin group profile extract</code>  &lt;br&gt; • if wildcard in any_value:  &lt;br&gt;   - <code>search class(group) filter(any_value)</code>  &lt;br&gt;   - followed by <code>R_admin group profile extract</code> for each group</td>
</tr>
</tbody>
</table>
Table 43. SDBM search filters (continued)

<table>
<thead>
<tr>
<th>Filter</th>
<th>Search behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>racfid=any_value</td>
<td><strong>Description:</strong> find user and group profiles for the RACF users and groups whose names match <em>any_value</em> (can contain wildcards)</td>
</tr>
</tbody>
</table>
|                         | **Allowed base:**  
|                         |     suffix           |  
|                         |     profiletype=user, suffix           |  
|                         |     profiletype=group, suffix           |  
|                         | **Returns:** DN-only entries                     |  
|                         | **Commands:**  
|                         |     if scope includes all users:  
|                         |         search class(user) filter(*any_value*) |  
|                         |     if scope includes all groups:  
|                         |         search class(group) filter(*any_value*) |  
| racfnosshortname=any_value | **Description:** find user profile for the RACF user whose LNOTES SNAME value is *any_value* |  
|                         | **Allowed base:**  
|                         |     suffix           |  
|                         |     profiletype=user, suffix           |  
|                         | **Returns:** complete entry                     |  
|                         | **Commands:**  
|                         |     R_usermap                                  |  
|                         |     – followed by R_admin user profile extract |  
| racfndsusername=any_value | **Description:** find user profile for the RACF user whose NDS UNAME value is *any_value* |  
|                         | **Allowed base:**  
|                         |     suffix           |  
|                         |     profiletype=user, suffix           |  
|                         | **Returns:** complete entry                     |  
|                         | **Commands:**  
|                         |     R_usermap                                  |  
|                         |     – followed by R_admin user profile extract |  
|                         |
Table 43. SDBM search filters (continued)

<table>
<thead>
<tr>
<th>Filter</th>
<th>Search behavior</th>
</tr>
</thead>
</table>
| racfomvsgroupid=number | **Description:** find group profile for one of the RACF groups whose OMVS GID values match number  
**Allowed base:**  
suffix  
profiletype=group,suffix  
**Returns:** complete entry  
**Commands:**  
– getgрид(number)  
– followed by R_admin group profile extract |
| racfomvsgroupid;allOMVSids=number | **Description:** find group profiles for all the RACF groups whose OMVS GID values match number  
**Allowed base:**  
suffix  
profiletype=group,suffix  
**Returns:** DN-only entries  
**Commands:**  
search class(group) gid(number) |
| racfomvsuid=number | **Description:** find user profile for one of the RACF users whose OMVS UID values match number  
**Allowed base:**  
suffix  
profiletype=user,suffix  
**Returns:** complete entry  
**Commands:**  
– getpwuid(number)  
– followed by R_admin user profile extract |
<table>
<thead>
<tr>
<th>Filter</th>
<th>Search behavior</th>
</tr>
</thead>
</table>
| racfomvsuid; allOMVSid=number              | **Description:**
find user profiles for all the RACF users whose OMVS UID values match *number*

**Allowed base:**
suffix
  profltype=user,suffix

**Returns:**
DN-only entries

**Commands:**
search class(user) uid(*number*)

| racfuserid=*any_value*                  | **Description:**
find connection profiles for RACF users whose names match *any_value* (can contain wildcards)

**Allowed base:**
suffix
  profltype=connect,suffix

**Returns:**
DN-only entries

**Commands:**
- if no wildcard in *any_value:*
  
  R_admin user profile extract

- if wildcard in *any_value*
  
  - search class(user) filter(*any_value*)
  
  - followed by R_admin user profile extract for each user

| *(racfuserid=*any_value1*) *(racfgroupid=*any_value2)* | **Description:**
find connection profiles for RACF users whose names match *any_value1* and who belong to RACF groups whose names match *any_value2* (both can contain wildcards)

**Allowed base:**
suffix
  profltype=connect,suffix

**Returns:**
DN-only entries

**Commands:**
- if no wildcard in *any_value1:*
  
  R_admin user profile extract

- if no wildcard in *any_value2*
  
  R_admin group profile extract

- if wildcard in both *any_value1* and *any_value2*
  
  - search class(group) filter(*any_value2*)
  
  - followed by R_admin group profile extract for each group
Except for the AND filter for connections, complex search filters that include NOT, AND, OR, LE, or GE constructs are not supported.

The values for the **profilename**, **racfgroupid**, **racfid**, and **racfuserid** filters can include the wildcards supported by RACF. These wildcards are '*' which represents any number of characters, and '%' which represents one character. For example:

```
(&(racfuserid=usr*)(racfgroupid=*grp))
```

searches for all the connections between users whose names begin with *usr* and groups whose names end with *grp*.

To include multiple levels of qualifiers in a resource profile name search, include either \\** or *\* in the **profilename** filter. For example, **profilename=XYZ.\*** searches for all resource profiles that have XYZ as the first qualifier. Do not use ** in the filter because this is not a valid LDAP filter. The result of a search with the filter **profilename=*** is:

```
ldap_search: Protocol error
ldap_search: additional info: R010043 Substring filter for attribute 'profilename' has no value
```

Although an '*' or '**' can be part of a resource profile name, there is no way to indicate in the **profilename** filter that an asterisk or double asterisk is part of the name rather than a wildcard. For example, a search using a filter such as **profilename=ABC* returns all profile names beginning with ABC, including the ABC* profile (if it exists).

**Note about searching universal groups:** Most of the members of a RACF universal group are not actually contained in the group's list of members. As a result, a search of the entry for a universal group does not return most of the group's members. In addition, a search for the connection entry corresponding to a member of a universal group can return different results depending on the connection search filter that is used:

- If the **racfuserid** part of the connection search filter does not contain a wild card, then the connection entry is returned for the specified **racfuserid**.
- If the **racfuserid** part of the connection search filter contains a wild card, then the connection entry for a user is returned only if the user is explicitly contained in the universal group's list of members.

**Searching the entire RACF database**

Most searches that query the entire RACF database, for example, a subtree search from any of the top directory entries except the setopts entry, return only the DN (distinguished name) attribute. You may then obtain more specific data about a particular user, group, connection, or resource on a follow-up search using a specific DN as the search base.

The exceptions to this are searches using the “application ID” filters:

```
krbprincipalname=<any_name>
racflnotesshortname=<any_value>
racfndsusername=<any_value>
racfomvspsgroupid=<number>
racfomvspsuid=<number>
```

Because these searches can match only a single RACF user, the entire user entry is returned in the search results.

**RACF restriction on amount of output:** When processing certain LDAP search requests, SDBM uses the RACF **R_admin** "run command" interface to issue RACF **search** commands. The **R_admin** "run command" interface limits the number of records in its output to 4096. This means that the RACF **search** command output might be incomplete if you have a large number of users, groups, connections, or resources. See [z/OS Security Server RACF Callable Services](https://www.ibm.com/support/knowledgecenter/SSTJTP_7.2.0/com.ibm.zos.racf.zosics.doc/racf CallableServices66.html) on the RACF restriction. The restriction only affects those SDBM searches that issue the RACF **search** command. See Table 43 on page 289 to determine which SDBM searches are affected.
**RACF restriction on amount of input:** RACF limits the number of operands that are specified in RACF commands. If the number of operands surpasses this limit, RACF ignores some of the operands and processes the command. Therefore, an SDBM add or modify operation containing many attributes appears to run successfully but some of the attributes might not be set. For more information, see [z/OS Security Server RACF Command Language Reference](#).

**LDAP restriction on RACF data:** With the exception of the RACF user password or password phrase envelopes, all field values sent by RACF to LDAP must consist of printable characters. If a RACF field contains unprintable characters, the value returned in the LDAP output will probably not match the RACF value and will probably not be printable. If a RACF field contains binary zeros, the LDAP output might be truncated. In particular, make sure that the installation DATA field in RACF user and resource profiles does not contain binary zeros or other unprintable characters.

**Retrieving RACF user password and password phrase envelopes**

SDBM returns the RACF user password envelope when the `racfPasswordEnvelope` attribute is specified in the attributes to be returned from a search of a RACF user. Similarly, the RACF user password phrase envelope is returned when the `racfPassPhraseEnvelope` attribute is specified on the search. Each envelope is returned by the LDAP server as a binary data berval (binary data and length). If the `racfPasswordEnvelope` and `racfPassPhraseEnvelope` attributes are not specified on the search request, the RACF envelopes are not returned.

**Note:** When using a utility such as `Idapsearch` to retrieve the password or password phrase envelopes, the returned value is base-64 encoded.

**Using SDBM to change a user password or password phrase in RACF**

There are two ways to use SDBM to change a user password or password phrase in RACF.

1. The user password or password phrase of the bind user can be changed during an LDAP simple bind to SDBM. The simple bind occurs as part of an LDAP function such as search, add, modify, compare, or delete. The password or password phrase change is provided in the password portion of the LDAP simple bind. The change must be in the following format:

   `currentvalue/newvalue`

   The current and new value must both be passwords or password phrases. An error is returned if one of the values is a password and the other is a password phrase.

   The forward slash (/) is used as the indication of a password or password phrase change during the LDAP simple bind. Password or password phrase changes made using the LDAP simple bind to the SDBM backend of the z/OS LDAP server are subject to the system password rules. A password or password phrase change fails with LDAP return code `LDAP_INVALID_CREDENTIALS` and LDAP reason code of:

   **R000101** The new password is not valid

   if the new password or password phrase does not pass the rules established on the system.

   **Note:** A forward slash (/) is a legal character in a password phrase (but not in a password). During SDBM bind, a backward slash (\) is an escape character to indicate the next character is part of the password or password phrase and has no special meaning. The backward slash is removed during bind processing. Therefore, during bind, a forward slash in a password phrase must be preceded by a backward slash to indicate the forward slash is part of the password phrase and is not the password phrase change indicator. For example, the password phrase `this\sslash\isparto\f\value\2use` must be specified as `this\sslash\isparto\f\value\2use` during bind. A backward slash is also a legal character in a password phrase (but not in a password). Therefore, a backward slash in a password phrase must be preceded by another backward slash to indicate that it is not an escape character.

   Once the bind succeeds, the password or password phrase is changed even if the LDAP function eventually fails.
For example, the following command changes the password phrase for RACF user U1 from abc to xyz, assuming the SDBM suffix is sysplex=sysplexa:

```
ldapsearch -h ldaphost -p ldapport -D racfid=u1,profiletype=user,sysplex=sysplexa
   -w abc/xyz -s base -b "" objectclass=* 
```

2. To change any RACF user's password, create an LDIF file that modifies the racfPassword attribute for that user and then invoke ldapmodify to change the password. If the syntax of the new password is not valid, the command fails, returning "ldap_modify: Unknown error". (Note that this response can also be returned under other circumstances.)

For example, the following LDIF file, pw.mod, resets the password for RACF user U1 to xyz, assuming the SDBM suffix is sysplex=sysplexa. The racfAttributes: noexpired record is added to result in a new password that is not expired. If noexpired is not specified, then the password is reset but is expired, requiring U1 to change the password at next logon.

```
dn: racfid=u1,profiletype=USER,sysplex=sysplexa 
changetype: modify 
add: x 
racfpassword: xyz 
racfattributes: noexpired 
```

Then, assuming that the RACF user admin1 has the necessary RACF authorization to update RACF, the command:

```
ldapmodify -h ldaphost -p ldapport -D racfid=admin1,profiletype=user,sysplex=sysplexa 
   -w passwd -f pw.mod 
```

modifies the password for U1.

A RACF user's password phrase is changed the same way as described above, using the racfPassPhrase attribute.

**Using LDAP operation utilities with SDBM**

The LDAP operation utilities described in IBM Tivoli Directory Server Client Programming for z/OS can be used to update data in RACF. Following are some examples. These examples assume that the RACF user admin1 has the necessary RACF authorization to make these RACF updates and that sysplex=sysplexa is the SDBM suffix.

**Example: adding a user to RACF**

If the LDIF file user.add contains:

```
dn: racfid=newuser,profiletype=USER,sysplex=sysplexa 
objectclass: racfUser 
racfid: newuser 
```

The following command adds user ID newuser to RACF:

```
ldapadd -h ldaphost -p ldapport -D racfid=admin1,profiletype=user,sysplex=sysplexa 
   -w passwd -f user.add 
```

Note that the only required attribute to add a user is the user ID specified as racfid. This mimics the RACF adduser command.

**Example: modifying a user in RACF**

To add a TSO segment for newuser, the LDIF file user.mods could contain:

```
dn: racfid=newuser,profiletype=USER,sysplex=sysplexa 
changetype: modify 
objectclass: SAFTsosegment 
SAFAccountNumber: 123 
SAFHoldClass: H 
SAFLogonSize: 1024 
```
The command:

```
ldapmodify -h ldaphost -p ldapport -D racfid=admin1,profiletype=user,sysplex=sysplexa
-w passwd -f user.mods
```

modifies the RACF user profile for user ID newuser, adding a TSO segment with the specified values.

**Example: searching for user information in RACF**

To see the information in RACF for newuser, the following search command can be performed:

```
ldapsearch -h ldaphost -p ldapport -D racfid=admin1,profiletype=user,sysplex=sysplexa -w passwd
-b "racfid=newuser,profiletype=user,sysplex=sysplexa" "objectclass=*"
```

The results that are returned are most of the non-default data that RACF displays on a `listuser` command, but using LDAP attribute names. Following is an example for newuser:

```
racfid=NEWUSER,profiletype=USER,sysplex=SYSPLEXA
racfid=NEWUSER
racfauthorizationdate=07/18/05
racfowner=RACFID=ADMIN1,PROFILETYPE=USER,SYSPLEX=SYSPLEXA
racfpasswordinterval=186
racfdefaultgroup=RACFID=G1,PROFILETYPE=GROUP,SYSPLEX=SYSPLEXA
racflogondays=SUNDAY
racflogondays=MONDAY
racflogondays=TUESDAY
racflogondays=WEDNESDAY
racflogondays=THURSDAY
racflogondays=FRIDAY
racflogondays=SATURDAY
racflogontime=ANYTIME
racfconnectgroupname=RACFID=G1,PROFILETYPE=GROUP,SYSPLEX=SYSPLEXA
racfhavepasswordenvelope=YES
racfhavepassphraseenvelope=YES
racfpassphrasedatechangedate=06/11/07
racfattributes=PASSWORD
racfattributes=PASSPHRASE
safaccountnumber=123
safholdclass=H
saflogonsize=1024
safmaximumregionsize=0
safuserdata=0000
objectclass=RACFBASECOMMON
objectclass=RACFUSER
objectclass=SAFTSOSEGMENT
```

**Example: searching for a user's password and password phrase envelopes in RACF**

The following search returns the `racfPasswordEnvelope` and `racfPassPhraseEnvelope` attributes:

```
ldapsearch -h ldaphost -p ldapport -D racfid=admin1,profiletype=user,sysplex=sysplexa
-w passwd -L -b racfid=newuser,profiletype=user,sysplex=sysplexa
"objectclass=*" racfpasswordenvelope racfpassphraseenvelope
```

The result returned is:

```
dn: racfid=newuser,profiletype=user,sysplex=sysplexa
racfpasswordenvelope:: base-64_encoded_password_envelope
racfpassphraseenvelope:: base-64_encoded_passphrase_envelope
```

**Example: adding a group to RACF**

If the LDIF file `group.add` contains:

```
298
```

z/OS V1R11.0 IBM Tivoli Directory Server Administration and Use for z/OS
The following command adds group ID grp222 to RACF:

```
ldapadd -h ldaphost -p ldapport -D racfid=admin1,profiletype=user,sysplex=sysplexa
          -w passwd -f group.add
```

Note that the only required attribute to add a group is the group ID specified as racfid. This mimics the RACF `addgroup` command.

The LDAP commands for modifying, searching, and removing a RACF group using SDBM are very similar to the corresponding commands for a RACF user. See the examples in this section for a RACF user for more information.

**Example: connecting a user to a group in RACF**

To connect `newuser` to group `grp222`, the LDIF file `connect.add` could contain:

```
dn: racfuserid=newuser+racfgroupid=grp222,profiletype=connect,sysplex=sysplexa
objectclass: racfconnect
racfuserid: newuser
racfgroupid: grp222
```

The command:

```
ldapadd -h ldaphost -p ldapport -D racfid=admin1,profiletype=user,sysplex=sysplexa
          -w passwd -f connect.add
```

makes `newuser` a member of the `grp222` group. Note that `grp222` must be an existing RACF group ID, `newuser` must be an existing RACF user ID, and the only required attributes to add a connection are `racfuserid` (the user ID) and `racfgroupid` (the group ID).

**Example: searching for information about a user's connection to a group in RACF**

To see information about `newuser`'s connection to the `grp222` group, the following search can be performed:

```
ldapsearch -h ldaphost -p ldapport -D racfid=admin1,profiletype=user,sysplex=sysplexa
          -w passwd
          -b "racfuserid=newuser+racfgroupid=grp222,profiletype=connect,sysplex=sysplexa"
          "objectclass=*"
```

The result returned is the non-default information from the GROUP section that RACF displays on a `listuser` command, but using LDAP attribute names. Following is an example for `newuser`'s connection to `grp222`:

```
racfuserid=NEWUSER+racfgroupid=GRP222,profiletype=CONNECT,sysplex=SYSPLEXA
racfuserid=NEWUSER
racfgroupid=GRP222
racfconnectauthdate=07/18/05
racfconnectowner=RACFID=ADMIN1,PROFILETYPE=USER,SYSPLEX=SYSPLEXA
racfconnectgroupauthority=USE
racfconnectgroupauthority=NONE
racfconnectgroupmembership=NONE
objectclass=RACFBASECOMMON
objectclass=RACFCONNECT
```

To see all the groups that `newuser` is connected to, either of the following searches can be performed:

```
ldapsearch -h ldaphost -p ldapport -D racfid=admin1,profiletype=user,sysplex=sysplexa
          -w passwd
          -b "profiletype=connect,sysplex=sysplexa"
          "racfuserid=newuser"
```

or

```
ldapsearch -h ldaphost -p ldapport -D racfid=admin1,profiletype=user,sysplex=sysplexa
          -w passwd
          -b "profiletype=connect,sysplex=sysplexa"
          "racfuserid=newuser"
```
ldapsearch -h ldaphost -p ldapport -D racfid=admin1,profiletype=user,sysplex=sysplexa -w passwd 
-b "profiletype=connect,sysplex=sysplexa" "(&(racfuserid=newuser)(racfgroupid=*))"

For both commands, the results are:
racfuserid=NEWUSER+racfgroupid=G1,profiletype=CONNECT,sysplex=sysplexa
racfuserid=NEWUSER+racfgroupid=GRP222,profiletype=CONNECT,sysplex=sysplexa

Note that G1 was the default group to which newuser was connected when newuser was created.

Example: removing a user from a group in RACF
The following command removes newuser from the grp222 group (the equivalent of the RACF remove command):
ldapdelete -h ldaphost -p ldapport -D racfid=admin1,profiletype=user,sysplex=sysplexa -w passwd  
"racfuserid=newuser+racfgroupid=grp222,profiletype=connect,sysplex=sysplexa"

Example: removing a user from RACF
The following command removes the newuser user profile from RACF, also removing all of newuser's connections to groups (the equivalent of a RACF deluser command):
ldapdelete -h ldaphost -p ldapport -D racfid=admin1,profiletype=user,sysplex=sysplexa -w passwd  
"racfid=newuser,profiletype=user,sysplex=sysplexa"

Example: adding a resource profile in the facility class and giving a user and a group access to the profile
If the LDIF file resource.add contains:\ndn: profilename=NEW.RESOURCE,profiletype=facility,sysplex=sysplexa
objectclass: racfResource
objectclass: extensibleObject
profilename: NEW.RESOURCE
racfuacc: read
racfnotify: admin1
racfaccesscontrol: ID(u1) ACCESS(UPDATE)
racfaccesscontrol: ID(g1) ACCESS(CONTROL) WHEN(TERMINAL(T2))

The following command adds the NEW.RESOURCE resource profile to the facility class in RACF and gives the requested access:
ldapadd -h ldaphost -p ldapport -D racfid=admin1,profiletype=user,sysplex=sysplexa  
-w passwd -f resource.add

Note: A RACF redefine command followed by two RACF permit commands are issued.

Example: refreshing the raclist for the facility class
If the LDIF file setropts.mod contains:
dn: cn=setropts,sysplex=sysplexa
changetype: modify
racfraclist: facility
racfsetroptsattributes: refresh

The command:
ldapmodify -h ldaphost -p ldapport -D racfid=admin1,profiletype=user,sysplex=sysplexa  
-w passwd -f setropts.mod

issues a RACF setropts command to refresh the raclisted profiles in the facility class.

Notes:
1. The PROGRAM class is not refreshed using RACLIST. Instead, issue the ldapmodify command with  
setropts.mod containing:
2. Refreshing the CFIELD class using SDBM is not supported.

Deleting attributes

If a request is made to delete the racfAccessControl, racfAttributes, racfConnectAttributes, racfResourceAttributes, or racfSetroptsAttributes attribute and no values are provided, SDBM adds the following to the appropriate RACF command (even if the profile does not currently have that value):

- racfAccessControl - RESET(ALL)
- racfAttributes - NOADSP NOAUDITOR NOGRPACC NOOIDCARD NOOPERATIONS NOSPECIAL
- racfConnectAttributes - NOADSP NOAUDITOR NOGRPACC NOOPERATIONS NOSPECIAL
- racfResourceAttributes - NOSINGLEDSN NOTVTOC NOWARNING
- racfSetroptsAttributes - NOWHEN(PROGRAM)

Deleting a specific value for these attributes requires that the value itself be specified on the delete operation.

For example, to remove the OPERATIONS and AUDITOR values from the racfAttributes values of user ID user1 (leaving any other racfAttributes values the user has), issue an ldapmodify command with the following file:

dn: racfid=user1,profiletype=user,sysplex=sysplexa
changeType: modify
delete: racfAttributes
racfAttributes: OPERATIONS
racfAttributes: AUDITOR

To remove all the racfAttributes values listed above of user ID user1, issue an ldapmodify command with the following file:

dn: racfid=user1,profiletype=user,sysplex=sysplexa
changeType: modify
delete: racfAttributes

In addition, you can use the racfAttributes attribute to remove an entire segment from a user. For example, to remove the OMVS segment from user ID user1, issue an ldapmodify command with one of the following files:

dn: racfid=user1,profiletype=user,sysplex=sysplexa
changeType: modify
delete: racfAttributes
racfAttributes: OMVS

or

dn: racfid=user1,profiletype=user,sysplex=sysplexa
changeType: modify
add: racfAttributes
racfAttributes: NOOMVS

Following are some additional examples of deleting attributes:

- dn: racfid=user1,profiletype=user,sysplex=sysplexa
  changeType: modify
delete: racfProgrammerName

Returns: LDAP_UNWILLING_TO_PERFORM

The racfProgrammerName attribute is one that cannot be deleted.
• dn: racfid=user1,profiletype=user,sysplex=sysplexa
  changetype: modify
  delete: racfBuilding
  racfBuilding: 001
  Returns: **LDAP_UNWILLING_TO_PERFORM**
  You cannot specify a value to be removed for **racfBuilding**.

• dn: racfid=user1,profiletype=user,sysplex=sysplexa
  changetype: modify
  delete: racfBuilding
  Expected result: successful removal of the attribute **racfBuilding** and **LDAP_SUCCESS** returned.
Chapter 17. Kerberos authentication

The LDAP server allows clients to authenticate to the server by using IBM’s Network Authentication and Privacy Service which is better known as Kerberos Version 5. Kerberos is a trusted third party, private-key, network authentication system. In Kerberos, a ticket, a packet of information used by a client to prove its identity, is passed to a server in place of a user name and password. This ticket is encrypted and cannot be duplicated. After the server verifies the client ticket, it sends its own ticket to the client in order for the client to authenticate it. Once the mutual authentication process is complete, the client and server have authenticated each other.

The LDAP server supports Kerberos integrity and confidentiality services. Upon successful completion of a SASL bind operation using the GSS API mechanism, the negotiated quality of protection (QOP) will be used for subsequent messages sent over the connection. This QOP will continue to be used until the completion of a new SASL bind request. If the new SASL bind request fails, the connection will revert to anonymous authentication with no integrity or confidentiality services.

Setting up for Kerberos

Kerberos Version 5 binds, defined in RFC 2222: Simple Authentication and Security Layer (SASL) are performed using the Generic Security Services Application Programming Interface (GSS API) defined in RFC 2743: Generic Security Service Application Program Interface Version 2, Update 1 and RFC 2744: Generic Security Service API Version 2 : C-bindings. From this point on the phrase “GSS API bind” is used to refer to Kerberos Version 5 binds. Before attempting to perform a Kerberos GSS API bind, be sure to:

1. Have the Network Authentication and Privacy Service (Kerberos 5) installed and configured and the service started.
2. Create a Kerberos identity for the user ID that will start the LDAP server. For example:

   
   ALTUSER LDAPSrv PASSWORD(password) NOEXPIRED KERB(KERBNAME(ldap_prefix/primary-dns-hostname))

   
   where ldap_prefix is either ldap or LDAPSrv and primary-dns-hostname is the primary hostname for the system in DNS. The Kerberos principal name associated with the LDAP server can be ldap_prefix/primary-dns-hostname@krbRealmName. The krbRealmName is the Kerberos realm in which the LDAP server will operate. Use ldap to assure interoperability with all LDAP clients. LDAPSrv is accepted, but this value is not usable with many non-z/OS LDAP clients.
3. If the KDC (Key Distribution Center) is not located on the same machine as the LDAP server, you have to generate a keytab file for the server. To generate a keytab for the server, issue the following commands:

   a. First check the version of the server’s Kerberos key (this is necessary since the version is updated every time the password is changed):

      LISTUSER LDAPSrv NORACF KERB

   b. Now the keytab command can be issued from the z/OS shell with the KEY VERSION from the LISTUSER command:

      keytab add ldap_prefix/primary-dns-hostname -p password -v key-version

   The -k filename option may also be used if you want to use your own keytab file rather than the Kerberos default keytab file. It is also important to note that when issuing Kerberos commands all passwords must be in uppercase.

   If the KDC and LDAP server are on the same system, you do not need a keytab file. If the ID which starts the LDAP server has READ access to the IRR.RUSERMAP facility class in RACF, then this can be used instead of a keytab file. Following are the RACF commands to do this:

   RDEFINE FACILITY IRR.RUSERMAP UACC(NONE)
   PERMIT IRR.RUSERMAP CLASS(FACILITY) ID(LDAPSrv) ACCESS(READ)
   SETR RAclist(FACILITY) REFRESH
4. Enable your configuration file for Kerberos authentication.

```plaintext
# Global Section
supportKrb5 on
serverKrbPrinc LDAP/myhost.com@MYREALM.COM
krbLDAPAdmin ibm-kn=ldapadm@MYREALM.COM
krbKeytab none
# TDBM Section
krbIdentityMap on
# LDBM Section
krbIdentityMap on
# SDBM Section
krbIdentityMap on
# CDBM Section
krbIdentityMap on
```

**Notes:**

a. In the above example, `myhost.com` is the primary DNS host name of the LDAP server and `MYREALM.COM` is the Kerberos realm to which the LDAP server belongs.

b. The first portion of the `serverKrbPrinc` identity can either be `ldap` or `LDAP` in the server configuration file and in the Kerberos segment of the RACF ID where it is defined. Use `ldap` to assure interoperability with all LDAP clients. LDAP is accepted, but this value is not usable with many non-z/OS LDAP clients. Check your KDC for case requirements.

c. The `serverKrbPrinc` configuration option is optional if the Kerberos principal name (KERNAME field in RACF) of the LDAP server's userid is "ldap/primary-dns-hostname", where `primary-dns-hostname` is the primary DNS hostname of the LDAP server.

5. Start your server. Your LDAP server is now configured with Kerberos support.

### Schema for Kerberos

The LDAP server schema always contains the schema elements needed for Kerberos GSS API Authentication. No additional schema is needed. `Table 44` lists the Kerberos object classes and attributes.

**Table 44. Kerberos attributes and object classes**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Object Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>krbRealmName-V2</td>
<td>krbRealm-V2</td>
<td>This attribute represents the Kerberos realms of which entries in the LDAP server are members. The entry that contains this attribute also contains the <code>krbPrincSubtree</code> attribute.</td>
</tr>
<tr>
<td>krbPrincSubtree</td>
<td>krbRealm-V2</td>
<td>This attribute is in the same entry as the <code>krbRealmName-V2</code> attribute and it identifies the directory subtrees where entries may contain Kerberos information.</td>
</tr>
<tr>
<td>krbPrincipalName</td>
<td>(no object class)</td>
<td>The attribute is used to define the entry's Kerberos identity. This attribute is used for identity mapping. Currently this attribute is not associated with an object class. This means that for an entry to contain this attribute you can add the object class <code>extensibleObject</code> or define and add your own object class.</td>
</tr>
<tr>
<td>krbAliasedObjectName</td>
<td>krbAlias</td>
<td>This attribute allows an entry to be mapped to another entry's DN.</td>
</tr>
<tr>
<td>krbHintAliases</td>
<td>krbAlias</td>
<td>This attribute is used as an authorization list. If another entry's DN is in this list and that entry specified this entry as a <code>krbAliasedObjectName</code> then the mapping is allowed.</td>
</tr>
</tbody>
</table>
### Identity mapping

The following sections describe the mapping that is done depending on your configuration. After all the identity mapping takes place you are left with a list of DNs that are used for access control and group gathering.

#### Default mapping

The GSS API bind operation passes a Kerberos identity to the LDAP server which in its initial form cannot be used for access control in the server. This Kerberos identity known as `principal@REALM` is converted to a DN of the form `ibm-kn=principal@REALM`. Now this Kerberos DN is used in access control lists. This is known as the default mapping and is always performed when a SASL bind with a mechanism of GSS API is performed.

For example, if you performed a Kerberos bind as `jeff@IBM.COM` you would be mapped to `ibm-kn=jeff@IBM.COM` and this DN is added to a list of DNs that will be used for access control throughout the server.

#### TDBM, LDBM, and CDBM mapping

The name specified in the BIND request must match the DN for the source principal used to establish the GSS API context, one of the mapped Kerberos identities, or must be null.

The DN for the source principal is formed as `ibm-kn=principal@REALM` where `principal@REALM` is obtained from the GSS API client credentials. Additional distinguished names can be assigned using Kerberos identity mapping. Kerberos identity mapping for a backend is enabled by setting the `krbIdentityMap` configuration option to `on`.

When Kerberos identity mapping is enabled for a backend, the following steps are performed:

1. Search each naming context managed by the backend for an entry with `objectClass=krbRealm-V2` and `krbRealmName-V2=REALM`, where `REALM` is the realm portion of the bound Kerberos identity. If an entry is found, the `krbPrincSubtree` attribute values for the entry specify the directory subtrees to search in the next step.

2. Search each directory subtree specified by a `krbPrincSubtree` attribute value for entries containing a `krbPrincipalName` attribute with the same value as the bound Kerberos identity. The DN of each matching entry is added to the list of mapped Kerberos identities. If a matching entry also contains the `krbAliasedObjectName` attribute and the aliased entry specified by the `krbAliasedObjectName` attribute contains a `krbHintAliases` attribute value matching the name of the matching entry, then the DN of the aliased entry is also added to the list of mapped Kerberos identities.

3. Search each naming context managed by the backend for entries with `objectClass=ibm-securityIdentities` and either a `userPrincipalName` attribute with the value `principal@realm` or an

---

**Table 44. Kerberos attributes and object classes (continued)**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Object Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>altSecurityIdentities</td>
<td>ibm-securityIdentities</td>
<td>If a user is defined to a case-insensitive Kerberos server, then the Kerberos identity associated with this entry is stored as an altSecurityIdentities rather than a krbPrincipalName.</td>
</tr>
<tr>
<td>ibm-kn</td>
<td>(no object class)</td>
<td>This attribute is a pseudo-DN so that Kerberos identities can be represented as DNs for access control. Currently this attribute is not associated with an object class. This means that for an entry to contain this attribute you can add the object class extensibleObject or define and add your own object class.</td>
</tr>
</tbody>
</table>
altSecurityIdentities attribute with the value kerberos:principal@REALM. kerberos is any mixed-case spelling of KERBEROS. The DN of each matching entry is added to the list of mapped Kerberos identities. Alias and referral search processing is not performed during Kerberos identity mapping.

**SDBM mapping**

If an SDBM backend is configured in the LDAP server configuration file and the krbIdentityMap configuration option is on, then the SDBM backend tries to map the Kerberos identity to the appropriate RACF ID. SDBM checks that the realm specified in the Kerberos identity is the RACF local realm and searches for a RACF user whose KERBNAME value in the KERB segment is the same as the principal in the Kerberos identity. If a RACF ID is found, then the SDBM DN that represents the RACF ID is added to the list of DNs. In this case, the bound user can then perform SDBM operations to access RACF data, under the context of the RACF ID.

**Configuring access control**

Since we now have a list of alternate DNs, access control has been changed to operate on the list of DNs rather than just a single DN. Group gathering is also performed on all of the DNs in the list. The following examples show how access control could be configured for Kerberos binds.

1. Setting up new ACLs in your directory:
   Use ibm-kn=principal@REALM for your aclEntry values.
   Example:
   ```
   dn: cn=Scott,o=IBM,c=US
   aclEntry: access-id:ibm-kn=jeff@IBM.COM:normal:r
   ```
   If jeff@IBM.COM performed a Kerberos bind to the server, he will be mapped to ibm-kn=jeff@IBM.COM and he would get read access to normal attributes in the Scott entry.

2. Use existing ACLs (Method 1). This method is used for Kerberos identities that are defined to IBM KDCs or case-sensitive KDCs.
   a. Set up and add the realm entry in the database.
      Example:
      ```
      dn: krbRealmName-V2=IBM.COM,o=IBM,c=US
      objectclass: krbRealm-V2
      krbRealmName-V2: IBM.COM
      krbPrincSubtree: o=IBM,c=US
      ```
      This example states that if a bound Kerberos identity has a realm of IBM.COM, then identity mapping is performed in the o=IBM,c=US subtree.

   b. Add the krbPrincipalName attribute to your entries.
      Example:
      ```
      dn: cn=Jeff,o=IBM,c=US
      objectclass: extensibleObject
      krbPrincipalName: jeff@IBM.COM
      ```
      In this example, the realm entry for jeff@IBM.COM is found and the o=IBM,c=US subtree is searched for krbPrincipalName = jeff@IBM.COM. Because there is no krbAliasedObjectName attribute in the Jeff entry, only the DN cn=Jeff,o=IBM,c=US is added to the DN list along with the default mapping of ibm-kn=jeff@IBM.COM.
      Therefore, if cn=Jeff,o=IBM,c=US was already defined in another entry's aclEntry, then jeff@IBM.COM will still have that access to the entry. For example:
      ```
      dn: cn=Ken,o=IBM,c=US
      aclEntry: access-id:cn=Jeff,o=IBM,c=US:normal:w
      ```
      In this example jeff@IBM.COM will still maintain access to the cn=Ken,o=IBM,c=US entry since TDBM or LDBM mapping was performed.
c. The `krbAliasedObjectName` attribute can also be used for identity mapping.

Example:

```
dn: cn=Jeff,o=IBM,c=US
objectclass: extensibleObject
objectClass: krbAlias
krbPrincipalName: jeff@IBM.COM
krbAliasedObjectName: cn=Tim,o=IBM,c=US
```

In this example, the realm entry for `jeff@IBM.COM` is found and the `o=IBM,c=US` subtree is searched for `krbPrincipalName = jeff@IBM.COM`. The search results in `cn=Jeff,o=IBM,c=US` being added to the DN list. Because there is a `krbAliasedObjectName` attribute in the Jeff entry, we need to look at the Tim entry before we add `cn=Tim,o=IBM,c=US` to the DN list. In order to use Tim's DN for access control he must authorize Jeff to do so. Tim's entry must look like the following:

```
dn: cn=Tim,o=IBM,c=US
objectclass: krbAlias
krbHintAliases: cn=Jeff,o=IBM,c=US
```

Since Tim has Jeff listed as a `krbHintAliases` value, the value of `krbAliasedObjectName` `cn=Tim,o=IBM,c=Us` can be added to the DN list. If the Tim entry did not contain the `krbHintAliases` with Jeff as its value, then Tim's DN would not be added to the DN list.

Therefore, if `cn=Tim,o=IBM,c=US` was already defined in another entry's `aclEntry` then `jeff@IBM.COM` will still have that access to the entry. For example:

```
dn: cn=Kim,o=IBM,c=US
aclEntry: access-id:cn=Tim,o=IBM,c=US:normal:w
```

In this example, `jeff@IBM.COM` still maintains write access to the Kim entry since TDBM, LDBM, or CDBM mapping was performed and Jeff was aliased to Tim.

3. Use existing ACLs (Method 2). This method should be used for case-insensitive KDCs. Set up your TDBM, LDBM, or CDBM entries with the `altSecurityIdentities` attribute.

Example:

```
dn: cn=Jeff,o=IBM,c=US
objectclass: ibm-securityIdentities
altSecurityIdentities: KERBEROS:jeff@IBM.COM
```

Now if `jeff@IBM.COM` performs a Kerberos bind he will be mapped to `ibm-kn=jeff@IBM.COM` as well as `cn=Jeff,o=IBM,c=US`.

Therefore, if `cn=Jeff,o=IBM,c=US` was already defined in another entry's `aclEntry` then `jeff@IBM.COM` still has that access to the entry. For example:

```
dn: cn=Ken,o=IBM,c=US
aclEntry: access-id:cn=Jeff,o=IBM,c=US:normal:w
```

In this example `jeff@IBM.COM` still maintains write access to the Ken entry since TDBM, LDBM, or CDBM mapping was performed.

Example of setting up a Kerberos directory

The following diagram shows an example of how you could set up a Kerberos directory.

Note: Because of space limitations in the diagram, the entries in the example do not contain all of the necessary information to make them valid directory entries. For example, object classes and required attributes have been left out of many of the entries.
Assume that Kerberos support has been enabled for this server, all backends have set `krbIdentityMap` to `on`, and the JEFF user ID has performed a `kinit` to acquire a Kerberos ticket before issuing the GSS API Kerberos bind.

The user JEFF with a Kerberos identity of `jeff@IBM.COM` is performing a Kerberos GSS API Bind to an LDAP server which has been configured with a TDBM backend, an LDBM backend, and an SDBM backend.

During the bind process the Kerberos identity `jeff@IBM.COM` by default is mapped to `ibm-kn=jeff@IBM.COM` and this value is added to the list of DNs that is used for access control.

After default mapping is performed, each of the backends attempt to perform identity mapping:
1. The LDBM backend first looks for the Kerberos realm object with a krbRealmName-V2=IBM.COM and will not find one. Now the backend attempts to find the entry that contains altSecurityIdentities=KERBEROS:jeff@ibm.com. The entry with the DN cn=Jeff,o=IBM,c=US matches this criteria and the DN is added to the alternate DN list.

2. Now the server moves on to the TDBM backend and tries to find the Kerberos realm object with a krbRealmName-V2=IBM.COM. This time the realm object is found so all of the krbPrincSubtree values of the realm object are collected. Next, the server searches each of these subtrees (in this example, only the o=Lotus,c=US subtree) for entries that contain krbPrincipalName=jeff@IBM.COM. In this backend the entry cn=Jeff,o=Lotus,c=US is found and is added to the DN list. Next the JEFF entry is checked for the krbAliasedObjectName attribute. There is a krbAliasedObjectName specified so authorization of the alias needs to be performed. The alias is cn=Tim,o=Lotus,c=US so the Tim entry must be checked for the attribute krbHintAliases with a value of cn=Jeff,o=Lotus,c=US. This value does exist so the DN cn=Tim,o=Lotus,c=US is added to the access control DN list.

   **Note:** If the value cn=Jeff,o=Lotus,c=US did not exist in Tim's krbHintAliases, then Tim did not want you to alias him, so the DN cn=Tim,o=Lotus,c=US would not have been added to the DN list.

3. Finally, the server gets to the SDBM backend and invokes a RACF API that attempts to map the Kerberos identity jeff@IBM.COM to its associated RACF ID. In this example, the API returns the JEFF user ID and the DN racfid=JEFF,profiletype=user,sysplex=plex1 is constructed and added to the list of access control DNs.

At this point, the bind has completed and the list of DNs that is used for access control is as follows:

    ibm-kn=jeff@IBM.COM
    cn=Jeff,o=IBM,c=US
    cn=Jeff,o=Lotus,c=US
    cn=Tim,o=Lotus,c=US
    racfid=JEFF,profiletype=user,sysplex=plex1

See "Associating DNs and access groups with a bound user" on page 352 for information on group gathering after a successful bind.

Now that jeff@IBM.COM is bound to the server and his list of alternate DNs has been generated, he now has authority to perform other operations:

- Because jeff@IBM.COM was mapped to ibm-kn=jeff@IBM.COM he has read and write permission to normal attributes in the cn=Scott,o=IBM,c=US entry.
- jeff@IBM.COM also has read and write permission to the normal attributes in the cn=Ken,o=IBM,c=US entry because of his identity also being mapped to cn=Jeff,o=IBM,c=US.
- Modify operations would be permitted on the cn=Shayne,o=IBM,c=US entry since jeff@IBM.COM was also mapped to cn=Tim,o=Lotus,c=US and Tim has write access to Shayne.
- Read access is also permitted on the cn=Shayne,o=IBM,c=US entry because jeff@IBM.COM was mapped to the SDBM DN racfid=JEFF,profiletype=user,sysplex=plex1 who has read permission to the cn=Shayne,o=IBM,c=US entry.

You can see from this example that your access control is based on the combination of all the mapped DN's access control permissions.

### Kerberos operating environments

Because Kerberos Version 5 is interoperable with other Kerberos 5 implementations, there are a variety of different operating environments that can exist.

- Another KDC other than the z/OS KDC can be used to store Kerberos principals. Users get a ticket from the other KDC rather than the z/OS KDC. The LDAP server could also be registered to this other KDC. However a trusted realm between the z/OS KDC and the external KDC must be established.
- Another KDC can be used along with the z/OS KDC where both will store Kerberos identities. In this scenario a trusted realm needs to be configured between the two realms.
z/OS user IDs can be set up to contain an external KDC's Kerberos identities so when SDBM identity mapping is performed you can still be mapped to a RACF ID if you are strictly using the external or foreign KDC for Kerberos identities. This is done by setting up a \texttt{KERBLINK} and a trusted realm. The following \texttt{KERBLINK} example adds the foreign Kerberos identity jeff@KRB2000.IBM.COM to the RACF user JEFF:

\begin{verbatim}
RDEFINE KERBLINK /.../KRB2000.IBM.COM/jeff APPLDATA('JEFF')
\end{verbatim}

For information about how to set up trusted realms and \texttt{KERBLINK}, refer to \textit{z/OS Integrated Security Services Network Authentication Service Administration}.
Chapter 18. Native authentication

The z/OS LDAP server has the ability to authenticate to the Security Server through the TDBM, LDBM, or CDBM backends by specifying a Security Server password or password phrase on a simple bind to the backend. Authorization information is still gathered by the LDAP server based on the DN that performed the bind operation. The LDAP entry that contains the bind DN should contain either the **ibm-nativeld** or **uid** attribute to specify the Security Server ID that is associated with this entry. The ID and password or password phrase are passed to the Security Server and the verification of the password or password phrase is performed by the Security Server. Another feature of native authentication is the ability to change your password or password phrase on the Security Server by issuing an LDAP modify command.

**Notes:**
1. The SDBM backend does not have to be configured in order to use native authentication.
2. After a successful native authentication bind, the bound user can send LDAP requests to any of the configured backends. If SDBM is configured, SDBM operations are performed under the context of the Security Server ID that was used during the native authentication bind. For all other backends, LDAP operations are performed using the normal bind information (the bind DN and the groups to which it belongs).
3. The use of RACF passtickets is supported by the z/OS LDAP server when using native authentication. The job name associated with the LDAP server started task should be used as the application name when generating RACF passtickets. Refer to *[z/OS Security Server RACF Macros and Interfaces]* for more information about RACF passtickets.

### Initializing native authentication

To enable native authentication, perform the following steps:

1. Install and configure RACF or another Security Server.
2. Configure an LDAP server to run with an LDBM, TDBM, or CDBM backend and then start the server.
   - Specify the native authentication options in your LDAP server configuration file. For example:
     ```
     # TDBM Section
     useNativeAuth SELECTED
     nativeAuthSubtree o=IBM,c=US
     nativeAuthSubtree o=Lotus,c=US
     nativeUpdateAllowed ON
     ```
3. Be sure that the entries that are to perform native authentication contain either the **ibm-nativeld** attribute or a single-valued **uid** attribute with the appropriate Security Server ID as its value. It is important to note that a multi-valued **uid** without an **ibm-nativeld** causes the bind to fail because the LDAP server does not know which ID to use.

### Schema for native authentication

The LDAP server schema always contains the schema elements needed for native authentication. No additional schema is needed.

Following is the native authentication attribute type:

**ibm-nativeld**

Specifies the Security Server ID that is to be associated with this entry.

Following is the native authentication object class:

**ibm-nativeAuthentication**

Allows specifying the **ibm-nativeld** attribute in entries.
---

**Defining participation in native authentication**

There are many different configuration options for native authentication which are discussed in this section.

The main configuration option, `useNativeAuth`, can be set to `selected`, `all`, or `off`. If you want all entries in a certain subtree to participate in native authentication then you would choose `all` for this option. However, if you would like specific entries in the specific subtrees to be subject to native authentication, then choose `selected` for the `useNativeAuth` option. When `selected` is used, only entries with the `ibm-nativeld` attribute are subject to native authentication.

Next, consider what portions of your directory should have the ability to participate in native authentication. If the entire directory should participate, then set the `nativeAuthSubtree` configuration option to `all`. If there are different subtrees in your directory which contain entries that need to bind natively or perform native password or password phrase modifications, then you need to list all the subtrees with multiple `nativeAuthSubtree` configuration options.

**Note:** If the DN that is listed in the `nativeAuthSubtree` options contains a space character in it, then the entire DN must be enclosed in quotes in the LDAP server configuration file.

In order for an entry to bind natively or perform a native password or password phrase modify, that entry must contain a mapping to the Security Server identity that is associated with the user. This can be accomplished by using either the `ibm-nativeld` attribute or the `uid` attribute. If your directory entries already contain a single-valued `uid` attribute (which holds the Security Server user ID), then these entries are already configured for native authentication if you plan on using the `useNativeAuth all` option. If you do not plan on using `uids` for mapping, then you can specify the `ibm-nativeld` attribute for your Security Server ID associations and this attribute is used with `selected` or `all` specified for the `useNativeAuth` option. If both the `ibm-nativeld` and `uid` attributes exist in an entry, the `ibm-nativeld` value is used. The user ID specified by either the `uid` or `ibm-nativeld` attributes must contain a valid OMVS segment with an OMVS UID value in the Security Server. If a native entry has an existing `userPassword` attribute value because it was originally created under a non-native authentication subtree and the Security Server identity that is specified has not yet been defined in the Security Server, the LDAP server attempts an LDAP simple bind. Similarly, if a Security Server identity is defined but it does not contain an OMVS segment, the LDAP server attempts an LDAP simple bind.

- If you use the `useNativeAuth` option, also specify the `nativeUpdateAllowed` option to enable native password or password phrase changes in the Security Server to occur through the TDBM, LDBM, or CDBM backend.

An entry that is participating in native authentication cannot normally contain the `userPassword` attribute. An LDAP add request of an entry that contains a `userPassword` attribute value fails. An LDAP modify request that enables an entry for native authentication removes any existing `userPassword` attribute values for the entry.

**Binding with native authentication**

As mentioned above, there are two LDAP operations affected: bind and password or password phrase modify. There is a set of criteria that is used to determine if an entry actually participates in native authentication. This criteria changes depending on the configuration options that have been selected. The following table outlines all the possible operating modes for native authentication binding.

**Table 45. Operating modes for native authentication binding**

<table>
<thead>
<tr>
<th>Operation</th>
<th>useNativeAuth</th>
<th>nativeUpdate Allowed</th>
<th>ibm-nativeld</th>
<th>uid</th>
<th>Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bind</td>
<td>selected</td>
<td>any value</td>
<td>User1</td>
<td></td>
<td>Entry is configured correctly and native authentication is attempted.</td>
</tr>
</tbody>
</table>
Table 45. Operating modes for native authentication binding (continued)

<table>
<thead>
<tr>
<th>Operation</th>
<th>useNativeAuth</th>
<th>nativeUpdate Allowed</th>
<th>ibm-nativeld</th>
<th>uid</th>
<th>Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bind</td>
<td>selected</td>
<td>any value</td>
<td>User1</td>
<td></td>
<td>Entry is not correctly configured for native authentication so an LDAP simple bind is attempted. The uid attribute is not used when useNativeAuth is selected.</td>
</tr>
<tr>
<td>Bind</td>
<td>selected</td>
<td>any value</td>
<td></td>
<td></td>
<td>Entry has not been configured for native authentication so an LDAP simple bind is attempted.</td>
</tr>
<tr>
<td>Bind</td>
<td>all</td>
<td>any value</td>
<td>User1</td>
<td>User2</td>
<td>The ibm-nativeld attribute is used to attempt native authentication.</td>
</tr>
<tr>
<td>Bind</td>
<td>all</td>
<td>any value</td>
<td>User1</td>
<td></td>
<td>Entry is configured correctly and native authentication is attempted.</td>
</tr>
<tr>
<td>Bind</td>
<td>all</td>
<td>any value</td>
<td></td>
<td></td>
<td>For ease of implementation, a LDAP simple bind is attempted, even though you have specified that all entries should use native authentication. This entry should be configured correctly.</td>
</tr>
</tbody>
</table>

**Notes:** This table assumes that the entry is located within native authentication subtrees.

In native authentication binding, the LDAP server invokes `__passwd()` using the mapped user ID and the password or password phrase supplied in the bind request. The following **errno** values returned by `__passwd()` have an LDAP reason code defined for them:

Table 46. The **errno** values returned by `__passwd()` when binding

<table>
<thead>
<tr>
<th>errno value</th>
<th>Reason</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>EACCES</td>
<td>R004111</td>
<td>The password is not correct</td>
</tr>
<tr>
<td>EINVAL</td>
<td>R004112</td>
<td>A bind argument is not valid</td>
</tr>
<tr>
<td>EMVSERR</td>
<td>R004107</td>
<td>The <code>__passwd</code> function failed; not loaded from a program controlled library</td>
</tr>
<tr>
<td>EMVSEXPIRE</td>
<td>R004109</td>
<td>The password has expired</td>
</tr>
<tr>
<td>EMVSPASSWORD</td>
<td>R004128</td>
<td>Native authentication password change failed: The new password is not valid, or does not meet requirements</td>
</tr>
<tr>
<td>EMVSSAFEXTRERR</td>
<td>R004110</td>
<td>The user id has been revoked</td>
</tr>
<tr>
<td>EMVSSAF2ERR (system problem)</td>
<td>R004176</td>
<td>The <code>__passwd()</code> function failed with error <code>error_code</code></td>
</tr>
<tr>
<td>EMVSSAF2ERR (userid problem)</td>
<td>R004108</td>
<td>Native user ID <code>name</code> is either not defined or no UID is present in the OMVS segment</td>
</tr>
<tr>
<td>ESRCH</td>
<td>—</td>
<td>Attempts a simple bind</td>
</tr>
</tbody>
</table>

**Note:** The same reason codes are issued when binding with a password or a password phrase.

The return code returned by LDAP is **LDAP_OPERATIONS_ERROR** when the **errno** value is EMVSERR or EMVSSAF2ERR (system or userid problem). For the other **errno** values except ESRCH, the return code is **LDAP_INVALID_CREDENTIALS**. For ESRCH, a simple bind is attempted (which can fail and return other return codes and reason codes).
Updating native passwords and password phrases

Performing a native password or password phrase modify is as simple as issuing an `ldapmodify` command to perform a delete followed by an add of the `userPassword` attribute.

**Note:** The `userPassword` attribute is used as a mechanism to change the native password or password phrase, but an entry that is using native authentication cannot actually include the `userPassword` attribute. An add request of an entry fails if it contains the `userPassword` attribute. A modify request of an entry will remove any existing `userPassword` attribute values from the entry. You cannot issue a single delete, add, or replace of `userPassword` values; you can only specify the combination of delete followed by an add.

Specify the current password or password phrase on the delete statement followed by the new password or password phrase on the add statement. The delete must occur before the add for native password or password phrase modify. For example, if the file `pw.mod` contains:

```none
cn=You,o=IBM,c=US
-userpassword= currentpassword
+userpassword= newpassword
```

then the following command modifies the native password (assuming the bind DN has the authority to do this):

```bash
ldapmodify ... -D cn=You,o=IBM,c=US -w currentpassword -f pw.mod
```

An error is returned if the user id specified by the `ibm-nativeld` or `uid` attribute value is not defined to the Security Server. Also, the current and new `userPassword` values must both be passwords or password phrases. An error is returned if one of the values is a password and the other is a password phrase.

The following table outlines all the possible operating modes for native authentication password updates. The same operating modes and behaviors also apply to native authentication password phrase updates.

<table>
<thead>
<tr>
<th>Operation</th>
<th>useNativeAuth</th>
<th>nativeUpdate</th>
<th>ibm-nativeld</th>
<th>uid</th>
<th>Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modify-Replace</td>
<td>selected</td>
<td>Yes</td>
<td>User1</td>
<td></td>
<td>Operation is not allowed because the entry is configured for native authentication. A modify-delete followed by a modify-add must be performed.</td>
</tr>
<tr>
<td>Modify-Replace</td>
<td>selected</td>
<td>Yes</td>
<td></td>
<td></td>
<td>Entry is not configured for native authentication so a regular LDAP password replace is attempted.</td>
</tr>
<tr>
<td>Modify-Replace</td>
<td>all</td>
<td>Yes</td>
<td></td>
<td></td>
<td>Operation is not allowed. modify-delete followed by a modify-add must be performed.</td>
</tr>
<tr>
<td>Modify-Delete</td>
<td>selected</td>
<td>Yes</td>
<td>User1</td>
<td></td>
<td>Entry is configured for native authentication so the value specified is used to change User1’s Security Server password if a modify-add follows this operation. If a modify-add does not follow, then the operation fails. Also, if the Security Server ID is not defined, the operation fails.</td>
</tr>
</tbody>
</table>
### Table 47. Operating modes for updating native password or password phrases (continued)

<table>
<thead>
<tr>
<th>Operation</th>
<th>useNativeAuth</th>
<th>nativeUpdate</th>
<th>ibm-nativeld</th>
<th>uid</th>
<th>Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modify-Delete</td>
<td>selected</td>
<td>Yes</td>
<td></td>
<td></td>
<td>Entry is not configured for native authentication so a regular LDAP modify-delete is attempted.</td>
</tr>
<tr>
<td>(password)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modify-Delete</td>
<td>all</td>
<td>Yes</td>
<td>User1</td>
<td>User2</td>
<td>Entry is configured for native authentication so the value specified is used to change User1’s Security Server password if a modify-add follows this operation. If a modify-add does not follow, then the operation fails. Also, if the Security Server ID is not defined, the operation fails.</td>
</tr>
<tr>
<td>(password)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modify-Delete</td>
<td>all</td>
<td>Yes</td>
<td>User1</td>
<td></td>
<td>Entry is configured for native authentication so the value specified is used to change User1’s Security Server password if a modify-add follows this operation. If a modify-add does not follow, then the operation fails. Also, if the Security Server ID is not defined, the operation fails.</td>
</tr>
<tr>
<td>(password)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modify-Delete</td>
<td>all</td>
<td>Yes</td>
<td></td>
<td></td>
<td>A regular LDAP modify-delete is allowed in this case to allow for old LDAP passwords stored in TDBM, LDBM, or CDBM to be removed.</td>
</tr>
<tr>
<td>(password)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modify-Add</td>
<td>selected</td>
<td>Yes</td>
<td>User1</td>
<td></td>
<td>If a password modify-delete was previously performed, then a Security Server password change for User1 is attempted. If the modify-delete has not been performed, then the operation fails. Also, if the Security Server ID is not defined, the operation fails.</td>
</tr>
<tr>
<td>(password)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modify-Add</td>
<td>selected</td>
<td>Yes</td>
<td></td>
<td></td>
<td>Entry is not configured for native authentication so a regular LDAP modify-add is attempted.</td>
</tr>
<tr>
<td>(password)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modify-Add</td>
<td>all</td>
<td>Yes</td>
<td>User1</td>
<td>User2</td>
<td>If a password modify-delete was previously performed then a Security Server password change for User1 is attempted. If the modify-delete has not been performed, then the operation fails. Also, if the Security Server ID is not defined, the operation fails.</td>
</tr>
<tr>
<td>(password)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modify-Add</td>
<td>all</td>
<td>Yes</td>
<td>User1</td>
<td></td>
<td>If a password modify-delete was previously performed, then a Security Server password change for User1 is attempted. If the modify-delete has not been performed, then the operation fails. Also, if the Security Server ID is not defined, the operation fails.</td>
</tr>
<tr>
<td>(password)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 47. Operating modes for updating native password or password phrases (continued)

<table>
<thead>
<tr>
<th>Operation</th>
<th>useNativeAuth</th>
<th>nativeUpdate Allowed</th>
<th>ibm-nativeld</th>
<th>uid</th>
<th>Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modify-Add</td>
<td>all</td>
<td>Yes</td>
<td></td>
<td></td>
<td>Operation fails because the entry is not correctly configured for native authentication.</td>
</tr>
<tr>
<td>(password)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add (entry with</td>
<td>selected</td>
<td>User1</td>
<td></td>
<td></td>
<td>Entry is configured for native authentication so adding an entry with a password is not allowed.</td>
</tr>
<tr>
<td>password)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add (entry with</td>
<td>selected</td>
<td>User1</td>
<td></td>
<td></td>
<td>Entry is not configured for native authentication so the operation is attempted.</td>
</tr>
<tr>
<td>password)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add (entry with</td>
<td>selected</td>
<td>User1</td>
<td></td>
<td></td>
<td>Entry is not configured for native authentication so the add operation is attempted.</td>
</tr>
<tr>
<td>password)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add (entry with</td>
<td>all</td>
<td>User1</td>
<td>User2</td>
<td></td>
<td>Operation fails. Native entries cannot contain LDAP passwords.</td>
</tr>
<tr>
<td>password)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add (entry with</td>
<td>all</td>
<td>User1</td>
<td></td>
<td></td>
<td>Operation fails. Native entries cannot contain LDAP passwords.</td>
</tr>
<tr>
<td>password)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add (entry with</td>
<td>all</td>
<td>User1</td>
<td></td>
<td></td>
<td>Operation fails. Native entries cannot contain LDAP passwords.</td>
</tr>
<tr>
<td>password)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: This table assumes that the entry is located within native authentication subtrees.

To update a native password or password phrase, the LDAP server invokes `_passwd()` using the mapped user ID and the old and new passwords or password phrases supplied in the modify delete/add request. The following `errno` values returned by `_passwd()` have an LDAP reason code defined for them:

Table 48. The `errno` values returned by `_passwd()` when updating password or password phrase

<table>
<thead>
<tr>
<th><code>errno</code> value</th>
<th>Reason</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>EACCES</td>
<td>R004111</td>
<td>The password is not correct</td>
</tr>
<tr>
<td>EINVAL</td>
<td>R004112</td>
<td>A bind argument is not valid</td>
</tr>
<tr>
<td>EMVSERR</td>
<td>R004107</td>
<td>The _passwd function failed; not loaded from a program controlled library</td>
</tr>
<tr>
<td>EMVSEXPIRE</td>
<td>R004109</td>
<td>The password has expired</td>
</tr>
<tr>
<td>EMVSPPASSWORD</td>
<td>R004128</td>
<td>Native authentication password change failed: The new password is not valid, or does not meet requirements</td>
</tr>
<tr>
<td>EMVSSAFEXTERR</td>
<td>R004110</td>
<td>The user id has been revoked</td>
</tr>
<tr>
<td>EMVSSAF2ERR (system problem)</td>
<td>R004176</td>
<td>The _passwd() function failed with error <code>error_code</code></td>
</tr>
<tr>
<td>EMVSSAF2ERR (userid problem)</td>
<td>R004108</td>
<td>Native user ID 'name' is either not defined or no UID is present in the OMVS segment</td>
</tr>
<tr>
<td>ESRCH</td>
<td>R004118</td>
<td>Native user ID 'name' is either not defined or no UID is present in the OMVS segment</td>
</tr>
</tbody>
</table>

Note: The same reason codes are issued when updating a password or a password phrase. The return code returned by LDAP is `LDAP_OPERATIONS_ERROR` when the `errno` value is `EMVSERR`, `EMVSSAF2ERR` (system or userid problem), or `ESRCH`. For the other `errno` values, the return code is `LDAP_INVALID_CREDENTIALS`. 
Updating native passwords or password phrases during bind

**Note:** This section applies only to changing native passwords during bind. This method cannot be used to change the userPassword value during a bind to a TDBM, LDBM, or CDBM entry that does not use native authentication.

It is also possible to change the RACF password or password phrase of a TDBM, LDBM, or CDBM entry participating in native authentication during an LDAP simple bind. This may be necessary if the `ldapmodify` command above fails with LDAP return code `LDAP_INVALID_CREDENTIALS` and LDAP reason code:

R004109 The password has expired

The simple bind occurs as part of an LDAP function such as search, compare, add, or modify. The password or password phrase change is provided in the password portion of the LDAP simple bind. The change must be in the following format:

`currentvalue/newvalue`

The current value and the new value must both be passwords or both be password phrases. An error is returned if one of the values is a password and the other is a password phrase.

The forward slash (/) is used as the indication of a password or password phrase change during the LDAP simple bind. Password or password phrase changes made using the LDAP simple bind to a TDBM, LDBM, or CDBM entry participating in native authentication are subject to the system password or password phrase rules. A password or password phrase change fails with LDAP return code `LDAP_INVALID_CREDENTIALS` and LDAP reason code of:

R004128 Native authentication password change failed: The new password is not valid, or does not meet requirements

if the new password or password phrase does not pass the rules established on the system.

**Note:** A forward slash (/) is a legal character in a password phrase (but not in a password). During native authentication bind, a backward slash (\) is an escape character to indicate the next character is part of the password or password phrase and has no special meaning. The backward slash is removed during bind processing. Therefore, during bind, a forward slash in a password phrase must be preceded by a backward slash (\) to indicate that the forward slash is part of the password phrase and is not the password phrase change indicator. For example, the password phrase `thisislash/ispartofthevalue\use` must be specified as `thisislash\ispartofthevalue\use` during bind. A backward slash is a legal character in a password phrase (but not in a password). Therefore, a backward slash in a password phrase must be preceded by another backward slash to indicate that it is not an escape character.

Once the bind succeeds, the password or password phrase is changed even if the LDAP function eventually fails. The `nativeUpdateAllowed` server configuration option setting does not control whether or not password or password phrase modifications can occur on an LDAP bind operation. The setting of `nativeUpdateAllowed` only controls password or password phrase modifications on a LDAP modify operation.

Assuming an LDBM or TDBM entry `cn=USER1,ou=END,o=IBM,c=US` is participating in native authentication and is mapped to user ID `USER1`, the following command changes the RACF password for user `USER1` from `abc` to `def`:

```
ldapsearch -h ldaphost -p ldapport -D "cn=USER1,ou=END,o=IBM,c=US" -w abc/def -b "ou=END,o=IBM,c=US" "objectclass=*"
```
Example of setting up native authentication

The following diagram shows an example of how you could set up native authentication.

**Note:** Because of space limitations in the diagram, the entries in the example do not contain all of the necessary information to make them valid directory entries. For example, object classes and required attributes have been left out of many of the entries.

---

**Example 1**

- Assuming these settings:
  - `useNativeAuth` selected
  - `nativeUpdateAllowed` on
  - `nativeAuthSubtree` `ou=END,o=IBM,c=US`
  - `nativeAuthSubtree` `ou=POK,o=IBM,c=US`

the following table indicates the results of operations involving each user entry:

<table>
<thead>
<tr>
<th>LDAP entry</th>
<th>Operation</th>
<th>Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>cn=User1,ou=END,o=IBM,c=US</code></td>
<td>Bind</td>
<td>Can bind natively because the entry contains a valid <code>ibm-nativeId</code>.</td>
</tr>
<tr>
<td><code>cn=User2,ou=END,o=IBM,c=US</code></td>
<td>Bind with native password change</td>
<td>Can change this native password because the entry contains a valid <code>ibm-nativeId</code>.</td>
</tr>
</tbody>
</table>

---

**Table 49. Behavior of native authentication in example 1**

---

**Figure 26. Native authentication example**

**Note:** In the behavior table for each of the following examples, a password phrase can be used instead of a password, with the same results.
Table 49. Behavior of native authentication in example 1 (continued)

<table>
<thead>
<tr>
<th>LDAP entry</th>
<th>Operation</th>
<th>Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>cn=User2,ou=END,o=IBM,c=US</td>
<td>All</td>
<td>Entry is not configured for native authentication so all operations are</td>
</tr>
<tr>
<td></td>
<td></td>
<td>regular LDAP operations.</td>
</tr>
<tr>
<td>cn=User3,ou=END,o=IBM,c=US</td>
<td>Bind</td>
<td>Attempts native authentication but fails because the Security Server ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>USER3 is not defined, then a regular LDAP bind is performed.</td>
</tr>
<tr>
<td></td>
<td>Bind with native</td>
<td>Cannot change the password on the bind because the Security Server ID</td>
</tr>
<tr>
<td></td>
<td>password change</td>
<td>USER3 is not defined.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Native password change attempted but fails because the Security Server</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ID USER3 is not defined.</td>
</tr>
<tr>
<td></td>
<td>modify-delete and</td>
<td>Can change this native password because the entry contains a valid</td>
</tr>
<tr>
<td></td>
<td>modify-add</td>
<td>IBM-nativeld.</td>
</tr>
<tr>
<td></td>
<td>(userPassword)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>modify-replace</td>
<td>Cannot perform a modify-replace of the userPassword attribute because</td>
</tr>
<tr>
<td></td>
<td>(userPassword)</td>
<td>the entry is subject to native authentication and password replace is</td>
</tr>
<tr>
<td></td>
<td></td>
<td>not allowed.</td>
</tr>
<tr>
<td>cn=User4,ou=POK,o=IBM,c=US</td>
<td>All</td>
<td>Performs regular LDAP operations because the entry does not contain the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IBM-nativeld attribute.</td>
</tr>
<tr>
<td>cn=User5,ou=POK,o=IBM,c=US</td>
<td>All</td>
<td>Performs regular LDAP operations because the entry does not contain the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IBM-nativeld attribute.</td>
</tr>
<tr>
<td>cn=User6,ou=RAL,o=IBM,c=US</td>
<td>All</td>
<td>Performs regular LDAP operations because the entry does not exist in a</td>
</tr>
<tr>
<td></td>
<td></td>
<td>native subtree.</td>
</tr>
</tbody>
</table>

Example 2

Assume these settings:

```bash
useNativeAuth all
nativeUpdateAllowed on
nativeAuthSubtree ou=END,o=IBM,c=US
nativeAuthSubtree ou=POK,o=IBM,c=US
```

the following table indicates the results of operations involving each user entry:

Table 50. Behavior of native authentication in example 2

<table>
<thead>
<tr>
<th>LDAP Entry</th>
<th>Operation</th>
<th>Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>cn=User1,ou=END,o=IBM,c=US</td>
<td>Bind</td>
<td>Can bind natively because the entry contains a valid IBM-nativeld.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Can change this native password because the entry contains a valid IBM-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>nativeld.</td>
</tr>
<tr>
<td></td>
<td>modify-delete and</td>
<td>Can change this native password because the entry contains a valid IBM-</td>
</tr>
<tr>
<td></td>
<td>modify-add</td>
<td>nativeld.</td>
</tr>
<tr>
<td></td>
<td>(userPassword)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>modify-replace</td>
<td>Cannot perform a modify-replace of the userPassword attribute because the</td>
</tr>
<tr>
<td></td>
<td>(userPassword)</td>
<td>entry is subject to native authentication and password replace is not</td>
</tr>
<tr>
<td></td>
<td></td>
<td>allowed.</td>
</tr>
</tbody>
</table>
Table 50. Behavior of native authentication in example 2 (continued)

<table>
<thead>
<tr>
<th>LDAP Entry</th>
<th>Operation</th>
<th>Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>cn=User2,ou=END,o=IBM,c=US</td>
<td>Bind</td>
<td>Because there are no native attributes in this entry, a regular LDAP bind is attempted.</td>
</tr>
<tr>
<td>Bind with native password change</td>
<td></td>
<td>Cannot change the password on the bind because the entry is not properly set up for native authentication. A regular LDAP bind is attempted.</td>
</tr>
<tr>
<td>modify-delete and modify-add (userPassword)</td>
<td></td>
<td>Because there are no native attributes on this entry, native authentication password update is not attempted. A regular modification of the userPassword attribute value is attempted.</td>
</tr>
<tr>
<td>modify-replace (userPassword)</td>
<td></td>
<td>Because there are no native attributes on this entry, native authentication password update is not attempted. A regular modification of the userPassword attribute value is attempted.</td>
</tr>
<tr>
<td>cn=User3,ou=END,o=IBM,c=US</td>
<td>Bind</td>
<td>Attempts native authentication but fails because the Security Server ID USER3 is not defined, then a regular LDAP bind is performed.</td>
</tr>
<tr>
<td>Bind with native password change</td>
<td></td>
<td>Cannot change the native password on the bind because the Security Server ID USER3 is not defined.</td>
</tr>
<tr>
<td>modify-delete and modify-add (userPassword)</td>
<td></td>
<td>Native password change is attempted but fails because the Security Server ID USER3 is not defined.</td>
</tr>
<tr>
<td>modify-replace (userPassword)</td>
<td></td>
<td>An attempt to modify-replace the userPassword attribute fails because the entry is configured for native authentication.</td>
</tr>
<tr>
<td>cn=User4,ou=POK,o=IBM,c=US</td>
<td>Bind</td>
<td>Can bind natively because the entry contains a valid uid (with one value).</td>
</tr>
<tr>
<td>Bind with native password change</td>
<td></td>
<td>Can change this native password because the entry contains a valid uid (with one value).</td>
</tr>
<tr>
<td>modify-delete and modify-add (userPassword)</td>
<td></td>
<td>Can change this native password because the entry contains a valid uid (with one value).</td>
</tr>
<tr>
<td>modify-replace (userPassword)</td>
<td></td>
<td>An attempt to modify-replace the userPassword attribute fails because the entry is configured for native authentication.</td>
</tr>
<tr>
<td>cn=User5,ou=POK,o=IBM,c=US</td>
<td>Bind</td>
<td>Native bind fails because 2 uid values exist.</td>
</tr>
<tr>
<td>Bind with native password change</td>
<td></td>
<td>Cannot change the native password on the bind because 2 uid attribute values exist.</td>
</tr>
<tr>
<td>modify-delete and modify-add (userPassword)</td>
<td></td>
<td>Cannot change the native password on modify operations because 2 uid attribute values exist.</td>
</tr>
<tr>
<td>modify-replace (userPassword)</td>
<td></td>
<td>An attempt to modify-replace the userPassword fails because the entry is configured for native authentication.</td>
</tr>
<tr>
<td>cn=User6,ou=RAL,o=IBM,c=US</td>
<td>All</td>
<td>Performs regular LDAP operations because the entry does not exist in a native subtree.</td>
</tr>
</tbody>
</table>
Using native authentication with Web servers

Many Web servers provide a user ID and password challenge for authentication. These can take advantage of native authentication. The Web server must be configured to do LDAP authentication. When the challenge to do LDAP authentication is presented, the user can enter the Security Server user ID and password or password phrase (from the system where the LDAP server is running). The Web server will search the LDAP directory for an entry where `uid` equals the input user ID. The Web server will use the returned DN and the inputted password or password phrase to do an `ldap_simple_bind()`. When the LDAP server determines this entry is subject to native authentication, it will retrieve the `ibm-nativeId` or `uid` value and verify the password or password phrase with the Security Server. Note that if `useNativeAuth` is set to `selected`, it may be necessary to place the Security Server user ID into both the `uid` and `ibm-nativeId` attributes of this entry to allow the Web server processing to work correctly with native authentication.
Chapter 19. CRAM-MD5 and DIGEST-MD5 authentication

The z/OS LDAP server allows clients to authenticate using the CRAM-MD5 (Challenge Response Authentication Mechanism) and DIGEST-MD5 SASL bind mechanisms. CRAM-MD5 is defined in RFC 2195: IMAP/POP AUTHorize Extension for Simple Challenge/Response. DIGEST-MD5 is defined in RFC 2831: Using Digest Authentication as a SASL Mechanism. Both the CRAM-MD5 and DIGEST-MD5 mechanisms are multi-stage binds where the server sends the client a challenge and then the client sends a challenge response back to the server to complete the authentication. The client challenge response contains a hash of the password entered by the user, the username, and other pieces of data encoded to the specifications of either the CRAM-MD5 or DIGEST-MD5 RFCs.

The CRAM-MD5 and DIGEST-MD5 SASL bind mechanisms are more secure than performing simple binds since the credentials are not passed in clear text. Also, the CRAM-MD5 and DIGEST-MD5 bind mechanisms on the z/OS LDAP server do not require any additional products to be installed or configured.

The z/OS LDAP server DIGEST-MD5 bind mechanism supports the integrity and confidentiality options defined in RFC 2831: Using Digest Authentication as a SASL Mechanism. Upon the successful completion of a DIGEST-MD5 bind, the negotiated quality of protection (qop) is used for subsequent messages sent over the connection. The negotiated qop continues until the completion of a new SASL bind request. If the new SASL bind request fails, the connection reverts to anonymous authentication with no integrity or confidentiality support.

The DIGEST-MD5 authentication mechanism is more secure than the CRAM-MD5 authentication mechanism because it prevents chosen plaintext password attacks. During a DIGEST-MD5 authentication exchange between a client and the server, there is additional information passed which is used to construct a more robust hashing algorithm when compared against a CRAM-MD5 authentication making it more difficult to decipher.

DIGEST-MD5 bind mechanism restrictions in the z/OS LDAP server

DIGEST-MD5 restrictions on the LDAP server:
1. The unspecified userid form of the authorization identity is not supported; however, the DN version is supported on the z/OS LDAP client and server.
2. Subsequent authentication is not supported.

Considerations for setting up a TDBM, LDBM, or CDBM backend for CRAM-MD5 and DIGEST-MD5 authentication

The following are considerations for setting up a TDBM, LDBM, or CDBM backend for CRAM-MD5 and DIGEST-MD5 authentication:
1. In order to use the CRAM-MD5 bind mechanism on the z/OS LDAP server, the TDBM, LDBM, or CDBM entries that you bind with should contain a \texttt{uid} attribute value. The \texttt{uid} attribute is always in the LDAP server schema. There are three ways to perform a CRAM-MD5 bind to the z/OS LDAP server:
   a. Only specifying the bindDN in the bind request in your client application. When using the z/OS LDAP operation utilities, such as \texttt{ldapsearch}, this is done by only specifying the \texttt{-D} option.
   b. Only specifying the username in the CRAM-MD5 bind mechanism in your client application. The username that is specified must map to one of the \texttt{uid} attribute values in one of the TDBM, LDBM, or CDBM entries. When using the z/OS LDAP operation utilities, such as \texttt{ldapsearch}, this is done by only specifying the \texttt{-U} option.
   c. Specifying both the bindDN in the bind request and the username in the CRAM-MD5 bind mechanism in your client application. The username that is specified must map to one of the \texttt{uid} attribute values in one of the TDBM, LDBM, or CDBM entries. The bindDN specified in the bind
request must map to the same distinguished name as the username. When using the z/OS LDAP
operation utilities, such as ldapsearch, this is done by specifying both the -D and the -U options.

For more information on the z/OS LDAP operation utilities, see IBM Tivoli Directory Server Client
Programming for z/OS.

Assuming that the password entered on the client application is correct, the CRAM-MD5 bind is
successful, otherwise it returns an LDAP credentials error.

2. In order to use the DIGEST-MD5 bind mechanism on the z/OS LDAP server, the TDBM, LDBM, or
CDBM entries that you bind with must contain a uid attribute value. The uid attribute is always present
in the server schema. There are two ways to perform a DIGEST-MD5 bind to the z/OS LDAP server:

a. Only specifying the username in the DIGEST-MD5 bind mechanism in your client application. The
username that is specified must map to one of the uid attribute values in one of the TDBM, LDBM,
or CDBM entries. When using the z/OS LDAP operation utilities, such as ldapsearch, this is done
by only specifying the -U option.

b. Specifying both the username and the authorization DN in the DIGEST-MD5 bind mechanism in
your client application. The username that is specified must map to one of the uid attribute values
in one of the TDBM, LDBM, or CDBM entries. The authorization DN that is specified must map to
the same distinguished name as the username. When using the z/OS LDAP operation utilities,
such as ldapsearch, this is done by specifying both the -D and the -U options.

For more information on the z/OS LDAP operation utilities, see IBM Tivoli Directory Server Client
Programming for z/OS.

Assuming that the password entered on the client application is correct, the DIGEST-MD5 bind will be
successful, otherwise it will return an LDAP credentials error.

3. It is strongly suggested that the uid attribute values specified on the entries to be used for
CRAM-MD5 or DIGEST-MD5 authentication be unique across every TDBM, LDBM, and CDBM
backend that is configured on the LDAP server. Authentication can fail if more than one entry has the
same uid attribute value.

4. In order for the CRAM-MD5 and DIGEST-MD5 binds to work properly, the userPassword attribute
values for the entry must be in clear text (not recommended) or encrypted in either DES or AES. DES
and AES encryption are recommended since they both encrypt the userPassword and provide clear
text decryption. For additional information on AES and DES encryption, refer to [Configuring for
encryption]. The z/OS LDAP server needs access to the clear text password so that the CRAM-MD5
and DIGEST-MD5 bind mechanisms work properly against that entry.

5. CRAM-MD5 and DIGEST-MD5 binds are not supported with entries that are participating in native
authentication.

6. CRAM-MD5 and DIGEST-MD5 binds are not supported to the SDBM backend.

CRAM-MD5 and DIGEST-MD5 configuration option

The digestRealm option in the LDAP server configuration file allows for the specification of a realm name
to be used to help create the CRAM-MD5 and DIGEST-MD5 hashes. The value of this option gets passed
on the initial challenge from the server to the client once it has been decided that a CRAM-MD5 or
DIGEST-MD5 bind is desired. See the digestRealm option on page 82. If the digestRealm configuration
option is not specified, the realm name defaults to the fully qualified hostname of the system where the
LDAP server is running assuming that a DNS (Domain Name Server) is available. If the digestRealm
option is not specified and the fully qualified hostname of the LDAP server can not be determined because
of a problem with the DNS (Domain Name Server), any CRAM-MD5 or DIGEST-MD5 binds attempted will fail.

Example of setting up for CRAM-MD5 and DIGEST-MD5

The following diagram shows an example of how you could set up your entries in your TDBM or LDBM
backend.
Note: Because of space limitations in the diagram, the entries in the example do not contain all of the necessary information to make them valid directory entries. For example, object classes and required attributes have been left out of many of the entries.

The following table outlines what happens if you attempt to do a CRAM-MD5 or DIGEST-MD5 bind from a client. The username refers to the \texttt{-U} option on the z/OS LDAP operation utilities, while the bindDN (CRAM-MD5) or authorization DN (DIGEST-MD5) is the \texttt{-D} option on the z/OS LDAP operation utilities. See \textit{IBM Tivoli Directory Server Client Programming for z/OS} for more details on the LDAP operation utilities. This table assumes that native authentication is not turned on under the subtrees: \texttt{o=lotus} and \texttt{o=IBM}.

<table>
<thead>
<tr>
<th>Username</th>
<th>BindDN (CRAM-MD5) or authorization DN (DIGEST-MD5)</th>
<th>Password</th>
<th>Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>USER2</td>
<td>cn=tim,o=lotus</td>
<td>pw2</td>
<td>Bind is successful to \texttt{cn=tim,o=lotus}</td>
</tr>
<tr>
<td>USER2</td>
<td>\texttt{cn=tim,o=lotus}</td>
<td>pw2</td>
<td>Bind is successful to \texttt{cn=tim,o=lotus}</td>
</tr>
<tr>
<td>USER2</td>
<td>\texttt{cn=jon,o=lotus}</td>
<td>pw2</td>
<td>Bind is not successful because the bindDN (CRAM-MD5) or authorization DN (DIGEST-MD5) \texttt{cn=jon,o=lotus} does not equal the username DN \texttt{cn=tim,o=lotus}</td>
</tr>
<tr>
<td>USER1</td>
<td></td>
<td>pw1</td>
<td>Bind is not successful, because there are multiple entries with the same username value: \texttt{cn=jon,o=lotus} and \texttt{cn=jay,o=IBM}</td>
</tr>
<tr>
<td>USER1</td>
<td>\texttt{cn=jay,o=IBM}</td>
<td>secret</td>
<td>Bind is successful to \texttt{cn=jay,o=IBM} because the username DN \texttt{cn=jay,o=IBM} equals the bindDN (CRAM-MD5) or authorization DN (DIGEST-MD5) \texttt{cn=jay,o=IBM}</td>
</tr>
</tbody>
</table>
Table 51. Behavior of CRAM-MD5 and DIGEST-MD5 authentication in example (continued)

<table>
<thead>
<tr>
<th>Username</th>
<th>BindDN (CRAM-MD5) or authorization DN (DIGEST-MD5)</th>
<th>Password</th>
<th>Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>USER1</td>
<td>cn=jon,o=lotus</td>
<td>pw1</td>
<td>Bind is successful to cn=jon,o=lotus because the username DN cn=jon,o=lotus equals the bindDN (CRAM-MD5) or authorization DN (DIGEST-MD5) cn=jon,o=lotus.</td>
</tr>
<tr>
<td>USER3</td>
<td></td>
<td>pw3</td>
<td>Bind is successful to cn=karen,o=lotus.</td>
</tr>
<tr>
<td>USER4</td>
<td>cn=karen,o=lotus</td>
<td>pw3</td>
<td>Bind is successful to cn=karen,o=lotus.</td>
</tr>
<tr>
<td>USER4</td>
<td>cn=matt,o=IBM</td>
<td>pw4</td>
<td>Bind is successful to cn=matt,o=IBM.</td>
</tr>
<tr>
<td>USER3</td>
<td>cn=karen,o=lotus</td>
<td>bad</td>
<td>Bind is not successful to username DN cn=karen,o=lotus and bindDN (CRAM-MD5) or authorization DN (DIGEST-MD5) cn=karen,o=lotus because the password is incorrect.</td>
</tr>
<tr>
<td>USER5</td>
<td>cn=nothere,o=lotus</td>
<td>pw5</td>
<td>Bind is not successful because the username DN cn=steve,o=IBM does not equal the non-existent bindDN (CRAM-MD5) or authorization DN (DIGEST-MD5) cn=nothere,o=lotus.</td>
</tr>
<tr>
<td>BAD</td>
<td></td>
<td>pw1</td>
<td>Bind is not successful because a uid value equal to BAD was not found in any of the entries in the TDBM or LDBM backend.</td>
</tr>
</tbody>
</table>
Chapter 20. Using extended operations to access Policy Director data

The extended operations (EXOP) backend supports two extended operations that open a connection to the target LDAP server to access z/OS Policy Director data. The **IBMLdapProxyControl** determines the target LDAP server. To set the target LDAP server when using z/OS Policy Director, use the RACF PROXY segment. See [z/OS Security Server RACF Security Administrator's Guide](#) for more information.

The LDAP extended operations are **GetDnForUserid** and **GetPrivileges**. These extended operations are generated when an application on z/OS calls the AZN APIs (see [Policy Director Authorization Services for z/OS and OS/390 Customization and Use](#) for more information on using the AZN APIs). When the EXOP backend receives a request for either of these two operations, it uses the required **IBMLdapProxyControl** to open an LDAP connection to a target LDAP server that has been set up to store Policy Director data. Then, depending on the request, the EXOP backend issues LDAP requests to the target server to retrieve the appropriate data.

**GetDnForUserid extended operation**

For the **GetDnForUserid** extended operation, the EXOP backend retrieves all of a user ID's distinguished names (DNs) stored in the target LDAP server. The client can filter the DNs returned by the EXOP backend by specifying a search base and object class names. The sequence of events for this extended operation is:

- If the client does not specify a search base, the EXOP backend searches for the DN of all entries in all of the target server’s naming contexts that contain an **ibm-nativeld** attribute set to the specified user ID and whose set of object classes include all of the optional specified object classes. If there are no naming contexts, no results will be returned.
- If the EXOP backend does not receive entries from the target LDAP server for this first set of searches, it attempts a similar set of searches, maintaining the filtering based on the optional object classes. For the second set of searches, however, instead of searching for entries with an **ibm-nativeld** attribute set to the specified user ID, it searches for entries with a **uid** attribute set to the specified user ID.

If the client does specify a search base, the EXOP backend will attempt the same sequence of searches described above, but instead of searching all of the target LDAP server’s naming contexts, it only searches the naming context specified in the search base.

**GetDnForUserid** summarizes some different error scenarios for this extended operation and the EXOP backend’s response to such scenarios.

**GetPrivileges extended operation**

For the **GetPrivileges** extended operation, the EXOP backend retrieves all of a subject’s Policy Director data. This subject is specified by its DN. The client can specify an optional domain name if the subject does not exist in the domain named DEFAULT. Refer to **GetPrivileges** for an ASN.1 description of all of the data that the EXOP backend retrieves when it receives this extended operations request.

To satisfy this request, the EXOP backend performs many searches then combines all of the results prior to returning it to the client. Furthermore, some of the searches may require searches across all of the target LDAP server’s naming contexts. For example, to find the groups the subject is a member of, the EXOP backend performs searches under all of the target LDAP server’s naming contexts. If there are no naming contexts, no search results will be returned.

**GetPrivileges** summarizes some different error scenarios for this extended operation and the EXOP backend’s response to such scenarios.
These extended operations are used by z/OS Policy Director. More information about z/OS Policy Director is in *Policy Director Authorization Services for z/OS and OS/390 Customization and Use*. 
Chapter 21. Static, dynamic, and nested groups

The LDAP server supports group definitions. These group definitions allow for a collection of names to be easily associated for access control checking or in application-specific uses such as a mailing list. See Chapter 22, “Using access control” for additional information on access control checking.

The LDAP server supports static, dynamic, and nested groups. It is possible to query static, dynamic, and nested group memberships with the use of the `ibm-allMembers` and `ibm-allGroups` operational attributes.

For a given group entry, the `ibm-allMembers` attribute enumerates all of the members that belong in that group. For a given user entry, the `ibm-allGroups` attribute determines the groups that the user has membership in.

A search request specifying the `ibm-allMembers` or `ibm-allGroups` attribute returns group membership information for just the backend containing the base entry. Access checking is performed for the `member` and `uniqueMember` attributes when obtaining the group membership information. Additional access checking is performed on any of the attributes contained in a dynamic group URL search filter on the `memberURL` attribute. Access checking is not performed on the `memberURL` and `ibm-memberGroup` attributes themselves.

### Static groups

A static group is defined as a group where the members are defined individually. The `accessGroup`, `accessRole`, `groupOfNames`, and `ibm-staticGroup` object classes each use a multi-valued attribute called `member` to define a list of distinguished names (DNs) that belong to the static group. The `groupOfUniqueNames` object class uses a multi-valued attribute called `uniqueMember` to define a list of distinguished names (DNs) that belong to the static group. The `uniqueMember` attribute type is treated as a distinguished name and not as a distinguished name with an optional unique identifier.

These attributes and object classes are always in the LDAP server schema. Except for the `groupOfNames` and `groupOfUniqueNames` object classes, they cannot be modified. The `groupOfNames` and `groupOfUniqueNames` object classes can be modified in limited ways, as described in “Changing the initial schema” on page 244. One modification you may consider making in these two object classes is to move the `member` or `uniqueMember` attribute from the MUST list to the MAY list. This will allow static group entries using these object classes to be created without any members and also allow all the members to be deleted from existing entries.

A typical static group entry is as follows:

```

dn: cn=ldap_team_static,o=endicott
objectclass: groupOfNames
 cn: ldap_team_static
 member: cn=jon,o=endicott
 member: cn=ken,o=endicott
 member: cn=jay,o=endicott
```

### Dynamic groups

A dynamic group is defined as a group in which membership is determined using one or more LDAP search expressions. Each time a dynamic group is used by the LDAP server, a user’s membership in the group is decided by determining if the user entry matches any of the search expressions. The `ibm-dynamicGroup` and `groupOfURLs` object classes each use the multi-valued attribute called `memberURL` to define the LDAP search expression. These object classes and attribute are always in the LDAP server schema and cannot be modified.
Dynamic groups allow the group administrator to define membership in terms of attributes and allow the
directory itself to determine who is or is not a member of the group. For example, members do not need to
be manually added or deleted when a person moves to a different project or location.

Alias and referral entries are not processed during the group membership search.

The following simplified LDAP URL syntax must be used as the value of memberURL attribute to specify
the dynamic group search expression.

```
ldap://baseDN[??[searchScope][?searchFilter]]
```

where

- **baseDN**
  Specifies the DN of the entry from which the search begins in the directory. The dynamic URL is
  not used if the base entry is not within the same backend as the dynamic group entry. This
  parameter is required.

- **searchScope**
  Specifies the extent of the search. The default scope is **base**.
  - **base** Returns information only about the baseDN specified in the URL.
  - **one** Returns information about entries one level below the baseDN specified in the URL. It
does not include the baseDN.
  - **sub** Returns information about entries at all levels below and including the baseDN.

- **searchFilter**
  Is the filter that you want applied to the entries within the scope of the search. See **ldapsearch** in
  IBM Tivoli Directory Server Client Programming for z/OS for additional information on LDAP
  search filters. The default is "objectclass=*".

**Note:** As the format above suggests, the host name must not be present in the syntax. The remaining
parameters are just like the normal LDAP URL syntax, defined in RFC 2255: The LDAP URL
Format (except there is no support for extensions in the URL). Each parameter field must be
separated by a ?, even if no parameter is specified. Normally, a list of attributes to return would
have been included between the baseDN and searchScope. An add or modify operation of a
dynamic group entry fails if it contains a memberURL attribute that is not in the correct format. This
prevents introducing an improperly formatted memberURL attribute into the LDAP server. If
migrating from an Integrated Security Services LDAP server on earlier releases and there are
memberURL attribute values that not properly formatted in the directory, they are ignored by the
LDAP server.

An entry is considered to be a member of the dynamic group if it falls within the search scope and
matches the search filter. Alias entries and referral entries are treated as normal entries during the group
membership search; no alias dereferencing or referral processing is performed.

A typical dynamic group entry is the following:

```
dn: cn=ldap_team_dynamic,o=endicott
objectclass: groupOfURLs
cn: ldap_team_dynamic
memberURL: ldap:///o=endicott??sub?(ibm-group=ldapTeam)
```

**Dynamic group search filter examples:**

A single entry in which the scope defaults to base and the filter defaults to "objectclass=*":

```
ldap://cm=Ricardo,ou=Endicott,o=ibm,c=us
```

The "In Flight Systems" team with a scope of one-level and the filter defaults to "objectclass=*":

```
ldap:///o=endicott??sub?(ibm-group=ldapTeam)
```
A subtree search for all the support staff in Endicott:
ldap:///ou=Endicott,o=ibm,c=us??sub?title=*Support

A subtree search for all the Garcias or Nguyens whose first name begins with an "A":
ldap:///o=ibm,c=us??sub?(&(|(sn=Garcia)(sn=Nguyen))(cn=A*))

A search filter that includes escaped percent signs, question marks and spaces in the base DN
(o=deltawing infosystems) and filter {(&(percent=10%)(description=huh?))}:
ldap:///o=deltawing%20infosystems,c=au??sub?(&(percent=10%25)(description=huh%3f))

Nested groups

A nested group is defined as a group that references other group entries, which can be static, dynamic, or
nested groups. The **ibm-nestedGroup** object class uses the multi-valued attribute called
**ibm-memberGroup** to indicate the DNs of the groups that are referenced by the nested group. This object
class and attribute are always in the LDAP server schema and cannot be modified. Nested groups allow
LDAP administrators to construct and display group hierarchies that describe both direct and indirect group
memberships. A group referenced within the nested group is ignored if it is not in the same backend as the
nested group. The group hierarchy established by a nested group cannot loop back to itself. The LDBM or
CDBM backend rejects an add or modify operation of a nested group entry if it results in a loop. To be
compatible with TDBM in the Integrated Security Services LDAP server on previous releases, the TDBM
backend allows such an add or modify operation of a nested group. When the nested group is expanded,
such as in an **ibm-allMembers** search of the group, TDBM detects the loop and continues with the next
part of the expansion.

**Note:** The **ibm-nestedGroup** object class is an **AUXILARY** object class and also requires a
**STRUCTURAL** object class.

A typical nested group entry is as follows:
dn: cn=ldap_team_nested,o=endicott
objectclass: container
objectclass: ibm-nestedGroup
  cn: ldap_team_nested
  ibm-memberGroup: cn=ldap_team_static,o=endicott
  ibm-memberGroup: cn=ldap_team_dynamic,o=endicott
  ibm-memberGroup: cn=ldaptest_team_nested,o=endicott

Determining group membership

The members of a group entry are determined depending on the type of group. Note that a group can be
multiple types (for instance, both dynamic and static).

1. **static group:** the values of the **member** attribute of the group entry if the object class of the group entry
   is **accessGroup**, **accessRole**, **groupOfNames**, or **ibm-staticGroup**, or the values of the
   **uniqueMember** attribute if the object class is **groupOfUniqueNames**.

2. **dynamic group:** the DN of each entry in this TDBM, LDBM, or CDBM backend that matches the scope
   and search filter contained in one of the values of the **memberURL** attribute of the group entry.
   Dynamic group membership is the union of all search expressions that are present on each of the
   individual **memberURL** attribute values even if the search expressions are contradictory, such as
   ldap:///o=ibm??sub?cn=bob and ldap:///o=ibm??sub?(!{cn=bob}). A dynamic search filter is ignored if
   the base in the search filter is not in the same TDBM, LDBM, or CDBM backend as the dynamic
   group.

3. **nested group:** the members of each static, dynamic, or nested group for each value of the
   **ibm-memberGroup** attribute in the nested group entry.
Zero-length values are ignored for the member, uniqueMember and ibm-memberGroup attributes.

Displaying group membership
Two operational attributes can be used for querying aggregate group membership. For a given group entry, the ibm-allMembers attribute enumerates the entire 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 attribute enumerates the entire set of groups within the same backend as the user entry to which that user has membership, including ancestor groups from nested group hierarchy. As with all operational attributes, they are only returned if explicitly requested and can not be specified on a search filter.

- The ibm-allGroups and ibm-allMembers search and comparison operations are only supported on entries within the TDBM, LDBM, or CDBM backend. These operations are not supported against users or groups that are present within the SDBM backend.

ACL restrictions on displaying group membership
The following ACL restrictions only apply when attempting to query ibm-allMembers or ibm-allGroups operational attributes. These rules do not apply when groups are gathered from all the backends that are participating in group gathering at authentication time. The entries and attributes used to evaluate ibm-allMembers and ibm-allGroups have ACL restrictions, against which the bound DN has to be checked. The members of a group are determined from three sources:

1. For static groups, the bound DN must have read access on the member or uniqueMember attribute if it is performing an ibm-allMembers or ibm-allGroups search operation, or compare access if performing a comparison operation. The member and uniqueMember attributes are in the normal access class.
2. For dynamic groups, the bound DN must have search access on all of the attributes that are present in the dynamic group filter for any of the DNs that are returned. The ACL access to the memberURL attribute does not matter when resolving ibm-allMembers or ibm-allGroups attributes.
3. For nested groups, there is no restriction on using the ibm-memberGroup attribute, but the restrictions described above apply to the groups referenced in the nested group entry. A referenced group is ignored if it is not in the same TDBM, LDBM, or CDBM backend as the nested group.

Specifying ibm-allMembers or ibm-allGroups in a search or compare operation also requires that the bound DN have read or compare access to the ibm-allMembers or ibm-allGroups attribute. Note that the ibm-allMembers and ibm-allGroups attributes are in the system access class.

For more information about access control permissions, see Chapter 22, “Using access control.”

ACL restrictions on group gathering
At authentication time, a list is created containing the static, dynamic, and nested groups of which the binding user is a member. No ACL processing is done when reading group entries for group gathering because it is not possible to know what access rights the binding user will have to any of the attributes or subtrees in the directory until all the groups have been fully determined.

Group examples

Examples of adding, modifying, and deleting group entries
Adding group entries: This example creates static group entries using the accessGroup, groupOfUniqueNames, and groupOfNames object classes.

```
ldapadd -h 127.0.0.1 -D "cn=admin" -w xxxx -f staticGrps.ldif
```

Where staticGrps.ldif contains:
This example creates a dynamic group entry that has an object class of groupOfURLs:

```
ldapadd -h 127.0.0.1 -D "cn=admin" -w xxxx -f dynamicGrp.ldif
```

Where dynamicGrp.ldif contains:
```
dn: cn=dynamic_team,o=Your Company
objectclass: groupOfURLs
cn: dynamic_team
memberurl: ldap:///o=Your Company??sub?(employeeType=ldapTeam)
```

This example creates a nested group entry with an object class of ibm-nestedGroup that references cn=dynamic_team,o=Your Company and cn=group1,o=Your Company.

```
ldapadd -h 127.0.0.1 -D "cn=admin" -w xxxx -f nestedGrp.ldif
```

Where nestedGrp.ldif contains:
```
dn: cn=nested_grp,o=Your Company
objectclass: ibm-nestedGroup
objectclass: person
cn: nested_grp
sn: group
ibm-memberGroup: cn=dynamic_team,o=Your Company
ibm-memberGroup: cn=group1,o=Your Company
```

**Modifying group entries:** In order to add a member to a static group, add the user's distinguished name as an additional value for the member or uniqueMember attribute. Following is an example:

```
ldapmodify -h 127.0.0.1 -D "cn=admin" -w xxxx -f modStaticGrp.ldif
```

Where modStaticGrp.ldif contains:
```
dn: cn=group1, o=Your Company
changetype: modify
add: member
member: cn=jeff, cn=tim, o=Your Company
```

```
dn: cn=group2, o=Your Company
changetype: modify
add: uniqueMember
uniqueMember: cn=joe,o=Your Company
```
In order to remove a member from a static group, remove the user's distinguished name from the set of `member` or `uniqueMember` attribute values in the static group entry. Following is an example:

```
ldapmodify -h 127.0.0.1 -D "cn=admin" -w xxxx -f modStaticGrp.ldif
```

Where `modStaticGrp.ldif` contains:

- `dn: cn=group1, o=Your Company`
- `changetype: modify`
- `delete: member`
- `member: cn=jeff, cn=tim, o=Your Company`
- `dn: cn=group2, o=Your Company`
- `changetype: modify`
- `delete: uniqueMember`
- `uniqueMember: cn=joe, o=Your Company`

In order to add a new search expression to a dynamic group, add the LDAP URL search expression as a value of the `memberURL` attribute. Following is an example:

```
ldapmodify -h 127.0.0.1 -D "cn=admin" -w xxxx -f modDynamicGrp.ldif
```

Where `modDynamicGrp.ldif` contains:

- `dn: cn=dynamic_team, o=Your Company`
- `changetype: modify`
- `add: memberURL`
- `memberURL: ldap:///o=Your Company??sub?(employeeType=javaTeam)`

In order to remove a search expression from a dynamic group entry, the `memberURL` attribute value containing the search expression must be removed from the group entry. Following is an example:

```
ldapmodify -h 127.0.0.1 -D "cn=admin" -w xxxx -f modDynamicGrp.ldif
```

Where `modDynamicGrp.ldif` contains:

- `dn: cn=dynamic_team, o=Your Company`
- `changetype: modify`
- `delete: memberURL`
- `memberURL: ldap:///o=Your Company??sub?(employeeType=javaTeam)`

In order to add a new group reference to an existing nested group entry, add the new group's DN as a value of the `ibm-memberGroup` attribute. Following is an example:

```
ldapmodify -h 127.0.0.1 -D "cn=admin" -w xxxx -f modNestedGrp.ldif
```

Where `modNestedGrp.ldif` contains:

- `dn: cn=nested_grp, o=Your Company`
- `changetype: modify`
- `add: ibm-memberGroup`
- `ibm-memberGroup: cn=group2, o=Your Company`

In order to remove a group reference entry from an existing nested group entry, the `ibm-memberGroup` attribute value containing the group reference DN must be deleted. Following is an example:

```
ldapmodify -h 127.0.0.1 -D "cn=admin" -w xxxx -f modNestedGrp.ldif
```

Where `modNestedGrp.ldif` contains:

- `dn: cn=nested_grp, o=Your Company`
- `changetype: modify`
- `delete: ibm-memberGroup`
- `ibm-memberGroup: cn=group2, o=Your Company`

**Deleting group entries:** In order to delete a static, dynamic, or nested group entry, delete the directory entry that represents the group. The `ldapdelete` command can be used to perform this delete operation.
This example deletes the static, dynamic, and nested group entries that were created in the above examples:

```
ldapdelete -h 127.0.0.1 -D "cn=admin" -w xxx -f deleteGrp.list
```

Where deleteGrp.list contains:

```
cn=nested_grp,o=Your Company
cn=group1,o=Your Company
cn=group2,o=Your Company
cn=group3,o=Your Company
cn=dynamic_team,o=Your Company
```

**Examples of querying group membership**

![Group hierarchy and membership for the examples](image)

The entries below are used in the following examples:

```
dn: o=ibm
objectclass: organization
aclEntry: group:CN=ANYBODY:normal:rsc:system:rsc
aclPropagate: TRUE
o: ibm
dn: cn=g1,o=ibm
objectclass: container
objectclass: ibm-nestedGroup
cn: g1
ibm-memberGroup: cn=g2,o=ibm
ibm-memberGroup: cn=g3,o=ibm
aclEntry: group:CN=ANYBODY:normal:rsc:system:rsc
dn: cn=g2,o=ibm
objectclass: accessGroup
cn: g2
member: cn=u1,o=ibm
member: cn=u2,o=ibm
aclEntry: group:CN=ANYBODY:normal:rsc:system:rsc
aclEntry: access-id:cn=u1,o=ibm:normal:rsc:system:rsc
aclEntry: access-id:cn=u2,o=ibm:normal:rsc:system:rsc:at.member:deny:rsc
dn: cn=g3,o=ibm
objectclass: container
objectclass: ibm-nestedGroup
cn: g3
ibm-memberGroup: cn=g4,o=ibm
ibm-memberGroup: cn=g5,o=ibm
ibm-memberGroup: cn=g6,o=ibm
dn: cn=g4,o=ibm
objectclass: accessGroup
cn: g4
aclEntry: group:CN=ANYBODY:normal:rsc:system:rsc
aclEntry: access-id:cn=u4,o=ibm:normal:rsc:system:rsc:at.member:deny:c
member: cn=u5,o=ibm
```
Note: The ibm-allMembers and ibm-allGroups attributes are system class attributes. The member and cn attributes are normal class attributes.

ibm-allGroups and ibm-allMembers search and comparison examples:

Example 1: This example shows an ibm-allMembers attribute search on a static group entry.

```
ldapsearch -L -D "cn=u6,o=ibm" -w secret6 -b "cn=g4,o=ibm" "objectclass=*" ibm-allMembers
```

dn: cn=g4,o=ibm
ibm-allmembers: cn=u5,o=ibm

Access checking done for cn=u6,o=ibm:
1. Read access to the \texttt{ibm-allMembers} attribute in \texttt{cn=g4,o=ibm}.

2. Read access to the \texttt{member} attribute in \texttt{cn=g4,o=ibm}.

\textbf{Example 2:} This example shows an \texttt{ibm-allMembers} attribute search on a dynamic group entry.

\begin{verbatim}
ldapsearch -L -D "cn=u6,o=ibm" -w secret6 -b "cn=g5,o=ibm" "objectclass=*" ibm-allMembers
\end{verbatim}

\begin{verbatim}
dn: cn=g5,o=ibm
ibm-allmembers: cn=u3,o=ibm
ibm-allmembers: cn=u4,o=ibm
\end{verbatim}

Access checking done for \texttt{cn=u6,o=ibm}:

1. Read access to the \texttt{ibm-allMembers} attribute in \texttt{cn=g5,o=ibm}.

2. Search access to the \texttt{cn} attribute in the returned entries, \texttt{cn=u3,o=ibm} and \texttt{cn=u4,o=ibm}, from the search filter specified in the \texttt{memberURL} attribute.

\textbf{Note:} \texttt{memberURL} attribute access rights do not matter.

\textbf{Example 3:} This example shows an \texttt{ibm-allMembers} attribute search on a nested group entry.

\begin{verbatim}
ldapsearch -L -D "cn=u6,o=ibm" -w secret6 -b "cn=g3,o=ibm" "objectclass=*" ibm-allMembers
\end{verbatim}

\begin{verbatim}
dn: cn=g3,o=ibm
ibm-allmembers: cn=g3,o=ibm
ibm-allmembers: cn=u3,o=ibm
ibm-allmembers: cn=u4,o=ibm
ibm-allmembers: cn=u5,o=ibm
\end{verbatim}

Access checking done for \texttt{cn=u6,o=ibm}:

1. Read access to the \texttt{ibm-allMembers} attribute in \texttt{cn=g3,o=ibm}.

2. Read access to the \texttt{member} attribute in \texttt{cn=g4,o=ibm}.

3. Search access to the \texttt{cn} attribute in the returned entries, \texttt{cn=u3,o=ibm} and \texttt{cn=u4,o=ibm}, from the search filter specified in the \texttt{memberURL} attribute of \texttt{cn=g5,o=ibm}.

4. Search access to the \texttt{cn} attribute in the returned entries, \texttt{cn=u3,o=ibm} and \texttt{cn=g3,o=ibm}, from the search filter specified in the \texttt{memberURL} attribute of \texttt{cn=g6,o=ibm}.

\textbf{Note:} Since \texttt{cn=u3,o=ibm} has already been added as an \texttt{ibm-allMembers} attribute value, a duplicate value will not be added.

\textbf{Note:} \texttt{ibm-memberGroup} access rights do not matter.

\textbf{Example 4:} This example shows an \texttt{ibm-allMembers} attribute search on a dynamic group entry when the bound user is not granted read access to the \texttt{ibm-allMembers} attribute.

\begin{verbatim}
ldapsearch -L -D "cn=u3,o=ibm" -w secret3 -b "cn=g5,o=ibm" "objectclass=*" ibm-allMembers
\end{verbatim}

\begin{verbatim}
dn: cn=g5,o=ibm
\end{verbatim}

Access checking done for \texttt{cn=u3,o=ibm}:

1. Read access to the \texttt{ibm-allMembers} attribute in \texttt{cn=g5,o=ibm} has been denied. Therefore, no \texttt{ibm-allMembers} attribute values will be added.

\textbf{Example 5:} This example shows an \texttt{ibm-allMembers} attribute search on a static group entry when the bound user does not have read authority on the \texttt{member} attribute.

\begin{verbatim}
ldapsearch -L -D "cn=u2,o=ibm" -w secret2 -b "cn=g2,o=ibm" "objectclass=*" ibm-allMembers
\end{verbatim}

\begin{verbatim}
dn: cn=g2,o=ibm
\end{verbatim}
Access checking done for cn=u2,o=ibm:
1. Read access to the **ibm-allMembers** attribute in cn=g2,o=ibm.
2. Read access to the **member** attribute in cn=g2,o=ibm has been denied. Therefore, the **member** attribute value will not be added as an **ibm-allMembers** attribute value.

**Example 6:** This example shows an **ibm-allMembers** attribute search on a dynamic group entry when the bound user does not have search authority in the entries that are to be returned for the attributes that are specified in the dynamic group filter.

```plaintext
ldapsearch -L -D "cn=u1,o=ibm" -w secret1 -b "cn=g5,o=ibm" "objectclass=*
ibm-allmembers

dn: cn=g5,o=ibm
ibm-allmembers: cn=u4,o=ibm
```

Access checking done for cn=u1,o=ibm:
1. Read access to the **ibm-allMembers** attribute in cn=g5,o=ibm.
2. Search access to the **cn** attribute in the returned entries, cn=u3,o=ibm and cn=u4,o=ibm, from the search filter specified in the **memberURL** attribute. However, search access has been denied on the **cn** attribute of cn=u3,o=ibm therefore it is not added as an **ibm-allMembers** attribute value.

**Example 7:** This example shows an **ibm-allMembers** comparison operation on a dynamic group entry.

```plaintext
ldapcompare -D "cn=u3,o=ibm" -w secret3 "cn=g5,o=ibm" "ibm-allmembers=cn=u3,o=ibm"

ldap_compare: Compare true
```

Access checking done for cn=u3,o=ibm:
1. Compare access to the **ibm-allMembers** attribute in cn=g5,o=ibm.
2. Search access to the **cn** attribute on the returned entries, cn=u3,o=ibm and cn=u4,o=ibm, from the search filter specified in the **memberURL** attribute.

**Example 8:** This example shows an **ibm-allGroups** attribute search where the user belongs to dynamic and nested group entries.

```plaintext
ldapsearch -L -D "cn=u6,o=ibm" -w secret6 -b "cn=u4,o=ibm" "objectclass=*
ibm-allGroups

dn: cn=u4,o=ibm
ibm-allgroups: cn=g5,o=ibm
ibm-allgroups: cn=g3,o=ibm
ibm-allgroups: cn=g1,o=ibm
```

Access checking done for cn=u6,o=ibm:
1. Read access to the **ibm-allGroups** attribute in cn=u4,o=ibm.
2. Search access on the **cn** attribute in cn=u4,o=ibm from the search filter specified in the **memberURL** attribute in cn=g5,o=ibm.

Since cn=g3,o=ibm has cn=g5,o=ibm as an **ibm-memberGroup** attribute value, cn=g3,o=ibm is added as an **ibm-allGroups** attribute also. cn=g1,o=ibm has cn=g3,o=ibm as an **ibm-memberGroup** value, therefore cn=g1,o=ibm is also added as an **ibm-allGroups** attribute value.

**Example 9:** This example shows an **ibm-allGroups** attribute search where the user belongs to static and nested group entries.

```plaintext
ldapsearch -L -D "cn=u1,o=ibm" -w secret1 -b "cn=u2,o=ibm" "objectclass=*
ibm-allGroups

dn: cn=u2,o=ibm
ibm-allgroups: cn=g2,o=ibm
ibm-allgroups: cn=g1,o=ibm
```

338  z/OS V1R11.0 IBM Tivoli Directory Server Administration and Use for z/OS
Access checking done for cn=u1,o=ibm:
1. Read access to the **ibm-allGroups** attribute in cn=u2,o=ibm.
2. Read access to the **member** attribute in cn=g2,o=ibm.

Since cn=g1,o=ibm has an **ibm-memberGroup** attribute value of cn=g2,o=ibm, cn=g1,o=ibm is added as an **ibm-allGroups** attribute value.

**Example 10:** This example shows an **ibm-allGroups** attribute search where the user being searched belongs to static and nested group entries. The bound user has read authority to the **ibm-allGroups** attribute of the user being searched, but does not have read authority on the **member** attribute in the static group entry.

```
ldapsearch -L -D "cn=u2,o=ibm" -w secret2 -b "cn=u2,o=ibm" "objectclass=*" ibm-allGroups
dn: cn=u2,o=ibm
Access checking done for cn=u2,o=ibm:
1. Read access to the **ibm-allGroups** attribute in cn=u2,o=ibm.
2. Read access to the **member** attribute of cn=g2,o=ibm is denied. Therefore, cn=g2,o=ibm is not added as an **ibm-allGroups** attribute value.
```

**Example 11:** This example shows an **ibm-allGroups** search where the bound user does not have read authority on the **ibm-allGroups** attribute.

```
ldapsearch -L -D "cn=u3,o=ibm" -w secret3 -b "cn=u4,o=ibm" "objectclass=*" ibm-allGroups
dn: cn=u4,o=ibm
Access checking done for cn=u3,o=ibm:
1. Read access to the **ibm-allGroups** attribute in cn=u4,o=ibm is denied. Therefore, no **ibm-allGroups** attribute values are added.
```

**Example 12:** This example shows an **ibm-allGroups** comparison operation where the bound user is allowed to determine that a user belongs to a nested group entry.

```
ldapcompare -D "cn=u2,o=ibm" -w secret2 "cn=u3,o=ibm" "ibm-allGroups=cn=g1,o=ibm"
ldap_compare: Compare true
```

Access checking done for cn=u2,o=ibm:
1. Compare access to the **ibm-allGroups** attribute in cn=u3,o=ibm.
2. Search access to the **cn** attribute of cn=u3,o=ibm is granted from the search filter specified in the **memberURL** attribute in cn=g5,o=ibm.

Since cn=g3,o=ibm has cn=g5,o=ibm as an **ibm-memberGroup** attribute value, cn=g3,o=ibm is added as an **ibm-allGroups** attribute as well. cn=g1,o=group has cn=g3,o=ibm as an **ibm-memberGroup** value, therefore cn=g1,o=group is also added as an **ibm-allGroups** attribute value. Therefore, the compare operation will return an **LDAP_COMPARE_TRUE** to the client application.
Chapter 22. Using access control

Access control of information in the LDAP server is specified by setting up Access Control Lists (ACLs). LDBM, TDBM, CDBM, or GDBM ACLs provide a means to protect information stored in an LDAP directory. Administrators use ACLs to restrict access to different portions of the directory, or specific directory entries. When using the LDBM, TDBM, CDBM, or GDBM backend, ACLs are created and managed using the *ldap_add* and *ldap_modify* APIs. ACLs can also be entered using the *ldif2ds* utility (TDBM only).

ACLs are represented by a set of attributes which appear to be a part of the entry. The attributes associated with access control, such as *entryOwner*, *ownerPropagate*, *aclEntry*, and *aclPropagate*, are unusual in that they are logically associated with each entry, but can have values which depend upon other entries higher in the directory hierarchy. Depending upon how they are established, these attribute values can be explicit to an entry, or inherited from an ancestor entry.

Use of LDAP’s SDBM backend allows a user to be authenticated to the directory namespace using the RACF ID and password. The RACF identity becomes associated with the user’s RACF-style distinguished name that was used on the LDAP bind operation. It is then possible to set up ACLs for entries managed by the LDBM, TDBM, CDBM, or GDBM backend using RACF-style user and group DNs. This controls access to LDBM, TDBM, CDBM, or GDBM database directory entries using the RACF user or group identities.

The LDAP server schema entry also has an ACL that can be set to control access to the schema entry.

Access control attributes

Access to LDAP directory entries and attributes is defined by Access Control Lists (ACLs). Each entry in the directory contains a special set of attributes which describe who is allowed to access information within that entry. Table 52 shows the set of attributes which are related to access control. More in-depth information about each attribute is given following the table.

It is possible to specify access control settings for individual attribute types. This is called attribute-level access control. Also, it is possible to explicitly deny access to information.

Table 52. ACL and entry owner attributes

<table>
<thead>
<tr>
<th>ACL attributes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>aclEntry</td>
<td>This is a multi-valued attribute that contains the names and permissions associated with those names that have access to information in the directory entry (or the entry along with the subtree of information below the entry, depending on the setting of the <em>aclPropagate</em> attribute).</td>
</tr>
<tr>
<td>aclPropagate</td>
<td>This is a single-valued boolean attribute which indicates whether the <em>aclEntry</em> information applies only to the directory entry it is associated with or to the entire subtree of information including and below the directory entry it is associated with. Note that propagation does not apply to entries that have an explicit <em>aclEntry</em> defined for the entry and that propagation stops at the next propagating ACL (*aclPropagate=<em>TRUE</em>) that is encountered in the directory subtree.</td>
</tr>
<tr>
<td>aclSource</td>
<td>This is a single-valued attribute that is managed by the LDAP server and cannot be changed by the <em>ldapmodify</em> command. This attribute, accessible for any directory entry, indicates the distinguished name of the entry that holds the ACL that applies to the entry. This attribute is useful in determining which propagating ACL is used to control access to information in the directory entry.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Entry owner attributes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>entryOwner</td>
<td>This is a multi-valued attribute that contains the distinguished names of users or groups that are considered owners of the directory entry (or the entry along with the subtree of information below the entry, depending on the setting of the <em>ownerPropagate</em> attribute).</td>
</tr>
</tbody>
</table>
Table 52. ACL and entry owner attributes  (continued)

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ownerPropagate</strong></td>
<td>This is a single-valued boolean attribute which indicates whether the entryOwner information applies only to the directory entry it is associated with or to the entire subtree of information including and below the directory entry it is associated with. Note that propagation does not apply to entries that have an explicit entryOwner defined for the entry and that propagation stops at the next propagating entryOwner (ownerPropagate=TRUE) that is encountered in the directory subtree.</td>
</tr>
<tr>
<td><strong>ownerSource</strong></td>
<td>This is a single-valued attribute that is managed by the LDAP server and cannot be changed by the ldapmodify command. This attribute indicates the distinguished name of the entry that holds the entryOwner that applies to the entry. This attribute is useful in determining which propagating entryOwner is used to control access to information in the directory entry.</td>
</tr>
</tbody>
</table>

**aclEntry attribute**

**aclEntry** is a multi-valued attribute which contains information pertaining to the access allowed to the entry and each of its attributes. aclEntry lists the following types of information:
- Who has rights to the entry (scope of the protection). Also called the subject.
- What specific attributes and classes of attributes (attribute access classes) that the subject has access to.
- What rights the subject has (permissions to specific attributes and classes of attributes).

**Syntax**

Following is the aclEntry attribute value syntax:

```
[access-id:|group:|role:]subject_DN[granted_rights]
```

The subject_DN is any valid DN which represents the object (entry) to which privileges are being granted. The DN ends when the first granted rights keyword is detected.

The granted_rights is specified as follows where object_rights_list is one or more elements of the set [aid], and attr_rights_list is one or more elements of the set [riwlsic].

```
[|object::[grant:|deny:]object_rights_list] [::normal::[grant:|deny:]attr_rights_list]
[|sensitive::[grant:|deny:]attr_rights_list] [::critical::[grant:|deny:]attr_rights_list]
[|restricted::[grant:|deny:]attr_rights_list] [::system::[grant:|deny:]attr_rights_list]
[::at::attr_name::[grant:|deny:]attr_rights_list]
```

Multiple specifications for the same access class or attribute type within the same aclEntry attribute value will be merged into a single specification. For example:

```plaintext
```

will result this merged access list

```plaintext
```

**Scope of protection**

The scope of the protection is based on the following three types of privilege attributes:

- **access-id**
  - The distinguished name of an entry being granted access.

- **group**
  - The distinguished name of the group entry being granted access.

- **role**
  - The distinguished name of the group entry being granted access.

The access-id, group, or role prefixes are accepted for compatibility with older levels of the LDAP server but are ignored.
Access control groups can be either static, dynamic, or nested groups. The following object classes are evaluated as group entries for the LDBM, TDBM, and CDBM backends: `ibm-staticGroup`, `groupOfNames`, `groupOfUniqueNames`, `accessRole`, `accessGroup`, `ibm-dynamicGroup`, `groupOfUrls`, and `ibm-nestedGroup`. See Chapter 21, “Static, dynamic, and nested groups” for additional information on static, dynamic, and nested groups.

Privilege attributes take the form of `type:name` where `type` refers to either `access-id`, `group`, or `role` and `name` is the distinguished name.

**Note:** The distinguished name that is used need not be the name of an entry in the directory. The distinguished name is the name that represents the user that has authenticated to the directory server.

The `type` portion of this clause is optional.

The access control implementation supports several “pseudo-DNs”. These are used to refer to large numbers of subject DN, which, at bind time, share a common characteristic in relation to either the operation being performed or the target object on which the operation is being performed. Currently, three pseudo DNs are defined:

- `group:cn=anybody`
- `group:cn=authenticated`
- `access-id:cn=this`

The `group:cn=anybody` refers to all subjects, including those that are unauthenticated (considered anonymous users). All users belong to this group automatically. The `group:cn=authenticated` refers to any DN which has been authenticated to the directory. The method of authentication is not considered. The `access-id:cn=this` refers to the bind DN which matches the target object's DN on which the operation is performed.

**Examples**

In this example, the DN type is `access-id` and the DN itself is `cn=personA, ou=deptXYZ, o=IBM, c=US`.

```
access-id:cn=personA, ou=deptXYZ, o=IBM, c=US
```

In this example, the DN type is `group` and the DN itself is `cn=deptXYZRegs, o=IBM, c=US`.

```
group:cn=deptXYZRegs, o=IBM, c=US
```

This is an example of how to use a RACF identity established with SDBM in an ACL.

```
access-id:racfid=YourID,profileType=user,sysplex=YourSysplex
access-id:racfid=YourGroup,profileType=group,sysplex=YourSysplex
```

**Attribute access classes**

Attributes requiring similar permission for access are grouped together in classes. Attributes are assigned to an attribute access class within the schema definitions. The `IBMAttributeTypes` attribute in the LDAP server schema entry holds the attribute type’s access class. The three attribute access classes are:

- `normal`
- `sensitive`
- `critical`

Each of these attribute access classes is discrete. If a user has write permission to `sensitive` attributes, then the user does not automatically have write permission to `normal` attributes. This permission must be explicitly defined.

The default attribute access class for an attribute is `normal`. By default, all users have read access to `normal` attributes. There are two additional attribute access classes used internally by LDAP for system attributes. These attribute access classes are `restricted` and `system`. You can specify these access classes when granting permissions in ACLs.
For example, a person's name would typically be defined in the **normal** class. Perhaps a social security number would be considered **sensitive**, and any password information for the user would be considered **critical**. Following are some example definitions excerpted from the LDAP server schema. Note that the attribute **userPassword** is defined with access class **critical**.

```plaintext
attributetypes: (
  2.5.4.49
  NAME ( 'dn' 'distinguishedName' )
  EQUALITY distinguishedNameMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
  USAGE userApplications
)
ibmattributetypes: (  
  ACCESS-CLASS normal
)

attributetypes: (  
  2.5.4.35
  NAME 'userPassword'
  DESC 'Defines the user password'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.40
  USAGE userApplications
)
ibmattributetypes: (  
  2.5.4.35
  ACCESS-CLASS critical
)
```

It is possible to specify access controls on individual attributes. However, when defining schema an access class is always defined for the attribute type. If not specified, that attribute type is defined to belong to the **normal** class.

**Note:** The **restricted** attributes are: **aclEntry**, **aclPropagate**, **entryOwner**, and **ownerPropagate**. In order to update access control information, you must have permissions to read and write these attributes. The **system** attributes include **aclSource** and **ownerSource** and other attributes for which the server controls the values. In order to update access control information, you must have permission to read and write these attributes. If the **system** keyword is not specified in an **aclEntry** attribute value, the system access will be set to 'system:rsc'.

**Access permissions**

Following is the set of access permissions.

**Table 53. Permissions which apply to an entire entry**

<table>
<thead>
<tr>
<th>Add</th>
<th>Add an entry below this entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete</td>
<td>Delete this entry</td>
</tr>
</tbody>
</table>

**Table 54. Permissions which apply to attribute access classes**

<table>
<thead>
<tr>
<th>Read</th>
<th>Read attribute values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Write</td>
<td>Write attribute values</td>
</tr>
<tr>
<td>Search</td>
<td>Search filter can contain attribute type</td>
</tr>
<tr>
<td>Compare</td>
<td>Compare attribute values</td>
</tr>
</tbody>
</table>

Following are some examples of valid **aclEntry** values:

access-id:cn=Tim, o=Your Company: normal:rwsc:sensitive:rsc:object:ad

role:cn=roleGroup, o=Your Company: object:ad: normal: rsc: sensitive: rsc
group:cn=group1, o=Your Company:system:csr:normal:sw
cn=Ken, o=Your Company:normal:rsc
group:cn=group2,dc=yourcompany,dc=com:normal:rwsc:at.cn:deny:w:sensitive:grant:rsc
cn=Karen,dc=yourcompany,dc=com:at.cn:grant:rwsc:normal:deny:rwsc
group:cn=anybody:normal:rsc
group:cn=authenticated:normal:rwsc:sensitive:rsc
access-id:cn=this:normal:rwsc:sensitive:rwsc:restricted:rwsc

See [Access determination](#) for information on how the aclEntry values are used to determine access.

The aclEntry attribute values are defined as directory strings but contain a distinguished name as one of the components of the attribute value. When performing a search using one of these attributes, only the distinguished name is used in the search filter and the value is normalized following the matching rules for a distinguished name. Two aclEntry attributes are considered to be the same if they have the same distinguished name.

**aclPropagate attribute**

Each entry with an explicit ACL has associated with it an aclPropagate attribute. By default, the entry’s explicit ACL is inherited down the hierarchy tree, and its aclPropagate attribute is set to TRUE. If set to FALSE, the explicit ACL for the entry becomes an override, pertaining only to the particular entry. The aclPropagate syntax is Boolean. See [Propagating ACLs](#) for more information.

**aclSource attribute**

Each entry has an associated aclSource. This reflects the DN with which the ACL is associated. This attribute is kept and managed by the server, but may be retrieved for administrative purposes. This attribute cannot be set, only retrieved.

The derivation of aclSource is further explained in [Propagating ACLs](#).

**entryOwner attribute**

Each entry has an associated entryOwner. The entryOwner might be a user or a group, similar to what is allowed within the aclEntry. However, the entryOwner subject has certain privileges over the entry.

Entry owners are, in essence, the administrators for a particular entry. They have full access on that particular entry, similar to the administrator DN. Note that the administrator DN has full permission on every entry in the database. Each entryOwner attribute value is a distinguished name. However, for compatibility with previous releases, the distinguished name can be preceded with access-id: group:, or role:.

**Note:** The distinguished name that is used need not be the name of an entry in the directory. The distinguished name is the name that represents the user that has authenticated to the directory server.

Entry owners are not constrained by permissions given in the aclEntry. They have complete access to any entry attribute, and can add and delete entries as desired.
Entry owners, the administrator DN, and users who have write permission for restricted attributes are the only people who are allowed to change the attributes related to access control. If a backend is configured for basic replication as a peer or read-only replica, only the administrator DN and the peerserver DN or masterserver DN can set the access control attributes within the backend directory. If a subtree is configured for advanced replication, only the administrator DN and the ibm-replicaCredentialsDN specified on the replication agreement can set the access control attributes within the configured replication context.

The entryOwner attribute values are defined as directory strings but contain a distinguished name as one of the components of the attribute value. When performing a search using one of these attributes, only the distinguished name is used in the search filter and the value is normalized following the matching rules for a distinguished name. Two entryOwner attributes are considered to be the same if they have the same distinguished name.

ownerPropagate attribute
Owner propagation works exactly the same as ACL propagation. By default, owners are inherited down the hierarchy tree, and their owner propagate attribute is set to TRUE. If set to FALSE, the owner becomes an override, pertaining only to the particular entry. The ownerPropagate syntax is boolean.

ownerSource attribute
Each entry also has an associated ownerSource. This reflects the DN with which the owner values are associated. This attribute is kept and managed by the server, but can be retrieved for administrative purposes. This attribute cannot be set, only retrieved.

Initializing ACLs with TDBM or LDBM
The TDBM or LDBM backend adds an ACL to each suffix entry if no aclEntry value is specified during the add of this entry (whether the add was done using ldapadd or ldif2ds). This improves performance of future ACL modifications made to an ACL placed on the suffix entry. The ACL that is used is:

```
aclEntry: cn=anybody:normal:rsc:system:rsc
aclPropagate: TRUE
```

Similarly, if no entry owner is specified when the suffix entry is created, entryOwner is added to the entry with a value set to the administrator DN, along with ownerPropagate TRUE.

Default ACLs with LDBM or TDBM
Every entry must have an ACL. If there is no ACL explicitly specified in the entry and no parent entry is propagating its ACL, then a default ACL is assigned to the entry. The default ACL is treated differently than a normal aclEntry value. The default value cannot be deleted. If an aclEntry value is later added to the entry, explicitly or by inheritance, the entire default aclEntry value is replaced. The LDAP server sets the value of the aclSource attribute to 'default' when the entry is using the default ACL. The default ACL is:

```
aclEntry: group:CN=ANYBODY:normal:rsc:system:rsc
```

Similarly, every entry must have an entry owner. If none is specified or inherited, a default entryOwner value set to the administrator DN is assigned to the entry. The default value cannot be deleted. If an entryOwner value is later added to the entry, explicitly or by inheritance, the entire default entryOwner value is replaced. The LDAP server sets the value of the ownerSource attribute to 'default' when the entry is using the default owner.
Initializing ACLs with GDBM

When the LDAP server is started with GDBM configured for the first time, the LDAP server creates the change log suffix entry, cn=changelog. The suffix entry is created with an aclEntry and entryOwner value that allows access only to the LDAP administrator and propagates the aclEntry and entryOwner values. Only the aclEntry and entryOwner attributes can be modified. When GDBM is configured to be file-based, the aclEntry and entryOwner attributes can be entirely deleted, in which case the default ACL is used. See Default ACLs with LDBM or TDBM for more information. When GDBM is configured to be DB2-based, these attributes cannot be entirely deleted. The root entry ACL is always propagated to provide access control to the change log entries because change log entries are not created with their own ACL. The change log root entry can be modified as long as change logging is enabled (the GDBM backend is configured), even if change logging is not on. Change log entries cannot be modified to override the inherited ACL values from the change log suffix entry.

Initializing ACLs with CDBM

When the LDAP server is started with CDBM configured for the first time, the LDAP server creates the following entries:

- cn=ibmpolicies
- cn=configuration
- cn=Replication,cn=configuration
- cn=Log Management,cn=configuration
- cn=Replication,cn=Log Management,cn=configuration

The cn=ibmpolicies suffix entry is created with the same initial ACL as a TDBM or LDBM suffix, which allows read access to anybody and propagates the aclEntry and entryOwner values. Therefore, only the LDAP administrator can update the cn=ibmpolicies suffix. The aclEntry and entryOwner values can be modified. If the aclEntry and entryOwner values are deleted, the default ACL is used.

The cn=configuration suffix entry is created with an aclEntry and entryOwner value that allows access only to the LDAP administrator and propagates the aclEntry and entryOwner values.

Note: It is strongly recommend that you do not entirely delete the aclEntry and entryOwner values. The default ACL is used if they are deleted, which would allow users other than the administrator access to sensitive configuration related data.

Initializing ACLs with schema entry

When the LDAP server is started for the first time, the LDAP server creates the LDAP server schema entry, cn=schema. The entry is created with the same initial ACL as a TDBM or LDBM suffix, which allows read access to anybody. Therefore, only the LDAP administrator can update the schema. The aclEntry and entryOwner values can be modified.

Access determination

The same distinguished name (DN) may be granted different access permissions to an entry, from specific access permissions to the DN and from group memberships (including the authenticated and anybody groups). The LDAP server uses the following algorithm to determine which permissions to grant a DN based on the values in the aclEntry attribute:

- if there is a specific value for the DN, the DN gets those permissions only
- else if there is a cn=this value and the DN is the distinguished name of the entry, the DN gets those permissions only
- else if there are one or more group values that the DN is a member of, the DN gets the union of the permissions for those groups
• else if there is a cn=authenticated value and the DN is authenticated to the directory with an LDAP bind operation, the DN gets those permissions only
• else if there is a cn=anybody value, the DN gets those permissions only
• otherwise the DN gets no permissions

Each of the access permissions is discrete. One permission does not imply another.

When using attribute-level permissions or grant/deny support, the order of evaluation of the separate permissions clauses is important. The access control permissions clauses are evaluated in a precedence order, not in the order in which they are found in the ACL entry value. There are four types of permissions settings: access-class grant permissions, access-class deny permissions, attribute-level grant permissions, and attribute-level deny permissions. The precedence for these types of permissions is as follows (from highest precedence to lowest):
• attribute-level deny permissions
• attribute-level grant permissions
• access-class deny permissions
• access-class grant permissions

Using this precedence, a deny permission takes precedence over a grant permission (for the same item specified) while attribute-level permissions take precedence over access-class permissions.

Following are examples for permissions:

Example 1:
```plaintext
aclEntry: group:cn=Anybody:normal:rsc
```

In this example, unauthenticated (anonymous) users have permission to read, search and compare all attributes within the normal attribute access class. ACL entry values for unauthenticated users use pseudoDN cn=Anybody.

Example 2:
```plaintext
```

In this example, the user corresponding to cn=personA, ou=deptXYZ, o=IBM, c=US has permission to add entries below the entry, to delete the entry, to read, write, search and compare both normal and sensitive attributes, and to read, search and compare critical attributes.

Example 3:
```plaintext
aclEntry: group:cn=Authenticated:normal:rwsc:sensitive:rwsc
```

In this example, users who have authenticated to the directory where a specific aclEntry value does not apply, will be allowed to read, write, search, and compare, normal and sensitive attributes in the directory entry.

Example 4:
```plaintext
aclEntry: cn=Tim,dc=yourcompany,dc=com:at.cn:deny:w:normal:rwsc
```

In this example, cn=Tim,dc=yourcompany,dc=com is granted read, write, search, and compare to normal attributes except for the cn attribute in which write access is denied. Note that the following ACL entry results in the same access:
```plaintext
aclEntry: cn=Tim,dc=yourcompany,dc=com:normal:rwsc:at.cn:deny:w
```

The evaluation of the permissions clauses is based on precedence, not order in the ACL entry value(s).
Example 5:

aclEntry: cn=Karen,dc=yourcompany,dc=com: normal: rwsc: sensitive: rsc: at. userpassword: w:
critical: deny: rwsc

In this example, cn=Karen,dc=yourcompany,dc=com is granted read, search, and compare to normal and sensitive attributes, and write to normal attributes and the userpassword attribute. All access to critical attributes (except for write in userpassword) is turned off.

Example 6:

aclEntry: group: cn=group1,dc=yourcompany,dc=com: normal: rwsc
aclEntry: group: cn=group2,dc=yourcompany,dc=com: sensitive: rwsc: at. cn: deny: w

In this example, a member of group1 only would be granted read, write, search, and compare to normal attributes. A member of both group1 and group2 would be granted read, write, search, and compare to normal and sensitive attributes, excluding write access to the cn attribute. This is an example where a member of both groups is granted access to less information than what is granted to each of the two groups individually.

Example 7:

aclEntry: access-id: cn=Tim,dc=yourcompany,dc=com: normal: rwsc: at. cn: rsc

In this example, cn=Tim,dc=yourcompany,dc=com is granted read, write, search, and compare on normal attributes and read, search, and compare on the cn attribute. Note that cn=Tim,dc=yourcompany,dc=com will also have write access to the cn attribute by virtue of cn having an access class of normal.

Example 8:

aclEntry: access-id: cn=Tim,dc=yourcompany,dc=com: normal: rwsc: at. cn: deny: rsc

In this example, cn=Tim,dc=yourcompany,dc=com is granted read, write, search, and compare on normal attributes and denied read, search, and compare on the cn attribute. Note that cn=Tim,dc=yourcompany,dc=com will still have write access to the cn attribute by virtue of cn having an access class of normal.

Search

In order to read an attribute from the directory, the user must have read permission for the specific attribute or for the attribute access class to which the attribute belongs.

Filter

In order to use an attribute in a search filter supplied on a search operation, the user must have search permission for the specific attribute or for the attribute access class to which the attribute belongs.

Compare

In order to perform a compare operation on an attribute/value combination, the user must have compare permission for the specific attribute or for the attribute access class to which the attribute belongs.

Requested attributes

If the user has the search permission on all attributes contained in the filter, the server returns as much information as possible. All requested attributes that the user has read permission for are returned.

For example, let the aclEntry be

  group: cn=Anybody: normal: rsc: sensitive: c: critical: c

and let a client perform an anonymous search.
ldapsearch -b "c=US" "cn=LastName" title userpassword telephoneNumber

where title is a normal attribute, telephoneNumber is a sensitive attribute, and userpassword is a critical attribute. Users performing anonymous searches are given the permission granted to the cn=Anybody group. In this example, permission exists to the filter since cn is in the normal attribute access class, and cn=Anybody has s (search) permission to the normal attribute access class. What is returned however, is only the title attribute for any matching entry. The telephoneNumber and userPassword attributes are not returned since cn=Anybody does not have read permissions on the sensitive and critical attribute access classes.

### Propagating ACLs

ACLs can be set on any entry in the hierarchy. ACLs can propagate down through the directory hierarchy. These ACLs, called propagating ACLs, have the aclPropagate attribute set to TRUE. All descendents of this entry will inherit the ACL set at that point, unless overridden. In order to specify an ACL different from that of its parent, a new ACL must be explicitly set.

When setting the new ACL, there is again a choice of whether to propagate the ACL. If set to TRUE, the ACL will propagate down to all descendents. If set to FALSE, the ACL is not propagated; it instead becomes an override ACL. The ACL is not propagated down through the hierarchy, but instead applies only to the one particular entry that it is associated with within the hierarchy. If unspecified, aclPropagate is set to TRUE.

An entry without an explicit ACL receives its ACL from the nearest propagating ancestor ACL. If there is no propagating ACL, the entry receives the default ACL. Propagated ACLs do not accumulate as the depth in the tree increases. The scope of a propagated ACL is from the explicitly-set propagating ACL down through the tree until another explicitly-set propagating ACL is found.

The same rules apply to propagating the entry owner based on the ownerPropagate attribute.

### Example of propagation

Following is the explicit ACL for entry ou=deptXYZ, o=IBM, c=US:

```
acIPropagate: TRUE
aclEntry: group:cn=deptXYZRegs, o=IBM, c=US:normal:rcs:sensitive:rsc
aclEntry: group:cn=Anybody:normal:rsc
aclSource: ou=deptXYZ, o=IBM, c=US
```

In the absence of an explicit ACL for entry cn=personA, ou=deptXYZ, o=IBM, c=US, the following is the implicit, propagated ACL for the entry:

```
acIPropagate: TRUE
aclEntry: group:cn=deptXYZRegs, o=IBM, c=US:normal:rcs:sensitive:rsc
aclEntry: group:cn=Anybody:normal:rsc
aclSource: ou=deptXYZ, o=IBM, c=US
```

In this example, a propagating ACL has been set on ou=deptXYZ, o=IBM, c=US. No ACL has been set on the descendant cn=personA, ou=deptXYZ, o=IBM, c=US. Therefore, the descendant inherits its ACL value from the nearest ancestor with a propagating ACL. This happens to be ou=deptXYZ, o=IBM, c=US, which is reflected in the aclSource attribute value. The aclEntry and aclPropagate values are identical to those values in the explicit propagating ACL set at ou=deptXYZ, o=IBM, c=US.

### Examples of overrides

Following is an explicit ACL for entry o=IBM, c=US:

```
acIPropagate: TRUE
```
aclPropagate: TRUE
aclEntry: group:cn=IBMRegs, o=IBM, c=US:normal:rcs:sensitive:rsc
aclEntry: group:cn=Anybody:normal:rsc
aclSource: o=IBM, c=US

Following is an explicit ACL for entry ou=deptXYZ, o=IBM, c=US:

aclPropagate: FALSE
aclEntry: group:cn=deptXYZRegs, o=IBM, c=US:normal:rcs:sensitive:rsc
aclEntry: group:cn=Anybody:normal:rsc
aclSource: ou=deptXYZ, o=IBM, c=US

Note that in the explicit ACLs above, aclSource is the same as the entry DN. This attribute is generated and managed by the LDAP server; it cannot be set when modifying ACLs.

Following is an implicit ACL for entry cn=personA, ou=deptXYZ, o=IBM, c=US:

aclPropagate: TRUE
aclEntry: group:cn=IBMRegs, o=IBM, c=US:normal:rcs:sensitive:rsc
aclEntry: group:cn=Anybody:normal:rsc
aclSource: o=IBM, c=US

In this example, a propagating ACL has been set on o=IBM, c=US. An override ACL has been set (aclPropagate is FALSE) on the descendant ou=deptXYZ, o=IBM, c=US. Therefore, the ACL set at ou=deptXYZ, o=IBM, c=US pertains only to that particular entry.

The descendant cn=personA, ou=deptXYZ, o=IBM, c=US inherits its ACL value from the nearest ancestor with a propagating ACL (which is o=IBM, c=US as reflected in the aclSource). The ACL on ou=deptXYZ, o=IBM, c=US is not used because aclPropagate is FALSE.

Other examples
In these examples, the administrator DN will be cn=admin, c=US.

The following example shows the default ACL:

aclPropagate: TRUE
aclEntry: group:cn=Anybody:normal:rsc:system:rsc
aclSource: default
ownerPropagate: TRUE
entryOwner: access-id:cn=admin,c=US
ownerSource: default

The following example shows a typical ACL for entry cn=personA, ou=deptXYZ, o=IBM, c=US:

aclPropagate: TRUE
aclEntry: group:cn=deptXYZRegs, o=IBM, c=US:normal:rcs:sensitive:rsc
aclEntry: group:cn=Anybody:normal:rsc:system:rsc
aclSource: ou=deptXYZ, o=IBM, c=US
ownerPropagate: TRUE
entryOwner: access-id:cn=deptXYZMgr, ou=deptXYZ, o=IBM, c=US
ownerSource: ou=deptXYZ, o=IBM, c=US

This is an inherited ACL and an inherited owner. Both owner properties and ACL properties are inherited from entry ou=deptXYZ, o=IBM, c=US. In this example, members of group cn=deptXYZRegs, o=IBM, c=US have permission to read, search and compare attributes in both the normal and sensitive attribute access classes. They do not have permission to add or delete entries under this entry. Nor do they have permission to access any information or change any information on attributes in the critical attribute access class. Unauthenticated, as well as all other bound users, have permission to read, search, and compare attributes in the normal and system attribute access classes only. The personA has add and delete permission on the entry; read, write, search, and compare permissions on normal and sensitive
attributes; and read, search, and compare permission on critical attributes. The deptXYZMgr had full access to the entry since it is the owner of the entry. As always, the administrator also has unrestricted access to the entry.

Access control groups

Access control groups provide a mechanism for applying the same aclEntry or entryOwner attribute values to an entry for multiple users without having to create an explicit aclEntry or entryOwner for each user.

For the LDBM, CDBM, and TDBM backends, the following object classes are evaluated as access control group entries: accessGroup, accessRole, groupOfNames, groupOfUniqueNames, ibm-staticGroup, groupOfUrls, ibm-dynamicGroup, and ibm-nestedGroup. See Chapter 21, “Static, dynamic, and nested groups” for more information on static, dynamic, and nested groups.

Associating DNs and access groups with a bound user

After a successful bind request, a bind distinguished name is associated with the bound user. Depending upon the bind method, there can also be a list of alternate DNs associated with the bound user.

- For a simple bind, the bind DN is the DN specified in the bind request. There must be an entry in LDAP with that DN. The entry can be in an LDBM, SDBM, CDBM, or TDBM backend, or in a client operation plug-in extension. There are no alternate DNs.

- For a CRAM-MD5 bind, the bind request must specify a DN or a username. If a DN is specified, there must be an entry in LDAP with that DN. If a username is specified, there must be an entry in LDAP that contains the username as a uid attribute value. If both a DN and a username are specified, there must be an entry in LDAP with that DN and the username must be a uid attribute value in that entry. In all of these cases, the bind DN is the DN of the entry. The entry can be in an LDBM, CDBM, or TDBM backend, or in a client operation plug-in extension. There are no alternate DNs. See Chapter 19, “CRAM-MD5 and DIGEST-MD5 authentication,” for more information.

- For a DIGEST-MD5 bind, the bind request must specify a username and may optionally contain an authorization DN. If only a username is specified, there must be an entry in LDAP that contains the username as a uid attribute value. If both a username and an authorization DN are specified, there must be an entry in LDAP with the authorization DN as its DN and the username must be a uid attribute value in that entry. In both cases, the bind DN is the DN of the entry. The entry can be in a LDBM, CDBM, or TDBM backend or in a client operation plug-in extension. There are no alternate DNs. See Chapter 19, “CRAM-MD5 and DIGEST-MD5 authentication,” for more information.

- For a Kerberos bind, the bind DN is ibm-krb=principal@REALM where principal@REALM is the Kerberos identity specified in the bind request. There can not be an entry in a LDBM, CDBM, or TDBM backend or in a client operation plug-in extension corresponding to this DN. Each LDBM, TDBM, CDBM, and SDBM backend for which krbidentityMap is specified in the LDAP server configuration file is contacted to map the Kerberos identity to alternate DNs in that backend. The client operation plug-in extensions are also contacted if they have registered a SLAPI_TYPE_ALT_NAMES callback type routine and can contribute alternate DNs. See Chapter 17, “Kerberos authentication,” for more information.

- For a certificate (EXTERNAL) bind, the bind DN is normally the subject DN from the certificate specified in the bind request. There can not be an entry in an LDBM, CDBM, or TDBM backend or in a client operation plug-in extension corresponding to this DN. If there is a RACF user associated with the certificate and replace is specified for the sslMapCertificate option in the LDAP server configuration file, then an SDBM-style DN based on the RACF user name replaces the subject DN as the bind DN. If add is specified for the sslMapCertificate option, then the bind DN is not changed and the SDBM-style DN based on the RACF user name is added as an alternate DN. See “Support of certificate bind” on page 63 for more information.
After the bind DN and alternate DNs (if any) associated with the bound user are determined, the DNs of the groups that the bound user belongs to are also added to the bind information. The bind DN and group information are used in access control in LDAP operations from the bound user.

**Note:** Group gathering is not performed if any of the following is true:

1. The user binds as the **adminDN**, **peerServerDN**, or **masterServerDN**.
2. The **authenticateOnly** server control is specified as part of the bind request.

The groups are gathered in the following manner:

- The backend or client operation plug-in extension that contains the bind DN is contacted to contribute DNs of any group entries that contain the bind DN or any of the alternate DNs. If the bind DN is not in a backend or a client operation plug-in extension (for example, after a Kerberos bind), this step is skipped.
- Each LDBM, CDBM, or TDBM backend that has **extendedGroupSearching on** specified in the LDAP server configuration file is also contacted to contribute the DNs of any group entries in the backend that contain the bind DN or any of the alternate DNs. The client operation plug-in extensions are also contacted to contribute group DNs if they have registered a **SLAPI_TYPE_GROUPS** callback type routine. Note that SDBM does not support extended group searching.

**Deleting a user or a group**

Deleting a user or a group does not have any cascade effect on any **aclEntry** and **entryOwner** values that include that user or group. The user or group DN is not removed from the ACLs. If another user or group is subsequently created with the same DN, that user or group will be granted the privileges of the former user or group.

**Retrieving ACL information from the server**

In order to retrieve all of the ACL information in a namespace, use the **ldapsearch** command, as shown in the following example:

```
ldapsearch -h 127.0.0.1 -D "cn=admin, dc=Your Company,dc=com" -wxxxxxx
-b "dc=Your Company,dc=com" "(objectclass=*)" aclEntry aclPropagate aclSource
entryOwner ownerPropagate ownerSource
dn: dc=Your Company, dc=com
aclPropagate: TRUE
aclEntry: CN=ANYBODY:normal:rsc:system:rsc
aclSource: dc=Your Company, dc=com
ownerPropagate: TRUE
entryOwner: CN=ADMIN
ownerSource: default
```

This command performs a subtree search starting at the root of the tree (assuming the root of the tree is "dc=Your Company, c=com") and returns the six ACL attributes for each entry in the tree. It is necessary to specifically request the six ACL attributes because they are considered as "operational" and, therefore, can only be returned on a search if requested. (See IETF RFC 2251, [RFC 2251: Lightweight Directory Access Protocol (v3)](https://tools.ietf.org/html/rfc2251)).

ACL information (**aclEntry**, **aclPropagate**, **aclSource**, **entryOwner**, **ownerPropagate**, and **ownerSource**) is returned for all entries. For those entries that contain ACLs, the **aclSource** and **ownerSource** attributes contain the same DN as the entry DN. For those entries that do not contain ACLs, the **aclSource** and **ownerSource** attributes contain distinguished names of the entries that contain the ACL information (**aclEntry** and **entryOwner**) that are used for access control checking of information in that entry.
Notes:

1. It is possible for the aclSource and ownerSource attributes to contain the value default. This is not a distinguished name but rather represents that the ACL that applies to the entry is the default ACL.

2. If the tree is larger than the sizeLimit option in the LDAP server configuration file or on the search command, then not all entries are returned. See the sizeLimit configuration option in Chapter 8, Customizing the LDAP server configuration for more information.

You can also use the same method to get the ACL information for a portion of the namespace by specifying the -b searchbase parameter on the search command, where searchbase is the starting point for the search.

Creating and managing access controls

To create and update ACLs in LDBM, TDBM, CDBM, GDBM, or the schema entry, use a tool implementing ldap_modify APIs, such as ldapmodify. The ldapmodify utility allows creation, modification, and deletion of any set of attributes that are associated with an entry in the directory. Since access control information is maintained as a set of additional attributes within an entry, ldapmodify is a natural choice for administering access control information in LDBM, TDBM, CDBM, GDBM, or the schema entry.

See IBM Tivoli Directory Server Client Programming for z/OS for details on using the utilities, such as ldapmodify.

Creating an ACL

In order to create an ACL, the aclEntry and aclPropagate attributes must be added to the information stored for an entry. The aclEntry and aclPropagate attributes are added to an entry by either specifying them as part of the entry information when the entry is added to the directory or by modifying the entry after it exists to contain the aclEntry and aclPropagate information.

It is possible to create an ACL without specifying the aclPropagate attribute. In this case, the aclPropagate attribute is assumed to have a value of TRUE and is added into the directory entry automatically, along with the aclEntry information.

Since the ldapmodify utility is very powerful, all the possible ways of adding the aclEntry and aclPropagate information cannot be shown here. The examples shown here describe the more common uses of the ldapmodify utility to add ACL information.

Figure 29 shows how to add a propagating ACL with three aclEntry values to an existing entry replacing any current aclEntry value.

```
$ ldapmodify -h 127.0.0.1 -D "cn=admin" -w xxxx -f newAcl.ldap
```

Where newAcl.ldap contains:

```
dn: cn=tim, o=Your Company
changetype: modify
replace: aclEntry
aclEntry: cn=jeanne, o=Your Company:
  normal:rsc:sensitive:rsc:critical:rsc
aclEntry: cn=jeff, cn=tim, o=Your Company: normal:rsc
aclEntry: cn=tim, o=Your Company:
  normal:rwsc:sensitive:rwsc:critical:rwsc
```

Figure 29. Example of adding propagating ACL to existing entry in directory

The ACL added in Figure 29 is created as a propagating ACL since the aclPropagate attribute is not specified and so assumed to be TRUE. This means that the ACL will apply to all entries below cn=tim, o=Your Company that do not already have an ACL associated with them. Note that the first and last
aclEntry values span two lines in the newAcl.ldif file. In order to do this, the first character on the continued line must be a space character, as shown in the example.

While it is not required that the administrator update all ACL information, the examples in this section all use the administrator when updating ACLs. Further, the use of -h 127.0.0.1 implies that the ldapmodify commands are performed from the same system on which the LDAP server is running and that the LDAP server is listening on TCP/IP port 389. Refer to the ldapmodify command description in [IBM Tivoli Directory Server Client Programming for z/OS] for more details on the -h, -p, -D, and -w command-line options. The ACL attributes can be updated from any LDAP client as long as the user performing the updates has the proper authorization to update the ACL information.

The ACL attributes are defined to be in a special access class called restricted. Therefore, in order to allow someone other than the LDAP administrator to update the ACL attributes, they must either be the entry owner or have the proper authorization to restricted attributes. Figure 30 shows an example of adding an ACL so that cn=jeanne, o=Your Company can update the ACL information:

```bash
$ ldapmodify -h 127.0.0.1 -D "cn=admin" -w xxxx -f newAcl.ldif
```

Where newAcl.ldif contains:
```
dn: cn=jeanne, o=Your Company
changetype: modify
replace: aclEntry
aclEntry: cn=jeanne, o=Your Company:
  normal:rsc:sensitive:rsc:critical:rsc:restricted:rwsc
aclEntry: cn=jeff, cn=tim, o=Your Company:normal:rsc
aclEntry: cn=tim, o=Your Company:
  normal:rsc
-
add: aclPropagate
aclPropagate: TRUE
```

Figure 30. Example of adding propagating ACL to existing entry in the directory.

The ACL added in Figure 30 allows cn=jeanne, o=Your Company to update the ACL information for this entry. In addition, since the ACL is a propagating ACL, this allows cn=jeanne, o=Your Company to create new ACL information against any entry that is controlled by this ACL. Care must be taken here, however, since it is possible for cn=jeanne, o=Your Company to set up an ACL which then does not allow cn=jeanne, o=Your Company update capability on the ACL information. If this occurs, a user with sufficient authority (the administrator, for example) must be used in order to reset/change the ACL information.

Figure 31 shows an example of adding a non-propagating ACL. A non-propagating ACL applies only to the entry to which it is attached and not to the subtree of information that might be stored below the entry in the directory.
Setting up a non-propagating ACL is similar to setting up a propagating ACL. The difference is that the `aclPropagate` attribute value is set to `FALSE`.

Modifying an ACL

Once an ACL exists for an entry in the directory, it may have to be updated. To do this, the `ldapmodify` command is used. The examples in this section use the `ldapmodify` command, however, any LDAP client application issuing LDAP modify operations to the LDAP server may be used. Therefore, modifications to ACL information need not be performed from the same system on which the LDAP server is running.

Modifications to ACLs can be of a number of different types. The most common modifications are to:

- Add an additional `aclEntry` value to the ACL to allow another person or group access to the entry
- Change an ACL from propagating to non-propagating (not permitted for the GDBM change log suffix, `cn=changelog`)
- Remove an `aclEntry` value which exists in the ACL to disallow another person or group access to the entry that they had before.

Figure 32, Figure 33, and Figure 34 show examples of these modifications, respectively.

Access determination shows how an additional `aclEntry` value is added to existing ACL information.

```
$ ldapmodify -h 127.0.0.1 -D "cn=admin" -w xxxx -f newAcl.ldif
```

Where `newAcl.ldif` contains:
```
dn: cn=jeff, cn=tim, o=Your Company
changetype: modify
replace: aclEntry
  aclEntry: cn=tim, o=Your Company:normal:rwsc:sensitive:rwsc:
  critical:rwsc:restricted:rwsc
  aclEntry: cn=jeff, cn=tim, o=Your Company:normal:rwsc:
  sensitive:rwsc:critical:rwsc:
  aclEntry: cn=jeanne, o=Your Company:normal:rsc
-
  replace: aclPropagate
  aclPropagate: FALSE
-
```

Figure 31. Example of setting up a non-propagating ACL

Setting up a non-propagating ACL is similar to setting up a propagating ACL. The difference is that the `aclPropagate` attribute value is set to `FALSE`.

```
$ ldapmodify -h 127.0.0.1 -D "cn=admin" -w xxxx -f modAcl.ldif
```

Where `modAcl.ldif` contains:
```
dn: cn=jeff, cn=tim, o=Your Company
changetype: modify
add: aclEntry
  aclEntry: cn=dylan, cn=tim, o=Your Company:
-`
```

Figure 32. Example of adding an aclEntry attribute value

In Figure 32 `cn=dylan, cn=tim, o=Your Company` is granted permissions against the `cn=jeff, cn=tim, o=Your Company` entry in the directory. The existing ACL information remains in the entry; the `aclEntry` attribute value for `cn=dylan, cn=tim, o=Your Company` is added to this information.

Figure 33 shows how to modify an existing ACL to be non-propagating instead of propagating.
In Figure 33, the existing ACL against `cn=tim, o=Your Company` is modified to be a non-propagating ACL instead of a propagating ACL. This means that the ACL will no longer apply to entries below `cn=tim, o=Your Company` in the directory tree. Instead, the first propagating ACL that is found in an entry above `cn=tim, o=Your Company` will be applied to the entries below `cn=tim, o=Your Company`. If no propagating ACL is found in the entries above `cn=tim, o=Your Company`, then the default ACL is used.

Figure 34 shows how to remove an `aclEntry` value from existing ACL information:

```
$ ldapmodify -h 127.0.0.1 -D "cn=admin" -w xxxx -f modAcl.ldif
```

Where `modAcl.ldif` contains:
```
dn: cn=tim, o=Your Company
changetype: modify
replace: aclPropagate
aclPropagate: FALSE
```

Figure 33. Example of modifying aclPropagate attribute

In Figure 33, the existing ACL against `cn=tim, o=Your Company` is modified to be a non-propagating ACL instead of a propagating ACL. This means that the ACL will no longer apply to entries below `cn=tim, o=Your Company` in the directory tree. Instead, the first propagating ACL that is found in an entry above `cn=tim, o=Your Company` will be applied to the entries below `cn=tim, o=Your Company`. If no propagating ACL is found in the entries above `cn=tim, o=Your Company`, then the default ACL is used.

```
$ ldapmodify -h 127.0.0.1 -D "cn=admin" -w xxxx -f modAcl.ldif
```

Where `modAcl.ldif` contains:
```
dn: cn=jeff, cn=tim, o=Your Company
changetype: modify
delete: aclEntry
```

Figure 34. Example of removing a single aclEntry attribute value

In Figure 34, the `aclEntry` attribute value for `cn=dylan, cn=tim, o=Your Company` is removed from the ACL information for entry `cn=jeff, cn=tim, o=Your Company`. Only the distinguished name part of the `aclEntry` value needs to be specified when deleting the value.

Deleting an ACL

In order to delete an ACL that is attached to an entry in the directory, the `aclEntry` and `aclPropagate` attributes must be deleted from the entry. To do this, use the `ldapmodify` command to delete the entire attribute (all values) from the entry.

Note: This is not allowed for the change log suffix entry, `cn=changeLog`, when GDBM is DB2-based.

Figure 35 shows an example of deleting an ACL from an entry.

```
$ ldapmodify -h 127.0.0.1 -D "cn=admin" -w xxxx -f delAcl.ldif
```

Where `delAcl.ldif` contains:
```
dn: cn=jeff, cn=tim, o=Your Company
changetype: modify
delete: aclEntry
```

Figure 35. Example of deleting an ACL from an entry

In Figure 35, the existing ACL against `cn=jeff, cn=tim, o=Your Company` is removed. This means that the ACL will no longer apply to the entry. Instead, the first propagating ACL that is found in an entry above `cn=jeff, cn=tim, o=Your Company` will be applied to `cn=jeff, cn=tim, o=Your Company`. If no propagating ACL is found in the entries above `cn=jeff, cn=tim, o=Your Company`, then the default ACL is used.
Creating an owner for an entry

In addition to the access control list control of directory entries, each entry can have assigned to it an entry owner or set of entry owners. As an entry owner, full access is allowed to the entry. Entry owners are granted add and delete permission, as well as read, write, search, and compare for all attribute classes. Entry owners can add and modify ACL information on the entries for which they are specified as the owner.

Entry owners are listed in the entryOwner attribute. Just like aclEntry information, entryOwner information can be propagating or non-propagating based on the setting of the ownerPropagate attribute. Like the aclSource attribute for aclEntry information, the ownerSource attribute lists the distinguished name of the entry that contains the entryOwner attribute which applies to the entry. The ownerSource attribute is set by the server and cannot be directly set when modifying the ACLs.

In order to create an entry owner, the entryOwner and ownerPropagate attributes must be added to the information stored for an entry. The entryOwner and ownerPropagate attributes are added to an entry by either specifying them as part of the entry information when the entry is added to the directory or by modifying the entry after it exists to contain the entryOwner and ownerPropagate information.

It is possible to create an entry owner without specifying the ownerPropagate attribute. In this case, the ownerPropagate attribute is assumed to have a value of TRUE and is added into the directory entry automatically.

Since the ldapmodify command is very powerful, all the possible ways of adding the entryOwner and ownerPropagate information cannot be shown here. The examples shown here describe the more common uses of the ldapmodify command to add entry owner information.

Figure 36 shows how to add a propagating entry owner with two entryOwner values to an existing entry.

```
$ ldapmodify -h 127.0.0.1 -D "cn=admin" -w xxxx -f newOwn.ldif
```

Where newOwn.ldif contains:
```
dn: cn=tim, o=Your Company
changetype: modify
replace: entryOwner
entryOwner: cn=joe, o=Your Company
entryOwner: cn=carol, o=Your Company
```

Figure 36. Example of adding a propagating set of entry owners to existing entry in the directory

The entry owners added in Figure 36 are created as a propagating set of entry owners since the ownerPropagate attribute is not specified and so assumed to be TRUE. This means that the entry owners will apply to all entries below cn=tim, o=Your Company that do not already have an entry owner associated with them.

While it is not required that the LDAP administrator update all entry owner information, the examples in this section all use the administrator as the entry owner updating ACLs. Further, the use of -h 127.0.0.1 implies that the ldapmodify commands are performed from the same system on which the LDAP server is running and that the LDAP server is listening on TCP/IP port 389. Refer to the ldapmodify command description in IBM Tivoli Directory Server Client Programming for z/OS for more details on the -h, -p, -D, and -w command-line options. The entry owner attributes can be updated from any LDAP client as long as the user performing the update has the proper authorization to update the entry owner information.

The entry owner attributes, like the ACL attributes, are defined to be in a special access class called restricted. Therefore, in order to allow someone other than the LDAP administrator to update the entry
owner attributes, they must either be the entry owner or have the proper authorization to restricted attributes. See Figure 30 for an example of allowing users other than the LDAP administrator the ability to update entry owner information.

Figure 37 shows an example of adding a non-propagating entry owner. A non-propagating entry owner applies only to the entry to which it is attached and not to the subtree of information that might be stored below the entry in the directory.

$ ldapmodify -h 127.0.0.1 -D "cn=admin" -w xxxx -f newOwn.ldif

Where newOwn.ldif contains:

dn: cn=jeff, cn=tim, o=Your Company
changetype: modify
replace: entryOwner
entryOwner: cn=george, o=Your Company
entryOwner: cn=jane, o=Your Company
-
replace: ownerPropagate
ownerPropagate: FALSE
-

Figure 37. Example of setting up a non-propagating entry owner

Setting up a non-propagating entry owner is similar to setting up a propagating entry owner. The difference is that the ownerPropagate attribute value is set to FALSE.

Modifying an owner for an entry

Once an entry owner exists for an entry in the directory, it may have to be updated. To do this, the ldapmodify command is used. The examples in this section use the ldapmodify command, however, any LDAP client application issuing LDAP modify operations to the LDAP server may be used. Therefore, modifications to entry owner information need not be performed from the same system on which the LDAP server is running.

Modifications to entry owners can be of a number of different types. The most common modifications are to:

- Add an additional entryOwner value to the set of entry owners to allow another person or group to control the entry
- Change an entry owner from propagating to non-propagating (not permitted for the GDBM change log suffix, cn=changelog)
- Remove an entryOwner value which exists in the entry owner set to disallow another person or group access to control the entry that they had control over before.

Figure 38, Figure 39, and Figure 40 show examples of these modifications, respectively.

Figure 38 shows how an additional entryOwner value is added to existing entry owner information.

$ ldapmodify -h 127.0.0.1 -D "cn=admin" -w xxxx -f modOwn.ldif

Where modOwn.ldif contains:

dn: cn=jeff, cn=tim, o=Your Company
changetype: modify
add: entryOwner
entryOwner: cn=jeff, o=Your Company
-

Figure 38. Example of adding an entryOwner attribute value
In Figure 38, cn=george, o=Your Company is granted entry owner control of the cn=jeff, cn=tim, o=Your Company entry in the directory. The existing entry owner information remains in the entry; the entryOwner attribute value for cn=george, o=Your Company is added to this information.

Figure 39 shows how to modify an existing entry owner to be non-propagating instead of propagating.

$ ldapmodify -h 127.0.0.1 -D "cn=admin" -w xxxx -f modOwn.ldif

Where modOwn.ldif contains:

```
dn: cn=tim, o=Your Company
changeType: modify
replace: ownerPropagate
ownerPropagate: FALSE
```

Figure 39. Example of modifying the ownerPropagate attribute

In Figure 39, the existing entry owner set for cn=tim, o=Your Company is modified to be non-propagating instead of propagating. This means that the entry owner will no longer apply to entries below cn=tim, o=Your Company in the directory tree. Instead, the first propagating entry owner set that is found in an entry above cn=tim, o=Your Company will be applied to the entries below cn=tim, o=Your Company. If no propagating entry owner is found in the entries above cn=tim, o=Your Company, then the default entry owner is used.

Figure 40 shows how to remove an entryOwner value from existing entry owner information:

$ ldapmodify -h 127.0.0.1 -D "cn=admin" -w xxxx -f modOwn.ldif

Where modOwn.ldif contains:

```
dn: cn=jeff, cn=tim, o=Your Company
changeType: modify
delete: entryOwner
entryOwner: cn=george, cn=tim, o=Your Company
```

Figure 40. Example of removing a single entryOwner Attribute value

In Figure 40, the entryOwner attribute value for cn=george, cn=tim, o=Your Company is removed from the entry owner information for entry cn=jeff, cn=tim, o=Your Company. Only the distinguished name part of the entryOwner value needs to be specified when deleting the value.

Deleting an owner for an entry

In order to delete an entry owner set that is attached to an entry in the directory, the entryOwner and ownerPropagate attributes must be deleted from the entry. To do this, use the ldapmodify command to delete the entire attribute (all values) from the entry.

**Note:** This is not allowed for the change log suffix entry, cn=changelog, when GDBM is DB2-based.

Figure 41 shows an example of deleting an entry owner set from an entry.
In Figure 41, the existing entry owner set against cn=jeff, cn=tim, o=Your Company is removed. This means that the entry owner information will no longer apply to the entry. Instead, the first propagating entry owner set that is found in an entry above cn=jeff, cn=tim, o=Your Company will be applied to cn=jeff, cn=tim, o=Your Company. If no propagating entry owner set is found in the entries above cn=jeff, cn=tim, o=Your Company, then the default entry owner is used.

Creating a group for use in ACLs and entry owner settings

Sets of users can be grouped together in the directory by defining them as members of a group in the directory. A directory group, used for access control checking, is just another entry in the directory. A static, dynamic, or nested group entry can be used as a group on the aclEntry or entryOwner attributes. See Chapter 21, “Static, dynamic, and nested groups” for more information on creating, modifying, and deleting static, dynamic, and nested group entries.

When defining access controls or entry owner sets, names of group entries can be used in the same place as user entry names. When access control decisions are performed, a user’s group memberships can be used in determining if a user can perform the action requested.

Groups are added to access control information in just the same way as user entries are added to access control information. Figure 42 shows how a group can be added to the aclEntry information in an existing access control specification for an entry. Figure 43 shows how a group can be added as an entryOwner to an existing entry owner specification for an entry.

$ ldapmodify -h 127.0.0.1 -D "cn=admin" -w xxxx -f delOwn.ldif

Where delOwn.ldif contains:

dn: cn=jeff, cn=tim, o=Your Company
changetype: modify
delete: entryOwner
 -
delete: ownerPropagate
 -

Figure 41. Example of deleting an entry owner set from an entry

$ ldapmodify -h 127.0.0.1 -D "cn=admin" -w xxxx -f modAcl.ldif

Where modAcl.ldif contains:

dn: cn=jeff, cn=tim, o=Your Company
changetype: modify
add: aclEntry
aclEntry: group:cn=group1, o=Your Company: normal: rwsc: sensitive: rsc
 -

Figure 42. Example of adding a group to access control information

$ ldapmodify -h 127.0.0.1 -D "cn=admin" -w xxxx -f modOwn.ldif

Where modOwn.ldif contains:

dn: cn=jeff, cn=tim, o=Your Company
changetype: modify
add: entryOwner
entryOwner: cn=group1, o=Your Company
 -

Figure 43. Example of adding a group to entry owner information
Chapter 23. Basic replication

Once the z/OS LDAP server is installed and configured, users can access the directory, add entries, delete entries, or perform search operations to retrieve particular sets of information.

Replication is a process which keeps multiple directories in sync. 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.

There are several benefits realized through replication. The single greatest benefit is providing a means of faster searches. Instead of having all search requests directed at a single server, the search requests can be spread among several different servers. This improves the response time for the request completion.

Additionally, the replica provides a backup to the replicating server. Even if the replicating server crashes, or is unreadable, the replica can still fulfill search requests, and provide access to the data.

There are two types of basic replication:

- In peer to peer replication, each LDAP peer server is a read-write server. Updates processed on one peer server are replicated to all the other peer servers. Peer servers are read-write to all users.

  Note: The basic replication support for peer to peer replication is provided for failover support purposes. With basic peer to peer replication, there is no support for resolving simultaneous updates on multiple peer servers, which can cause a failure of replication. As a result, updates should be targeted to one peer server at a time.

- In basic read-only replication, a single read-write LDAP server (the master) replicates the updates it processes to a set of read-only replica servers.

  Master
  All changes to the directory are made to the master server. The master server is then responsible for propagating the changes to all other directories. It is important to note that while there can be multiple directories representing the same information, only one of those directories can be the master.

  Read-only replica
  Each of the additional servers which contain a directory replica. These replica directories are identical to the master directory. These servers are read-only to all users and will only accept updates from their master server.

If you need more advanced replication choices, see Chapter 24, "Advanced replication," on page 377.

Note: Basic and advanced replication are not allowed in the same server. If both are enabled in the server, the server fails to start.

A basic replication network can contain both peer replica servers and read-only replica servers. In this case, each peer server must act as a master to each read-only replica (in addition to being a peer to all the peer servers), so that updates that occur on any peer server are replicated to all the other peer and read-only replicas in the network.

Basic replication is supported when the servers involved are running in single-server or in multi-server mode. See Determining operational mode for more information about server operating modes.

In z/OS LDAP, basic replication is supported in LDBM and TDBM backends. Basic replication is not performed for the SDBM or GDBM backends or for the schema entry.
Basic replication in a sysplex

A set of LDAP servers sharing a backend directory in a sysplex can act as a master, read-only, or peer server to LDAP servers that are not in the sysplex. Each LDAP server in the sysplex must have the same replication options (masterServer, masterServerDN, masterServerPW, peerServerDN, and peerServerPW) in the backend section of the LDAP server configuration file. Do not make an LDAP server in the sysplex a replica of another LDAP server in the sysplex for a backend directory that they are sharing.

When the set of LDAP servers in the sysplex is set up to be a master or a peer replica server and changes occur to the shared LDAP directory, the LDAP server acting as the owner of the sysplex group replicates the directory changes to the LDAP server replicas that are not in the sysplex. These replicas are identified by replica entries in the shared directory.

When the set of LDAP servers in the sysplex is set up to be a read-only or a peer replica server and directory changes occur to a master or peer LDAP server that is not in the sysplex, that LDAP server replicates the changes to an LDAP server in the sysplex, identified in a replica entry in the directory. The changes are made to the backend directory in the sysplex and are seen by all the LDAP servers sharing the directory in the sysplex.

ibm-entryuuid replication

Basic replication of the ibm-entryuuid attribute is performed to any LDAP server that has 1.3.18.0.2.32.3 (the OID for the entry UUID capability) as a value in the ibm-enabledCapabilities attribute in the root DSE. z/OS LDAP servers have this capability. If the root DSE of a replica server does not contain the required capability, then the ibm-entryuuid attribute will not be replicated to that server, however, the entry and other attributes will be replicated.

Complex modify DN replication

Basic replication of Modify DN new superior operations will be performed to any LDAP server that has 1.3.18.0.2.32.33 (the OID for the subtree move capability) or 1.3.18.0.2.32.34 (the OID for the subtree rename capability) as a value in the ibm-enabledCapabilities attribute in the root DSE. z/OS LDAP servers have this capability. If a replica server is not at a supported level, Modify DN new superior operations will fail until the replica is removed from the replica collection.

Basic replication and ldif2ds

ldif2ds does not replicate changes when adding entries to the replicating server. If you are using ldif2ds to add entries to a replicating server you must also use it to add entries to each replica, with no intervening updates on the replicating server before the replica is loaded.

In order to maintain directory integrity, a load utility (ldif2ds) should be used on a read-only or peer replica only when initially populating the replica directory. If a load utility is used to add entries to a replica server after initial population, these changes are not reflected in the master directory. The replica directory might give erroneous information.

Password encryption and basic replication

To ensure data integrity and the correct working of the LDAP servers in the replication environment, the pwEncryption option in the configuration files for the servers involved in replication must be the same. If one of the servers involved in replication is a non-z/OS server, then the administrator must select a pwEncryption method that is supported by both servers for correct operation of replication. If no encryption methods are common between the servers, then password encryption should not be used.
When replicating between a z/OS LDAP server and a non-z/OS LDAP server and using crypt for password encryption, specify **pwCryptCompat off** in the backend section of the z/OS LDAP server configuration file. This setting indicates that the LDAP server should use the UTF-8 version of the crypt algorithm to encrypt passwords. When **userPassword** attribute values in crypt are replicated between z/OS and non-z/OS LDAP servers, the password will be the same on both platforms and therefore it will be usable.

If using AES or DES encryption and the key is stored in an LDAPKEYS dataset and both of the servers involved in replication are z/OS LDAP servers, the same key label and data key must be present in both server’s copy of the LDAPKEYS dataset. If the key is stored in ICSF CKDS, the same AES or DES key label and data key must be defined on both z/OS systems through the ICSF KGUP and CKDS facilities. For information about managing cryptographic keys, see [z/OS Cryptographic Services ICSF Administrator’s Guide](https://www.ibm.com/support/knowledgecenter/SSSLBS_8.5.1/com.ibm.doc.ezicfg.doc.pdf). The AES or DES key label is specified in the LDAP server configuration files of both of the LDAP servers involved in replication.

### Replicating server

In order for the basic replication process to occur, the following must happen:

- The replicating server (master or peer) must be aware of each replica that is to receive the change information.
- Each read-only and peer server must be aware of the replicating servers for the directory that it serves. See [LDAP update operations on read-only replicas](https://www.ibm.com/support/knowledgecenter/SSSLBS_8.5.1/com.ibm.doc.ezicfg.doc.pdf) for more information.

The replicating server becomes aware of the existence of the replica servers when entries with an object class of **replicaObject** are added to the directory. Each of these entries represents a particular replica server. The attribute/value pairs within the replica entry provide the information the replicating server needs in order to locate the replica server and send any updates to that server.

### Replica entries

The **replicaObject** object class is provided in the initial schema. Like other LDAP object class definitions, the **replicaObject** has mandatory and optional attributes. Each of the **replicaObject** attributes are single-valued. The following is a description of the mandatory attributes of **replicaObject**. Values in a replica entry are recognized at server startup and when a replica entry is added or modified. The internal number of how many replication operations have been set aside (the set aside count) for a replica is not reset when the replica entry is modified. In order to reset the count, either the server needs to be restarted or the replica entry needs to be removed and added. See [Basic replication error log](https://www.ibm.com/support/knowledgecenter/SSSLBS_8.5.1/com.ibm.doc.ezicfg.doc.pdf) for more information about the set aside count.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description and example</th>
</tr>
</thead>
</table>
| replicaHost    | This can be an IPv4 address, IPv6 address, or a hostname of the system where the replica server is running. Example:  
replicahost: 9.130.77.27  
replicahost: [5f1b:df00:ce3e:e200:20:800:2078:e3e3]  
replicahost: myMachine.ibm.com |
| replicaBindDN  | Specifies the LDAP distinguished name that the replicating server uses to bind to the replica when sending directory updates. The replicaBindDN and the masterServerDN or peerServerDN in the replica’s LDAP server configuration file must have the same value. Example:  
replicaBindDN: cn=Master |
<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description and example</th>
</tr>
</thead>
</table>
| replicaCredentials  | Contains the authentication information needed for the replicating server to authenticate to the replica using the **replicaBindDN**. The **replicaCredentials** attribute value will be encrypted if the secretEncryption option is specified in the LDAP server configuration file. This improves directory security because the bind password is no longer stored in the directory in clear text. The secretEncryption option is also used to encrypt pending updates while they are stored in the replication queue.  
Example: replicaCredentials: secret  |
| cn                   | Forms the RDN of the LDAP distinguished name of the **replicaObject** entry.  
Example: cn: myReplica  |

In the examples in [Table 55](#), when the replicating server receives and successfully finishes an update request, the update is also sent to myMachine.ibm.com on port 389 (the default port). The replicating server performs a bind operation using the DN of cn=Master and password of secret. See [Establishing the administrator DN and basic replication replica server DN and passwords](#) for more information specifying the replication server DN and password.

In addition, there are several attributes available that provide additional flexibility in configuring a replica server. For instance, an added description might better describe the replica server, and it might listen on a different port than the default port of 389. Examples of adding a description and changing the port to 400 are shown in [Table 56](#) which describes the optional attributes of **replicaObject**.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description and example</th>
</tr>
</thead>
</table>
| replicaPort          | Describes the port number on which the replica is listening for incoming requests. By default, the server listens on port 389.  
Example: replicaPort: 400  |
| replicaUpdateTimeInterval | Delays the propagation of additional updates for specified number of seconds. The default is for the replicating server to send updates immediately.  
Example: replicaUpdateTimeInterval: 3600 |
| replicaUseSSL        | Determines whether the replicating server should replicate over SSL/TLS. The default is to replicate without using SSL/TLS.  
Example: replicaUseSSL: TRUE |
| description          | Provides an additional text field for extra information pertaining to the replica entry.  
Example: description: Replica machine in the fourth floor lab  |
| seeAlso              | Identifies another directory server entry that might contain information related to this entry.  
Example: seeAlso: cn=Alternate Code, ou=Software, o=IBM, c=US  |
Table 56. Replica entry schema definition (optional attributes) (continued)

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description and example</th>
</tr>
</thead>
<tbody>
<tr>
<td>replicaBindMethod</td>
<td>Identifies the bind method to be used. If it is specified, it must be set to simple. Example: replicaBindMethod: simple</td>
</tr>
</tbody>
</table>

Basic replication only supports simple authentication. SASL EXTERNAL, GSSAPI, DIGEST-MD5, and CRAM-MD5 bind mechanisms are not supported as valid basic replication bind mechanisms.

There are several additional attributes that affect error handling during basic replication. See Basic replication error log for more information about error handling. These attributes are not in any object class, therefore, the extensibleObject object class must included in the replica entry when adding these attributes to the entry. Table 57 describes these attributes.

Table 57. Additional optional replication attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description and example</th>
</tr>
</thead>
<tbody>
<tr>
<td>ibm-slapdLog</td>
<td>Specifies the file name of the basic replication error log. This must be a UNIX System Services file (not a dataset). The file name can be fully-qualified or can be relative to the current working directory of the LDAP server. The current working directory is set when the LDAP server is started to the HOME environment variable if specified, or else to /etc/ldap. This format is not recommended. The value must be unique among all the replica entries in this LDAP server. If this attribute is not present in the replica entry or it has no value, error logging and setting aside will not occur. Example: ibm-slapdLog: /home/replog/replica1.errlog</td>
</tr>
<tr>
<td>ibm-slapdReplMaxErrors</td>
<td>Specifies the maximum number of basic replication errors that will be set aside in the basic replication error log before replication is allowed to stall. If this attribute is not present in the replica entry or if the value is 0, then no operations are set aside. In this case, errors are still logged and basic replication stalls when the first error occurs. This attribute is not used if a replication log file name has not been specified with the ibm-slapdLog attribute. Example: ibm-slapdReplMaxErrors: 5</td>
</tr>
</tbody>
</table>

Adding replica entries in TDBM or LDBM

In TDBM or LDBM, replica entries can be placed anywhere within the directory tree, although it is recommended that a replica entry be a leaf entry. Placing replica entries in the directory tree then requires that any parent entries of the replica entry be added to the directory before adding the replica entry. These entries must be added to both the replicating server and replica server before addition of the replica entry. This is needed on the replica server because these entries are being added at the replicating server without replication being active. If a replica entry is not placed as a leaf node in the directory tree, the only entries allowed below the replica entry are other replica entries. The LDAP server allows non-replica entries to be placed below replica entries; however, these entries will not be replicated to the replica servers.

The replica entry defines a replica for the backend containing the entry. Any changes made to the directory tree managed by that backend will be replicated to each replica defined for that backend. The replica entry does not define replicas for other backends in the LDAP server; therefore, if changes to all LDBM and TDBM directory trees managed by the LDAP server are to be replicated, then each backend must contain the appropriate replica entries to define replication for that backend.
The following is an example of a replica entry definition using LDIF format.

```ldif
dn: cn=myReplica,o=Your Company
objectclass: replicaObject
objectclass: extensibleObject
cn: myReplica
replicaHost: myMachine.ibm.com
replicaBindDN: cn=Master
replicaCredentials: secret
replicaPort: 400
replicaUseSSL: FALSE
description: Replica machine in the fourth floor lab
ibm-slapdLog: rol.errlog
ibm-slapdRep1MaxErrors: 5
```

**Searching a replica entry**

Most of the attributes in a replica entry are operational attributes. When searching a replica entry, the operational attributes are not included in the output unless they are specified in the attributes to be returned. The following command searches for all replica entries in a suffix and returns the complete replica entries in LDIF format:

```bash
ldapsearch -h ldaphost -p ldapport -D binddn -w passwd -L -b "suffix" "objectclass=replicaObject" "=" replicaHost replicaBindDN replicaCredentials replicaPort replicaUpdateTimeInterval replicaUseSSL replicaBindMethod
```

**Displaying basic replication status**

The LDAP server DISPLAY REPLICAS operator modify command can be used to display information about the status of replication to each replica server. See Displaying performance information and server settings for a description of the DISPLAY REPLICAS output.

**Basic replication maintenance mode**

Maintenance mode is the LDAP server setup mode for basic replication. This mode restricts access to the backends in an LDAP server to allow replica backends to be primed for basic replication. Access to the backends is as follows:

- read-only replica backend: The masterServerDN for the replica and the adminDN have unrestricted access
- peer replica backend: The peerServerDN for the replica and the adminDN have unrestricted access
- non-replica backends (including the schema entry): The adminDN has unrestricted access. The masterServerDN and peerServerDN have no access outside of the backends which specify them.

Other users can bind to the LDAP server, but cannot access any entries within the server.

ACL checking is performed during search operations from masterServerDN and peerServerDN but not during update and compare operations. No ACL checking is done for any operations from adminDN. In addition, the adminDN has the capability in maintenance mode to modify attributes that are read-only and are typically only set by the LDAP server, such as ibm-entryuuid.

**Note:** The LDAP server schema entry is not part of any replica backend. When the LDAP server is not in maintenance mode, masterServerDN and peerServerDN can only update the LDAP server schema if the schema entry ACL permits them to. When in maintenance mode, they cannot update the LDAP server schema at all. adminDN can always update the schema.

Pending replication entries are replicated to the other replica servers, but updates performed when in maintenance mode are not replicated.

Specify the -m option on the server startup command to start the LDAP server in maintenance mode.
The LDAP server MAINTMODE operator modify command can be used to change from maintenance mode to normal mode while the LDAP server is running. It can also be used to put a running server into maintenance mode. The following command can be sent to the LDAP server from the SDSF or the operator's console. If the command is entered from SDSF, it must be preceded by a slash (/).

```
f dssrv,maintmode state
```

where `state` can be **on** to turn maintenance mode on or **off** to turn maintenance mode off (and turn normal mode on).

---

**Replica server**

Initialization, or population, of a replica directory requires several steps.

1. With basic replication, changes to the LDAP server schema entry on the replicating server are not replicated. A separate update of the LDAP server schema on the replica will be required each time the schema is updated on the replicating server.

2. Replica servers must support the LDAP Version 3 protocol.

**Populating a replica**

1. Either start the replica and replicating servers in maintenance mode or use the LDAP server MAINTMODE ON operator modify command on each of these LDAP servers to put these servers into maintenance mode.

2. Unload the replicating server's directory contents if there are any entries. For TDBM or LDBM, use the `ds2ldif` utility (see `ds2ldif utility`).

3. You should make sure the schema for the replica server is the same as the schema for the replicating server.
   - If the replica and replicating server are both z/OS servers, the schema can be unloaded from the replicating server using `ds2ldif` and reloaded into the replica by using the administrator DN to run `ldapmodify`.

4. Using the administrator DN, run `ldapadd` to add a single replica entry into the backend directory on the replicating server to identify the new replica being populated.
   - Note that in order to load the replica entry, it is also necessary to load any parent entries in the directory hierarchy in hierarchy order.

5. If the replicating server does not contain any entries, go to step 8.

6. Transport the LDIF file created in step 2 to the replica server's location.

7. Load the LDIF file from 6 into the replica server. This can be done using the administrator DN to run `ldapadd` to load the LDIF file. Alternatively for TDBM, the replica server can be stopped and then the `ldif2ds` utility used to load the LDIF file.

8. Configure the replica (see next section).

9. Stop the replica server (if it is running) and then restart it in maintenance mode. If it contains a replica entry that defines this server as a replica of itself, use the administrator DN to run `ldapdelete` to remove that entry.

10. Use the LDAP server MAINTMODE OFF command on the replica server and the replicating server to change these servers to normal mode.

**Configuring the replica**

The key to a successful replica configuration rests in ensuring that the values in the replica entry on the replicating server (master or peer) accurately represent the relevant values on the replica server (read-only or peer). Configuring the replica involves specifying appropriate LDAP server configuration file option values to identify:
• the IP address and port on which the replica server should listen for communication from the replicating server
• the type of connection expected by the replicating server when it communicates to the replica server, either over a non-secure or secure connection
• the DN and password used by the replicating server

The following table identifies the relationship between the attributes in the replica entry on a z/OS LDAP replicating server and the configuration options on an IBM replica server. The values specified for these options must be equivalent. An example of what is meant by equivalent is when the replica server is listening on all of its network interfaces, then **replicaHost** must specify either the corresponding hostname or an IP address of one of the addresses.

<table>
<thead>
<tr>
<th>Attribute in replica entry on replicating server</th>
<th>Corresponding replica server configuration option or command line parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>replicaHost</td>
<td>The hostname or IP address specified on the <strong>listen</strong> configuration option or the -l LDAP server command line parameter.</td>
</tr>
<tr>
<td>replicaPort</td>
<td>The port number that is specified on the <strong>listen</strong> configuration option or the -l LDAP server command line parameter.</td>
</tr>
<tr>
<td>replicaUseSSL</td>
<td>Use of ldaps:// in the prefix of the <strong>listen</strong> configuration option or the -l LDAP server command line parameter corresponds to <strong>TRUE</strong> for replicaUseSSL; use of ldap:// corresponds to <strong>FALSE</strong>.</td>
</tr>
<tr>
<td>replicaBindDn</td>
<td>masterServerDN or peerServerDN configuration option</td>
</tr>
<tr>
<td>replicaCredentials</td>
<td>masterServerPW or peerServerPW configuration option</td>
</tr>
</tbody>
</table>

**Notes:**
1. If the replica server is a non-IBM server, you should consult their documentation for parameters that correspond to the parameters mentioned in the above table.
2. The value of the **listen** configuration option or –l command line parameter is an LDAP URL. For additional information about the **listen** option, see Chapter 8, “Customizing the LDAP server configuration.”
3. It is recommended that the **masterServerDN** or **peerServerDN** be a DN that is dedicated specifically to replication. It should not be used for any other operations.
4. The **masterServer**, **masterServerDN**, **masterServerPW**, **peerServerDN**, and **peerServerPW** options must follow the **database** option for that backend in the LDAP server configuration file.
5. Usage of the **masterServerPW** or **peerServerPW** configuration option is strongly discouraged in production environments. See Establishing the administrator DN and basic replication replica server DN and passwords for alternatives.
6. The **replicaCredentials** attribute will be encrypted if the **secretEncryption** configuration option is specified. This improves directory security because the bind password is no longer stored in the directory in clear text. The **secretEncryption** configuration option is also used to encrypt pending updates while they are stored in the replication queue.

**LDAP update operations on read-only replicas**

Update operations, such as add, delete, modify, and rename, should not be performed against a read-only replica server. Changes must be made to the master server, which then propagates the change to the read-only replica.

If update operations are sent to a read-only replica server, the replica server returns a referral containing the value in the **masterServer** option in the backend section of the LDAP server configuration file on the replica. The client then redirects the request to the master server. After the master server makes the update, it propagates the change to the read-only replica server, binding as the **replicaBindDn** value in the replica entry corresponding to that replica server (the **replicaBindDn** value must match the **masterServerDN** value in the replica server configuration file).
Changing a read-only replica to a master

When using read-only basic replication, it might become desirable to change one of the read-only replicas to be the master. Perhaps the machine where the replica server is installed is being upgraded, and you want this replica to now be the master LDAP server.

The following procedure should be followed to change a read-only replica to a master:

1. If the read-only replica is out of sync with the master server, use the procedure described in Recovering from basic replication out-of-sync conditions.
2. Use the LDAP server MAINTMODE ON operator modify command on the master server and on the replica server to put them into maintenance mode.
3. Using the administrator DN, unload all the replica entries (entries that describe replica servers) from the master server. Use a search command similar to the one shown in Searching a replica entry to create LDIF output containing the replica entries for each suffix in the backend. In the LDIF output, remove the replica entry for the read-only replica that is going to become the master. If the master is going to become a read-only replica, add a replica entry for the master in LDIF format to the output.
4. Using the administrator DN, run ldapdelete to remove the replica entries from the master.
5. Using the administrator DN, run ldapadd to add the unloaded replica entries to the replica server.
6. Stop the master and replica server.
7. Remove the masterServer, masterServerDN, and masterServerPW options from the LDAP server configuration file on the replica.
8. If the original master is being eliminated, the database on the master is no longer needed.
   • For TDBM, drop the TDBM DB2 database. The SPUFI script used to create the DB2 database also contains the commands to drop the database.
   • For LDBM, remove all the files in the LDBM database directory. See the description of the databaseDirectory option in Chapter 8, “Customizing the LDAP server configuration” for more information about the location of these files.
9. If the original master is going to become a replica, add the masterServer, masterServerDN, and masterServerPW options to the LDAP server configuration file on the original master. The masterServer value must point to the new master. See Establishing the administrator DN and basic replication replica server DN and passwords for more information about alternatives to specifying the masterServerPW option.
10. Start the new master server and new replica server (if the original master became a replica server).

Basic peer to peer replication

z/OS LDAP peer replication server provides failover support. With this support, if a LDAP server fails, the peer replication server can take over the role of the failing LDAP server and it is then available to process LDAP operations.

A z/OS LDAP peer replication server is a read/write replication server that can send and receive replicated entries. An LDAP server can have both peer replication servers and read-only replication servers defined as replicaObject entries.

Note: Basic peer to peer replication uses the same replica entry attribute values as shown in Replica server. The instructions in Adding replica entries in TDBM or LDBM also apply to peer replicas.

A basic peer to peer replication environment can be as simple as two LDAP servers that are peers to each other, or as complicated as several LDAP servers, where some servers are read-only replication servers and the other servers are peer replication servers. Every peer replication server must replicate to all other peer and read-only replication servers.
Server configuration

The peerServerDN and peerServerPW options in the backend section of the LDAP server configuration file are used to configure a basic peer to peer replication environment. See Chapter 8, “Customizing the LDAP server configuration” for more information.

Note: Usage of the peerServerPW configuration option is strongly discouraged in production environments. See Establishing the administrator DN and basic replication replica server DN and passwords for alternatives.

Basic replication conflict resolution

Minimal conflict resolution is done in a basic peer environment. For example, if peer replication server A receives an update to entry E at the same moment that peer B receives a delete of the same entry, basic replication can stall on server A. Ensure that your peer servers are not receiving conflicting operations. To avoid basic replication stalling, set up a replication error log to set aside replication errors. See Basic replication error log for more information.

When a conflict occurs, a notification will be sent to the console and server log.

Adding a peer replica to an existing server

For failover support, it might be necessary for you to add a peer replica for a backend to an existing server or set of servers. These servers can be stand-alone or already actively replicating.

In order to add a peer replica for a backend to a z/OS LDAP server, you should do the following:

1. Start the new peer replica in maintenance mode. The peer replica must have a peerServerDN and peerServerPW defined in the backend section of the LDAP server configuration file.

2. Stop the existing servers. For each existing server that is to become a peer server, update its configuration file to include the peerServerDN and peerServerPW configuration options. Restart the existing read-write servers in maintenance mode. See Establishing the administrator DN and basic replication replica server DN and passwords for alternatives to specifying the password in the configuration file.

3. Prime the new peer replica with all the data from an existing server. You can accomplish this by dumping the existing server’s directory (for TDBM or LDBM, use ds2ldif) and adding the data to the new peer replica (for TDBM or LDBM use ldapadd or, for TDBM, use ldif2ds). See Populating a replica for more information.

4. Add a replica entry to the existing servers to point to the new peer replica.

5. Add a replica entry in the new peer replica pointing to the existing server that was used to prime this server.

Note: If the existing server was a replicating server with replica entries defined to it, those replica entries might have been copied to the new peer replica in step 3 above. Ensure that this server does not contain a replica entry that defines this server as a replica of itself.

6. Turn off maintenance mode on all servers.

The existing servers and the new peer replica are now peer read-write replicas.

Upgrading a read-only replica to be a peer replica of the master server

It might be necessary for you to upgrade a read-only replica for a backend to a peer of its master, for example, if a peer of the master failed or further failover support is needed.

You should do the following to change a read-only replica for a backend to a peer replica:

1. Stop both the master server and the read-only replica.
2. Remove the `masterServer`, `masterServerDN`, and `masterServerPW` options from the backend section of the LDAP server configuration file of the read-only replica.

3. Add a `peerServerDN` and `peerServerPW` option to the backend section of each server's configuration file. The two servers will now be peer servers. See [Establishing the administrator DN and basic replication replica server DN and passwords](#) for alternatives to specifying the password in the configuration file.

4. Start both servers in maintenance mode.

5. In this backend, on the read-only replica being upgraded:
   - Add a replica entry for each replica that this backend on the master server points to (except the entry that previously pointed to the read-only replica that is being upgraded). This can include both peer servers and read-only replicas. Note that the master server might have other peer servers.
   - Add a replica entry to point to the master.

6. On the master, ensure that the credentials are valid in the replica entry for the read-only replica being upgraded.

7. Turn off maintenance mode on both servers.

The read-only replica and the master server are now peer read-write replicas for the backend.

---

**Downgrading a peer server to read-only replica**

It might be necessary for you to downgrade a backend from a peer server to a read-only replica, for example, if a previously upgraded read-only replica is no longer required to be a peer server, or to prevent out-of-sync conditions between peer servers.

You should do the following to downgrade a peer server to a read-only replica:

1. Stop the peer server.

2. Remove the `peerServerDN` and `peerServerPW` options from the backend section of the LDAP server configuration file.

3. Add `masterServer`, `masterServerDN`, and `masterServerPW` options to the backend section of the peer replica configuration file. If there are more than one peers, add a `masterServer` option for each one. See [Establishing the administrator DN and basic replication replica server DN and passwords](#) for alternatives to specifying the password in the configuration file.

4. Ensure that the credentials are valid in the replica entry for the newly downgraded peer server on all the replicating servers.

5. Start the server.

The peer server is now a read-only replica for the backend.

---

**SSL/TLS and basic replication**

SSL/TLS can be used to communicate between a replicating server (master or peer) and a replica server (read-only or peer).

**Replica server with SSL/TLS enablement**

Set the replica server up for SSL/TLS like a typical SSL/TLS server. It needs its own public-private key pair and certificate, and the LDAP server configuration file needs the standard SSL options (`listen`, `sslKeyRingFile`, and `sslKeyRingFilePW`). See [Setting up for SSL/TLS](#) for more information.

**Replicating server with SSL/TLS enablement**

The replicating server acts as an SSL/TLS client to the replica server.

To set up the replicating server, you must:
1. Run the gskkyman utility (see z/OS Cryptographic Services System SSL Programming), or the RACDCERT command (see z/OS Security Server RACF Command Language Reference), this time as if you were the client. The key database file, RACF keyring, or PKCS #11 token must contain the replicating server's key pair and certificate. Receive the replica's self-signed certificate and mark it as trusted.

2. In the LDAP server configuration file on the replicating server:
   - Set sslKeyRingFile to the replica key database file, RACF keyring, or PKCS #11 token created above.
   - If a replica key database file is used, set sslKeyRingFilePW to the password for the key database file, or set sslKeyRingPWStashFile to the file name where the password is stashed.

3. In the replica entry for this replica:
   - Set the replicaPort attribute to the replica's secure port number.
   - Set the replicaUseSSL attribute to TRUE.

See Setting up for SSL/TLS for more information.

Because the replicating server acts as an SSL/TLS client to the replica server, the replicating server binds with the replica server. The bind method used is simple bind. The SASL external bind method is not supported for basic replication.

### Basic replication error log

A replication error log holds information about each error that occurs during basic replication. To avoid stalling basic replication, the failed replication operation is taken off the replication queue so that replication can continue with the next operation. Depending on the error, the LDIF of the failed operation is set aside (added) to the error log.

There is one error log for each replica of a backend. The file name of the error log for a replica is specified by the ibm-slapdLog attribute in the replica entry for that replica within the backend. The file name must be unique across the LDAP server. If the attribute does not exist in the replica entry or the attribute has no value, no errors are logged or replication operations set aside during this backend's replication to that replica. In this case, basic replication to that replica stalls every time a failure occurs. The ibm-slapdReplMaxErrors attribute in the replica entry is set to control how many failed replication operations can be set aside each time the LDAP server is started before basic replication stalls for that replica.

In a sysplex environment, consider having the basic replication error log file shared by the LDAP servers which are sharing the replicating backend. Only the LDAP server that is the owner of the backend in the sysplex performs replication and writes to the replication error log if an error occurs. However, if that LDAP server stops, another LDAP server in the sysplex becomes owner and handles basic replication. If the error log file is shared, then the new owner can add error information to the same error log rather than beginning a new error log. This centralizes error information and eliminates the need to look for error logs on each server that might have become the backend owner.

The replication error log is used to correct basic replication in two ways:
- Use the error information to determine why replication failed.
- Use the ldapmodify utility to run the error log on the replica server, after resolving the basic replication problems. This performs the modifications that were set aside in the error log, therefore, bringing the backend in the replica to the same level as in the replicating server. You must bind as either the masterserverDN or peerserverDN, depending on the type of replica.

The following is an example of an error log entry:

```
# R004026 Entry cn=IBMUSER01, o=ibm,c=us not found in database. (tdbm_modify.c|1.125.1.1|584)
```
# setting change aside.
dn: cn=IBMUSER01, o=ibm,c=us
cn=IBMUSER01, o=ibm,c=us
changetype: modify
modify: uid
uid: IBMUSER1
modify: userpassword
userpassword:: Ym9i

The basic replication error log consists of three messages, each using one or more lines:
1. Message one indicates when the error occurred, the entry, and replica server.
2. Message two is the error message returned by the replica server.
3. Message three indicates what is being done. If the operation is set aside, this message is followed by the LDIF of the operation.

All non-LDIF information is prefixed with the comment character # so that the error log can be run through ldapmodify to synchronize the two servers.

Following is an example in which a replication error condition is logged but no set-aside of the modification is needed:

```
#(051219 14:13:12.366227): delete operation failed for cn=IBMUSER01,
o=ibm,c=us to dceimgtd.pdl.pok.ibm.com:390, rc=32
# A004026 Entry cn=IBMUSER01, o=ibm,c=us not found in database.
(tdbm_delete.c|1.58.1.1|280)
# Entry is already deleted, ignoring request.
```

There is no LDIF. Notice the third message indicates that request is being ignored.

**Troubleshooting basic replication**

If the replica server does not seem to be receiving updates from the replicating server (master or peer), there are several possible reasons. Check the following conditions for a possible quick fix:

- Check for messages from the replicating server.
- Verify that a replica entry for the replica server exists in the backend to be replicated in the replicating server, and was specified correctly to match with the replica server. If `cn=localhost` is used as the suffix for all replica entries for a backend, perform an `ldapsearch` with a base of `cn=localhost` and a filter of `objectClass=*`. Otherwise, perform an `ldapsearch` where the search base is the suffix defined in the backend section of the LDAP server configuration file and the filter is `objectClass=replicaObject`. If more than one suffix is configured for TDBM or LDBM, the search must be repeated using each suffix in the search base.
  
  See [IBM Tivoli Directory Server Client Programming for z/OS](https://www.ibm.com) for more information about `ldapsearch`.
- Verify that the `replicaHost` value in the replica entry for that replica specifies the machine on which the replica is running.
- Check that the values listed in the replica entry for that replica match those of the replica server configuration. Specifically, the `replicaPort`, `replicaBindDN`, and `replicaCredentials` should be verified.
- Check that the `replicaUpdateTimeInterval` specified in the replica entry for that replica has been set correctly.
- Verify that the replica server is running by performing an `ldapsearch` against the replica.
- Check that the default referral specified in the LDAP server configuration file in the replica server points to the replicating server.
- If the replica entry `replicaUseSSL` attribute is set to `TRUE`, verify the `replicaPort` attribute is set to the SSL port configured on the replica server. Verify the `sslKeyRingFile`, `sslKeyRingFilePW` or `sslKeyRingPWStashFile` values in the LDAP server configuration file on the replica server and on the replicating server are correct.
When adding a large number of entries, ensure that the region size for the replicating server is sufficient for replicating the entries to the replica. Entries on the replicating server are kept in memory during replication. If the region size is not sufficient, an out of memory condition can occur in the LDAP server. If possible, set the region size on the replicating server to 0M (or unlimited). If that cannot be done, set the region size to 14M (needed to run the LDAP server itself) plus twenty times the size of the largest LDIF file that is to be added to the replicating server.

The ibm-slapdLog and ibm-slapdReplMaxErrors attributes in a replica entry can be used to configure a replication error log for this replica. If basic replication fails, the error log holds all errors that occurred during replication and the LDIF for the set aside replication operations.

### Recovering from basic replication out-of-sync conditions

If a replica becomes out-of-sync with its replicating server for any reason, and normal replication processing is not correcting the situation, it might be necessary to reload the replica.

The following procedure should be followed to reload a replica:

1. Use the LDAP server MAINTMODE ON operator modify command on the replicating sever and on each of the replica servers to put them into maintenance mode.
2. Using the administrator DN, unload all the replica entries (entries that describe replica servers) from the master server. Use a search command similar to the one shown in [Searching a replica entry] to create LDIF output containing the replica entries for each suffix in the backend.
3. Using the administrator DN, run `ldapdelete` to remove the replica entries from the master. This resets the replication information in the replicating server.
4. For TDBM, run the following SPUFI on the replicating server to be sure that the server successfully completed the removal of the data in the DIR_REPLICA, DIR_REPENTRY and DIR_LONGREPENTRY tables:
   
   ```sql
   select count(*) from dbuserid.dir_replica
   ```

   where you substitute your database owner for `dbuserid`. The record count returned should be zero.
5. Stop all the replica servers.
6. Clear out the directory on each replica server.
   - For TDBM, drop and re-create the TDBM DB2 database. See [Creating the DB2 database and table spaces for TDBM or GDBM] for an example of the SPUFI commands needed to do this.
   - For LDBM, remove all the files in the LDBM database directory. See the description of the `databaseDirectory` option in Chapter 8, “Customizing the LDAP server configuration” for more information about the location of these files.
7. Run an unload utility on the replicating server. Use `ds2ldif` twice, once to unload the schema entry and a second time to unload the TDBM or LDBM directory entries.
8. Start the replica servers in maintenance mode.
9. Using an administrator DN, run `ldapmodify` to load the schema unloaded from the replicating server onto each replica.
10. On each replica, load the directory data retrieved above from the replicating server. For LDBM, you must use `ldapadd`. For TDBM, you can use `ldapadd` or use the `ldif2ds` load utility. `ldapadd` must be run using the administrator DN. If you use `ldif2ds`, you must stop the replica server before loading entries. In this case, restart the replica in maintenance mode after loading it.
11. Using an administrator DN, run `ldapadd` to add the replica entries unloaded in step 4 back into the replicating server.
12. Use the LDAP server MAINTMODE OFF operator modify command to take the replicating server and each replica out of maintenance mode.
Chapter 24. Advanced replication

Replication keeps data in multiple directory servers synchronized. Advanced replication includes the following function:

- Allows specific subtrees within the Directory Information Tree (DIT) to be chosen for participation in advanced replication topologies (in contrast to requiring the entire backend to participate or not participate)
- Allows the subtrees participating in an advanced replication environment to have different roles (for example, supplier or consumer)
- Additional replication topology choices can be combined to serve many different directory information architectures and data redundancy requirements
- External error log management using extended operations
- Operational attributes to determine the current state of the advanced replication environment
- External queue management using extended operations
- Schema replication

Advanced replication terminology

**Cascading replication**
A replication topology with multiple tiers of servers. A peer-master server replicates to a small set of read-only servers that replicate to other servers. Such a topology off-loads replication work from the master servers.

**Consumer server**
A server that receives changes from replication from another (supplier) server.

**Credentials entry**
An entry that identifies the method and required information that the supplier uses in binding to the consumer. For simple binds, this includes the distinguished name (DN) and password. This entry is specified in the replication agreement.

**Forwarding server**
A read-only server that replicates all changes sent to it. This contrasts to a peer-master server in that a peer-master server does not replicate changes sent to it from another peer-master server; it only replicates changes that are originally made on the peer-master server.

**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. This server also receives replication traffic from other gateway servers within the replication network, that 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 another subtree of the directory.

**Peer server**
The term used for a master server when there are multiple masters for a given subtree. A peer server does not replicate changes sent to it from another peer server; it only replicates changes that are originally made on it.

**Replica group**
The first entry created under a replication context has objectclass `ibm-replicaGroup` and represents a collection of servers participating in replication. It provides a convenient location to set ACLs to protect the replication topology information.
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 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
Identifies a portion of the Directory Information Tree (DIT) that is allowed to be replicated from one server to another. 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 filter
An entry containing the list of attributes that need to be replicated or excluded from replication corresponding to a particular type of entry. It can exist anywhere in the Directory Information Tree (DIT) but is always associated with an agreement.

Replication site
A gateway server and any master, peer, or replica servers configured to replicate together.

Replication topology
The set of entries in a directory that control what kind of information is replicated between LDAP servers and how it is replicated. These objects include:
- Replica groups
- Replica subentries
- Replication agreements
- Replication contexts
- Replication credentials entries
- Replication schedule entries

All LDAP servers in the replicating network should have the same replication topology.

Replication schedule
Replication can be scheduled to occur at particular times, with changes on the supplier accumulated and sent in a batch. The replication agreement contains the distinguished name (DN) for the entry that supplies the schedule.

Supplier server
A server that sends changes to another (consumer) server.
Replication topology

When advanced replication is configured, specific entries in the directory are identified as the roots of replication subtrees or replication contexts by adding the auxiliary objectclass `ibm-replicationContext` to them. Each of these replication contexts are replicated independently. The subtree continues down through the Directory Information Tree (DIT) until reaching the leaf entries or other replicated subtrees or contexts. Entries are added below the root of the replicated subtree to contain the replication configuration information. There are one or more replica group entries created directly under each replication context. For each replica group entry there is a corresponding replica subentry that identifies the role the server plays in the replication environment. Associated with each replica subentry are replication agreements that identify the servers that are supplied (replicated to) by each server and defining the credentials and schedule information.

By using advanced replication, a change made to one server is propagated to one or more additional servers. In effect, a change to one server can show up on multiple different LDAP servers. z/OS IBM TDS supports either basic or advanced replication but not both at the same time. Advanced replication includes:

- 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 advanced replication model changes the concept of master and replica servers. 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.

The types of directory roles as defined by function are: master-replica, peer-peer, forwarding (cascading), gateway

Table 58. Server roles

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Master-replica</td>
<td>A replica is an additional server that contains a copy of the directory information that is replicated from the master server. The replicated data can be the entire DIT or just a portion of the DIT that is replicated to the replica. The replica server provides a read-only backup of the replicated subtree.</td>
</tr>
<tr>
<td>Master-peer</td>
<td>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.</td>
</tr>
</tbody>
</table>

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.
Table 58. Server roles (continued)

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forwarding (Cascading)</td>
<td>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 that contains many widely dispersed replicas.</td>
</tr>
<tr>
<td>Gateway</td>
<td>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.</td>
</tr>
</tbody>
</table>

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 replication fails, it is repeated even if the master is restarted. Changes are replicated in the order that they are made on the master. See Recovering from advanced replication errors for more information.

If you are no longer using a replica, you must remove the replication agreement entry from the supplier. Leaving the entry causes the server to queue up all updates and uses unnecessary directory space. Also, the supplier continues trying to contact the missing consumer to try sending the data again. When a replication agreement is deleted, replication is halted immediately. That is, any updates in the replication queue are lost.

Advanced replication overview

This section presents a high-level description of the various advanced replication topologies.

Master-replica replication

The basic relationship in advanced 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, and 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, that 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.
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.

### Forwarding (cascading) replication

Forwarding (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:

![Cascading replication diagram](image)

**Figure 45. 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 timestamps of entries. See [Replication conflict resolution on page 383](#) for more information.

The following is an example of 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 46. Peer-to-peer replication
Advanced replication features

This topic presents a high-level overview of advanced replication features.

Partial replication

Partial replication is an advanced replication feature that replicates only the specified entries and a subset of attributes for the specified entries within a subtree. Using partial replication, an LDAP 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 about partial replication, see "Partial replication" on page 423.

Replication scheduling

Replication scheduling is an advanced replication feature that allows updates to be queued and then replicated at a certain time each day or during certain days of the week. Using replication scheduling, an LDAP administrator can schedule advanced replication to occur at optimal times when network traffic is minimal. For more information about replication schedule entries, see Schedule entries.

Replication conflict resolution

If there are replication conflicts involving delete or modifyDN operations, LDAP administrator intervention might be needed to correct the problems. For example, if an entry is renamed on one server while it is...
being modified on a second server, the 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 distinguished name (DN). All information necessary to redo the modify
with the correct name is preserved in the replication and error logs. Replication errors are rare
occurrences in a correctly configured replication topology, but it is not safe to assume that they never
occur.

Conflict resolution for add and modify operations in peer-to-peer replication is based on the
modifyTimeStamp attribute value. The entry update with the most recent modifyTimeStamp on any
server in a multi-master replication environment is the one that takes precedence. Replicated delete and
rename (modify DN) requests 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 that is specified in the ibm-slapdlog attribute of the cn=Replication,cn=Log
Management,cn=configuration entry.

Updates to the same entry made by multiple servers might cause inconsistencies in directory data
because conflict resolution is based on the modifyTimeStamp value of the entries. The most recent
modifyTimeStamp value takes precedence. If the data on your servers becomes inconsistent, use the
synchronization procedure in Recovering from advanced replication errors to re-synchronize the servers.

For advanced replication conflict resolution to work correctly, the supplier server must provide the modified
entry’s modifyTimeStamp value before the entry was updated on the supplier. The consumer server uses
the modifyTimeStamp attribute value to determine what to do with a modified entry. If the consumer
server receives a modifyTimeStamp value on an entry that is earlier than the same entry’s
modifyTimeStamp in its own server, then the modify request from the supplier server is ignored. However,
this same replication conflict resolution does not occur for the schema entry, cn=schema. The replicated
cn=schema entry is always replaced on the consumer server even if the consumer server has a later
modifyTimeStamp value.

Enabling advanced replication

Before advanced replication entries are allowed to be added to the TDBM or LDBM backends, the CDBM
backend must be configured in the LDAP server configuration file and the useAdvancedReplication
configuration option set to on in the CDBM backend. For example:

database CDBM GLDBCD31/GLDBCD64
databaseDirectory /var/ldap/cdbm
useAdvancedReplication on

Notes:

1. The CDBM backend is only allowed to be configured when the server compatibility is 5 or greater. See
page 100 for more information about the serverCompatLevel configuration option.
2. If useAdvancedReplication on is specified in the CDBM backend and basic replication entries with an
objectclass of replicaObject exist in any configured TDBM or LDBM backends, the server will not
start. Entries with an objectclass of replicaObject are not allowed to be added when advanced
replication is allowed. Basic and advanced replication environments are not supported at the same
time in the z/OS LDAP server. If planning to use an advanced replication environment, all basic
replication replicaObject entries must be removed from the TDBM or LDBM backends.
3. If there are advanced replication entries in the LDBM and TDBM backends and
useAdvancedReplication off is specified in the CDBM backend, the server will not start because
basic replication is intended to be used. Replication contexts, replica groups, replica subentries, and
replication agreement entries are not allowed to be added when basic replication is allowed.
4. The masterServer, masterServerDN, masterServerPW, peerServerDN, and peerServerPW
configuration options are not allowed to be specified in any LDBM or TDBM backends when the CDBM
backend is configured and the useAdvancedReplication option is set to on. The masterServer,
masterServerDN, masterServerPW, peerServerDN, and peerServerPW options are only valid when
the server is configured to run in a basic replication environment.
The \texttt{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 \texttt{cn=configuration} suffix are automatically created:

- \texttt{cn=configuration}
- \texttt{cn=Replication,cn=configuration}
- \texttt{cn=Log Management,cn=Configuration}
- \texttt{cn=Replication,cn=Log Management,cn=Configuration}

See \textit{CDBM backend configuration entries} for more information about the above entries and attribute values that affect advanced replication configuration.

### Supplier server entries

The following sections indicate the entries that must be added or modified in the supplier server to successfully configure advanced replication.

#### Replication contexts

A replication context is an entry in the directory with the auxiliary objectclass \texttt{ibm-replicationContext} that identifies the root of a replicated subtree. The auxiliary objectclass \texttt{ibm-replicationContext} is allowed to be added to any entry in the directory. See \textit{Table 59} for the optional attribute value for the \texttt{ibm-replicationContext} objectclass. The replication configuration information is maintained in a set of entries created below the base of a replication context. If there is more than one replication context present in the same subtree, the replication configuration information under the child replication context entry is used while the replication configuration under the parent entry is ignored.

If the \texttt{ibm-replicationContext} auxiliary objectclass is added to a non-suffix level entry in the directory, explicit \texttt{aclEntry} and \texttt{entryOwner} attribute values are required. When an \texttt{entryOwner} attribute value is required, it must start with an \texttt{access-id}: See \textit{Protecting replication topology entries} for more information about protecting the replication topology.

\textit{Table 59. ibm-replicationContext objectclass schema definition (optional attribute)}

<table>
<thead>
<tr>
<th>Attribute description and example</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{ibm-replicaReferralURL}</td>
</tr>
<tr>
<td>A single valued attribute that contains an ordered list of LDAP URLs with server name and optional port numbers separated by spaces. This list contains a list of servers that have update access to this replication context.</td>
</tr>
<tr>
<td>Example:</td>
</tr>
</tbody>
</table>

For the example in \textit{Table 59} assume that a replication context of \texttt{o=ibm} is used. When a client attempts an update operation on the consumer server under the \texttt{o=ibm} replication context, the referral list in the \texttt{ibm-replicaReferralURL} attribute value is sent back to the client indicating the supplier servers for the replication context.

#### Replica groups

A replica group entry is created directly under a replication context entry with the structural objectclass \texttt{ibm-replicaGroup}. See \textit{Table 60} for the optional and required attributes for the \texttt{ibm-replicaGroup} objectclass. A replica group entry represents a collection of servers participating in replication for the context. Multiple replica group entries are allowed to be created under a replication context. A replica group entry provides a convenient location to set ACLs to protect the replication topology information, however, these entries do not do anything as far as how the replication topology is configured. See \textit{Protecting replication topology entries} for more information about protecting the replication topology.
Table 60. `ibm-replicaGroup` objectclass schema definition (optional and required attributes)

<table>
<thead>
<tr>
<th>Attribute description and example</th>
</tr>
</thead>
<tbody>
<tr>
<td>description</td>
</tr>
<tr>
<td>An optional attribute that provides a text field for extra information pertaining to the replica group entry. This attribute does not affect advanced replication configuration.</td>
</tr>
<tr>
<td>Example:</td>
</tr>
<tr>
<td>description: Replica group 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><code>ibm-replicaGroup</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>A required attribute value that specifies the name of a replica group.</td>
</tr>
<tr>
<td>Example:</td>
</tr>
<tr>
<td><code>ibm-replicaGroup: Group1</code></td>
</tr>
</tbody>
</table>

**Replica subentries**

A replica subentry is created directly under a replica group with the structural objectclass `ibm-replicaSubentry`. See Table 61 for the optional and required attributes for the `ibm-replicaSubentry` objectclass. A replica subentry identifies the role the server plays in advanced replication (for example master or read-only replica). If the auxiliary objectclass `ibm-replicaGateway` is added to a replica subentry, the server’s role is a gateway server. See Gateway replication for more information. There should only be one replica subentry created for a single server under a given replication context.

Table 61. `ibm-replicaSubentry` objectclass schema definition (optional and required attributes)

<table>
<thead>
<tr>
<th>Attribute description and example</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>cn</code></td>
</tr>
<tr>
<td>A required attribute that specifies the common name of the replica subentry. This attribute does not affect advanced replication configuration.</td>
</tr>
<tr>
<td>Example:</td>
</tr>
<tr>
<td><code>cn: Subentry 1</code></td>
</tr>
</tbody>
</table>

| `description`                     |
| An optional attribute that provides an additional text field for extra information pertaining to the replica subentry. This attribute does not affect advanced replication configuration. |
| Example:                          |
| `description: Represents the LDAP server (master1) under this replication context` |

| `ibm-replicaServerID`             |
| A required attribute that specifies the server ID of the server that this entry represents. A server’s ID can be determined by searching the root DSE entry for the `ibm-serverID` attribute. This attribute cannot be modified after this entry is created. If this attribute value must be changed, all entries under the replica subentry must be deleted and then re-added. A server does not interrogate the replica subentry or any replication agreements beneath it when the `ibm-replicaServerID` attribute value does not match its own ID. See Replication agreements for more information about replication agreement entries. |
| Example:                          |
| `ibm-replicaServerID: supplier1` |
Table 61. *ibm-replicaSubentry* objectclass schema definition (optional and required attributes) (continued)

<table>
<thead>
<tr>
<th>Attribute description and example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ibm-replicationServerIsMaster</strong></td>
</tr>
<tr>
<td>A required boolean (true or false) attribute that indicates whether the server represented by this replica subentry, as determined by the <em>ibm-replicaServerID</em> attribute value, is a master server for the replication context.</td>
</tr>
<tr>
<td>If set to true, the server represented by this replica subentry, as determined by the <em>ibm-replicaServerID</em> attribute, is a master server for the replication context. If set to false, the server represented by this replica subentry, as determined by the <em>ibm-replicaServerID</em> attribute, is a forwarding server if there are any replication agreement entries, or else it is a just a read-only replica for the replication context.</td>
</tr>
<tr>
<td>Example:</td>
</tr>
<tr>
<td><em>ibm-replicationServerIsMaster</em>: true</td>
</tr>
</tbody>
</table>

For the examples in Table 61, the replica subentry represents a supplier server with a server ID of supplier1. It is also the master server under the replication context (*o=ibm*) where this replica subentry resides.

**Replication agreements**

A replication agreement is an entry in the directory with the structural object class *ibm-replicationAgreement* created directly under a replica subentry to define replication from the server represented by the subentry to another server. See Table 62 for the required attributes for the *ibm-replicationAgreement* objectclass. See Table 63 for the optional attributes for the *ibm-replicationAgreement* objectclass. A replication agreement entry is similar to a *replicaObject* entry used in basic replication. This object represents an individual connection from a supplier server to a consumer server. A replica subentry may have any number of replication agreement entries defined under it to specify each supplier agreement this server has under this replication topology.

Table 62. *ibm-replicationAgreement* objectclass schema definition (required attributes)

<table>
<thead>
<tr>
<th>Attribute description and example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>cn</strong></td>
</tr>
<tr>
<td>Common name of the replication agreement entry. This attribute does not affect advanced replication configuration.</td>
</tr>
<tr>
<td>Example:</td>
</tr>
<tr>
<td><em>cn</em>: agreement1</td>
</tr>
<tr>
<td><strong>ibm-replicaConsumerID</strong></td>
</tr>
<tr>
<td>Identifies the server ID of the consumer server. This value should match the <em>ibm-serverID</em> attribute value in the root DSE entry of the consumer server or a warning message is issued in the LDAP server log when the replication agreement initializes.</td>
</tr>
<tr>
<td>Example:</td>
</tr>
<tr>
<td><em>ibm-replicaConsumerID</em>: consumer1</td>
</tr>
<tr>
<td><strong>ibm-replicaCredentialsDN</strong></td>
</tr>
<tr>
<td>Specifies the distinguished name (DN) of the entry containing the credentials entry used to authenticate to the consumer server. See <a href="#">Credentials entries</a> for more information about credential entries.</td>
</tr>
<tr>
<td>Example:</td>
</tr>
<tr>
<td><em>ibm-replicaCredentialsDN</em>: cn=consumer1, cn=localhost</td>
</tr>
<tr>
<td>Attribute description and example</td>
</tr>
<tr>
<td>----------------------------------</td>
</tr>
<tr>
<td><strong>ibm-replicaURL</strong></td>
</tr>
<tr>
<td>Specifies the LDAP URL of the consumer server. The LDAP URL syntax is fully documented in <a href="#">RFC 2255: The LDAP URL Format</a>. The prefix of the LDAP URL indicates whether a non-secure or secure connection is used between the supplier and consumer servers. If the LDAP URL prefix is <code>ldap://</code>, a non-secure connection is used. If the LDAP URL prefix is <code>ldaps://</code>, a secure connection is used. See <a href="#">SSL/TLS and advanced replication</a> for more information about using SSL/TLS in an advanced replication environment.</td>
</tr>
<tr>
<td>Example:</td>
</tr>
<tr>
<td><code>ibm-replicaURL: ldaps://consumer1.ibm.com:500</code></td>
</tr>
</tbody>
</table>

For the examples in Table 62 when the replicating server receives and successfully finishes an update request, the update is also sent to the consumer server with an ID of `consumer1` that is located on hostname `consumer1.ibm.com` on secure port 500. The replicating server performs a simple or SASL EXTERNAL bind operation using the information provided in the credentials entry `cn=consumer1,cn=localhost`.

<table>
<thead>
<tr>
<th>Attribute description and example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>description</strong></td>
</tr>
<tr>
<td>Provides an additional text field for extra information pertaining to the replication agreement entry. This attribute does not affect advanced replication configuration.</td>
</tr>
<tr>
<td>Example:</td>
</tr>
<tr>
<td><code>description: Represents the replication agreement from the supplier1 server to the consumer1 server</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute description and example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ibm-replicaScheduleDN</strong></td>
</tr>
<tr>
<td>Specifies the DN of a schedule entry that determines when replication updates are sent to this consumer. If a schedule DN is not specified, advanced replication defaults to &quot;immediate&quot; replication mode. See <a href="#">Schedule entries</a> for more information about advanced replication scheduling.</td>
</tr>
<tr>
<td>Example:</td>
</tr>
<tr>
<td><code>ibm-replicaScheduleDN: cn=schedule,o=ibm</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute description and example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ibm-replicationCreateMissingEntries</strong></td>
</tr>
<tr>
<td>A boolean (true or false) indicating whether missing parent entries are to be created on the consumer server. If set to true, the missing parent entries are automatically created by the supplier server and replicated to the consumer server. If set to false or the attribute is not specified, the missing parent entries are not created on the consumer server.</td>
</tr>
<tr>
<td>Example:</td>
</tr>
<tr>
<td><code>ibm-replicationCreateMissingEntries: true</code></td>
</tr>
</tbody>
</table>
Table 63. *ibm-replicationAgreement* objectclass schema definition (optional attributes) (continued)

<table>
<thead>
<tr>
<th>Attribute description and example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ibm-replicationExcludedCapability</strong></td>
</tr>
<tr>
<td>A multi-valued attribute that lists the OIDs of features that the consumer server does not support. Operations related to these capabilities are excluded from the updates sent to the consumer in this replication agreement. If this attribute is not specified, no capabilities are excluded from being replicated.</td>
</tr>
<tr>
<td>Only the following capabilities are allowed to be excluded:</td>
</tr>
<tr>
<td>1. 1.3.18.0.2.32.4 – IBM filtered ACLs (not currently supported on z/OS)</td>
</tr>
<tr>
<td>2. 1.3.18.0.2.32.5 – IBM password policy (not currently supported on z/OS)</td>
</tr>
<tr>
<td>Example:</td>
</tr>
<tr>
<td>ibm-replicationExcludedCapability: 1.3.18.0.2.32.5</td>
</tr>
<tr>
<td><strong>ibm-replicationFilterDN</strong></td>
</tr>
<tr>
<td>Specifies the DN of a replication filter entry that contains filters that include or exclude the replication of certain entries or attribute types to the consumer server. See Partial replication for additional information about using partial replication.</td>
</tr>
<tr>
<td>Example:</td>
</tr>
<tr>
<td>ibm-replicationFilterDN: cn=filter,o=ibm</td>
</tr>
<tr>
<td><strong>ibm-replicationOnHold</strong></td>
</tr>
<tr>
<td>A boolean (true or false) indicating whether advanced replication from the replication agreement is suspended or not. If set to true, replication updates from the supplier server to the consumer server are queued until this attribute value is set to false, if set to false or this attribute is not specified, replication updates are handled normally.</td>
</tr>
<tr>
<td>This attribute value is also modified by the Cascading control replication and the Control replication extended operations. See Cascading control replication for more information about the Cascading control replication extended operation. See Control replication for more information about the Control replication extended operation.</td>
</tr>
<tr>
<td>Example:</td>
</tr>
<tr>
<td>ibm-replicationOnHold: false</td>
</tr>
</tbody>
</table>

To aid in enforcing the accuracy of the data within the replication agreement entry, when the supplier binds to the consumer, it retrieves the *ibm-serverID* attribute from the root DSE entry and compares it to the *ibm-replicaConsumerID* attribute value. A warning is logged in the LDAP server’s job log if these server IDs do not match.

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.

### Credentials entries

Because the replication agreement entry can be replicated, a DN to credentials object is used in the *ibm-replicaCredentialsDN* attribute value. This allows the supplier server credentials entry to be stored in an area of the DIT that is not replicated. Replicating the supplier server credentials entries (where 'clear text' passwords must be obtainable) represents a potential security exposure. The *cn=localhost* suffix in an LDBM or TDBM backend is an appropriate location for the creation of supplier server credential entries.

The objectclass of the entry specified in the *ibm-replicaCredentialsDN* attribute value in the replication agreement indicates the authentication method used by the supplier to authenticate with the consumer server. If the entry’s objectclass is *ibm-replicationCredentialsSimple*, the supplier server uses a simple bind to authenticate to the consumer. See Table 64 for the required attributes of the *ibm-replicationCredentialsSimple* objectclass. If the entry’s objectclass is *ibm-
replicationCredentialsExternal, the supplier server performs a SASL EXTERNAL bind to the consumer server. See Table 65 for the optional attributes of the ibm-replicationCredentialsExternal objectclass.

A consumer server credentials entry is required on the consumer server to identify the distinguished name that the supplier server is using to perform a simple or SASL EXTERNAL bind. See Consumer server entries for more information about the consumer server credential entries.

Table 64. ibm-replicationCredentialsSimple objectclass schema definition (required attributes)

<table>
<thead>
<tr>
<th>Attribute description and example</th>
</tr>
</thead>
<tbody>
<tr>
<td>replicaBindDN</td>
</tr>
<tr>
<td>Specifies the LDAP distinguished name that the replicating server uses to bind with the consumer server when sending directory updates.</td>
</tr>
<tr>
<td>Example:</td>
</tr>
<tr>
<td>replicaBindDN: cn=supplier,cn=localhost</td>
</tr>
</tbody>
</table>

| replicaCredentials               |
| Contains the authentication information needed for the replicating server to authenticate with the consumer server using the distinguished name specified in the replicaBindDN attribute value. |
| Example:                          |
| replicaCredentials: secret        |

For the examples in Table 64, the replication agreement uses a simple bind to the consumer server using a bind DN of cn=supplier,cn=localhost and a bind password of secret.

Table 65. ibm-replicationCredentialsExternal objectclass schema definition (optional attributes)

<table>
<thead>
<tr>
<th>Attribute description and example</th>
</tr>
</thead>
<tbody>
<tr>
<td>ibm-replicaKeyFile</td>
</tr>
<tr>
<td>Specifies the path and file name of the SSL/TLS key database file, the name of the RACF key ring, or the name of the PKCS #11 token to be used by the replication agreement to perform an SASL EXTERNAL bind. Specifying a value here, overrides the default that comes from the sslKeyRingFile configuration option. See page 104 for the acceptable formats for this attribute value.</td>
</tr>
<tr>
<td>Example:</td>
</tr>
<tr>
<td>ibm-replicaKeyFile: /home/server1/server1.kdb</td>
</tr>
</tbody>
</table>

Note: If a value is specified for this attribute, it must be the same for all replication agreements in the server. The LDAP server only supports having one opened SSL/TLS key database file, RACF key ring, or PKCS #11 token at a time. It is strongly recommended that all SSL certificates that need to be used by the server be placed in one SSL/TLS key database file, RACF key ring, or PKCS #11 token.
Table 65. ibm-replicationCredentialsExternal objectclass schema definition (optional attributes) (continued)

<table>
<thead>
<tr>
<th>Attribute description and example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ibm-replicaKeyLabel</strong></td>
</tr>
<tr>
<td>Specifies the label of the certificate that is used for LDAP server-client authentication for the SASL EXTERNAL bind. The certificate label must reside in the SSL/TLS key database file, RACF key ring, or PKCS #11 token being used for this credentials entry, as specified by the ibm-replicaKeyFile attribute value or the sslKeyRingFile configuration option if ibm-replicaKeyFile is not specified. Specifying a value here, overrides the default that comes from the sslCertificate configuration option. If the sslCertificate configuration option is not specified or is set to none, the default SSL certificate in the ibm-replicaKeyFile attribute value (or the sslKeyRingFile configuration option if ibm-replicaKeyFile is not specified) is used. See page 103 for more about the sslCertificate option. Example: ibm-replicaKeyLabel: EXTERNAL1</td>
</tr>
<tr>
<td><strong>ibm-replicaKeyPwd</strong></td>
</tr>
<tr>
<td>Specifies the password protecting access to the SSL/TLS key database file. It can also be used to specify a fully qualified file name where the password for the SSL/TLS key database file is stashed. This attribute should only be specified if the ibm-replicaKeyFile attribute value (or the sslKeyRingFile configuration option if ibm-replicaKeyFile is not specified) is an SSL/TLS key database file. If using an SSL stash file, it must be specified in the following format: file://filename where filename is the fully qualified z/OS Unix System Services file system location of the SSL stash file. This password value is encrypted if is added or modified when the secretEncryption configuration option is set to AES or DES under the backend containing the replication agreement. If secretEncryption is set to AES or DES, this improves directory security because the password is no longer stored in the directory in clear text. Example: ibm-replicaKeyPwd: secret</td>
</tr>
</tbody>
</table>

For the examples in Table 65, the replication agreement uses a SASL EXTERNAL bind to the consumer server with the SSL certificate label EXTERNAL1 in SSL key database file /home/server1/server1.kdb that has a password of secret.

A SASL EXTERNAL bind requires a secure connection from the replicating server to the replica server. The replication agreement entry must use an ibm-replicaURL attribute value with an LDAP URL prefix of ldaps:// to signify an SSL connection. The replicating server must have read access to the SSL/TLS key database file, RACF key ring, or PKCS #11 token that is specified in the sslKeyRingFile configuration option or the ibm-replicaKeyFile attribute value in the SASL EXTERNAL supplier server credentials entry. If the optional attribute values in Table 65 are not specified in the SASL EXTERNAL supplier server credentials entry, the default SSL configuration in the LDAP server configuration file is used.

See SSL/TLS and advanced replication for more information about using SSL/TLS in an advanced replication environment.

Schedule entries

An LDAP administrator can schedule advanced replication to occur at optimal times when network traffic is minimal for each individual replication agreement. Each replication agreement entry is allowed to have an ibm-replicaScheduleDN attribute value optionally specified. This attribute value identifies the distinguished name (DN) of a weekly schedule entry, that has an object class of ibm-replicationWeeklySchedule. See Table 66 for the schema definition of the ibm-
replicationWeeklySchedule objectclass. The weekly schedule entry allows an LDAP administrator to specify the distinguished name (DN) of additional entries that point to one or more daily replication schedule entries. If the distinguished name in the ibm-replicaScheduleDN attribute value cannot be found or is not a weekly schedule entry, advanced replication continues by ignoring the weekly replication schedule.

Table 66. ibm-replicationWeeklySchedule objectclass schema definition (optional attributes)

<table>
<thead>
<tr>
<th>Attribute description and example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>cn</strong></td>
</tr>
<tr>
<td>Common name of the weekly replication schedule entry. This attribute does not affect advanced replication configuration.</td>
</tr>
<tr>
<td>Example:</td>
</tr>
<tr>
<td>cn: myweekly</td>
</tr>
<tr>
<td><strong>description</strong></td>
</tr>
<tr>
<td>Provides an additional text field for extra information pertaining to the weekly schedule entry. This attribute does not affect advanced replication configuration.</td>
</tr>
<tr>
<td>Example:</td>
</tr>
<tr>
<td>description: Weekly schedule for advanced replication</td>
</tr>
<tr>
<td><strong>ibm-replWeeklySchedName</strong></td>
</tr>
<tr>
<td>Descriptive name for the weekly schedule entry. This attribute does not affect advanced replication configuration.</td>
</tr>
<tr>
<td>Example:</td>
</tr>
<tr>
<td>ibm-replWeeklySchedName: Weekly schedule for agreement 1</td>
</tr>
<tr>
<td><strong>ibm-scheduleMonday</strong></td>
</tr>
<tr>
<td>Specifies the distinguished name (DN) of a daily replication schedule entry for Monday.</td>
</tr>
<tr>
<td>Example:</td>
</tr>
<tr>
<td>ibm-scheduleMonday: cn=monday,o=ibm</td>
</tr>
<tr>
<td><strong>ibm-scheduleTuesday</strong></td>
</tr>
<tr>
<td>Specifies the distinguished name (DN) of a daily replication schedule entry for Tuesday.</td>
</tr>
<tr>
<td>Example:</td>
</tr>
<tr>
<td>ibm-scheduleTuesday: cn=tuesday,o=ibm</td>
</tr>
<tr>
<td><strong>ibm-scheduleWednesday</strong></td>
</tr>
<tr>
<td>Specifies the distinguished name (DN) of a daily replication schedule entry for Wednesday.</td>
</tr>
<tr>
<td>Example:</td>
</tr>
<tr>
<td>ibm-scheduleWednesday: cn=thursday,o=ibm</td>
</tr>
<tr>
<td><strong>ibm-scheduleThursday</strong></td>
</tr>
<tr>
<td>Specifies the distinguished name (DN) of a daily replication schedule entry for Thursday.</td>
</tr>
<tr>
<td>Example:</td>
</tr>
<tr>
<td>ibm-scheduleThursday: cn=thursday,o=ibm</td>
</tr>
</tbody>
</table>
Table 66. *ibm-replicationWeeklySchedule* objectclass schema definition (optional attributes) (continued)

<table>
<thead>
<tr>
<th>Attribute description and example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ibm-scheduleFriday</strong></td>
</tr>
<tr>
<td>Specifies the distinguished name (DN) of a daily replication schedule entry for Friday.</td>
</tr>
<tr>
<td>Example:</td>
</tr>
<tr>
<td>ibm-scheduleFriday: cn=friday,o=ibm</td>
</tr>
<tr>
<td><strong>ibm-scheduleSaturday</strong></td>
</tr>
<tr>
<td>Specifies the distinguished name (DN) of a daily replication schedule entry for Saturday.</td>
</tr>
<tr>
<td>Example:</td>
</tr>
<tr>
<td>ibm-scheduleSaturday: cn=saturday,o=ibm</td>
</tr>
<tr>
<td><strong>ibm-scheduleSunday</strong></td>
</tr>
<tr>
<td>Specifies the distinguished name (DN) of a daily replication schedule entry for Sunday.</td>
</tr>
<tr>
<td>Example:</td>
</tr>
<tr>
<td>ibm-scheduleSunday: cn=sunday,o=ibm</td>
</tr>
</tbody>
</table>

A daily replication schedule entry has an object class of *ibm-replicationDailySchedule*. See Table 67 for the schema definition of the *ibm-replicationDailySchedule* objectclass. A daily replication schedule entry allows an LDAP administrator to accomplish the following replication scheduling:

- Configure the time each day to start advanced replication for that replication agreement. This is accomplished by using the multi-valued *ibm-replicationImmediateStart* attribute.
- Allows replication to be turned off by specifying a batch time. This drains the replication queue and replication waits until the next scheduled time once the queue is fully drained. This is done by using the multi-valued *ibm-replicationBatchStart* attribute.
- Advanced replication can be turned on and off multiple times each day.

Table 67. *ibm-replicationDailySchedule* objectclass schema definition (optional attributes)

<table>
<thead>
<tr>
<th>Attribute description and example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>cn</strong></td>
</tr>
<tr>
<td>Common name of the daily schedule entry. This attribute does not affect advanced replication configuration.</td>
</tr>
<tr>
<td>Example:</td>
</tr>
<tr>
<td>cn: mydaily</td>
</tr>
<tr>
<td><strong>description</strong></td>
</tr>
<tr>
<td>Provides an additional text field for extra information pertaining to the daily schedule entry. This attribute does not affect advanced replication configuration.</td>
</tr>
<tr>
<td>Example:</td>
</tr>
<tr>
<td>description: Each day stops replication at 6:30 AM and then restarts at 9:45 AM and continues for the rest of the day until 6:30 AM the following day.</td>
</tr>
<tr>
<td><strong>ibm-replDailySchedName</strong></td>
</tr>
<tr>
<td>Descriptive name for the daily schedule entry. This attribute does not affect advanced replication configuration.</td>
</tr>
<tr>
<td>Example:</td>
</tr>
<tr>
<td>ibm-replDailySchedName: Daily schedule for replication agreement 1</td>
</tr>
</tbody>
</table>
### Attribute description and example

<table>
<thead>
<tr>
<th>Attribute description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ibm-replicationBatchStart</strong></td>
<td>A multi-valued attribute that indicates the time batch replication starts. All updates in the replication queue are replicated at the time specified and then replication waits until the next scheduled time.</td>
</tr>
<tr>
<td><strong>Note:</strong> If advanced replication is waiting, it is not allowed to be resumed with either a Cascading control replication extended operation or a Control replication extended operation. However, &quot;replicate now&quot; on a Control replication extended operation can be used to immediately drain the replication queue if replication is waiting. See Cascading control replication for more information about the Cascading control replication extended operation. See Control replication for more information about the Control replication extended operation.</td>
<td></td>
</tr>
<tr>
<td>The attribute value format is: Thh:mm:ss</td>
<td></td>
</tr>
<tr>
<td>where:</td>
<td></td>
</tr>
<tr>
<td>hh - Hour based on a 24 hour clock (00 – 23)</td>
<td></td>
</tr>
<tr>
<td>mm - Minutes (00-59)</td>
<td></td>
</tr>
<tr>
<td>ss - Seconds (00-59)</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>ibm-replicationBatchStart: T06:30:00</td>
<td></td>
</tr>
<tr>
<td>This value indicates that queued replication updates are replicated at 6:30 AM until the replication queue is drained; then replication waits.</td>
<td></td>
</tr>
</tbody>
</table>

| **ibm-replicationImmediateStart** | A multi-valued attribute that indicates when advanced replication immediately starts and continues until the next ibm-replicationBatchStart attribute value or replication is otherwise suspended. |
| Advanced replication is allowed to be suspended or resumed with a Cascading control replication extended operation or a Control replication extended operation. See Cascading control replication for more information about the Cascading control replication extended operation. See Control replication for more information about the Control replication extended operation. |
| The attribute value format is: Thh:mm:ss |
| where: |
| hh - Hour based on a 24 hour clock (00 – 23) |
| mm - Minutes (00-59) |
| ss - Seconds (00-59) |
| Example: |
| ibm-replicationImmediateStart: T09:45:00 |
| This value indicates that replication starts at 9:45 AM. |

| **ibm-replicationTimesUTC** | A boolean (true or false) indicating whether the time values specified in the ibm-replicationBatchStart and ibm-replicationImmediateStart attributes are in GMT or local time. If set to true, GMT is used for time values. If set to false or the attribute is not specified, local time is used for time values. |
| Example: |
| ibm-replicationTimesUTC: true |

Assuming that each daily schedule in the weekly schedule entry uses the examples in Table 67, advanced replication occurs daily as follows:

- Because an ibm-replicationBatchStart attribute value of T06:30:00 is specified, the replication queue is drained and replication waits at 6:30 AM each day. All future replication updates are queued.
• At 9:30 AM each day, advanced replication restarts because an `ibm-replicationImmediateStart` attribute value has been specified. Replication immediately starts and continues until the next day at 6:30 AM.

When the LDAP server starts and there is a weekly replication schedule entry configured, advanced replication inherits the state of the most recent `ibm-replicationBatchStart` or `ibm-replicationImmediate` time. If the weekly schedule examples are used in Table 67 and the LDAP server starts at 7:00 AM, replication is suspended until 9:30 AM when the next `ibm-replicationImmediate` time is encountered. This processing occurs even if there is a missing daily schedule in the weekly schedule entry.

### Consumer server entries

The only replication related entry needed on the consumer server is the consumer server credentials entry. The consumer server credentials entry must reside under the `cn=configuration` suffix in the CDBM backend.

**Note:** The consumer server credentials entry differs from the supplier server credentials entry. See Credentials entries for more information about the supplier server credentials entry.

The consumer server credentials entry is used on the consumer server to verify that it is actually a supplier server performing a simple or SASL EXTERNAL bind. A consumer server only accepts update operations from its supplier server and the LDAP administrator when using the Server Administration control. There are two types of consumer server credential entries that can be used, one that has an objectclass of `ibm-slapdReplication` and the other has an objectclass of `ibm-slapdSupplier`.

When a supplier server replicates updates to its consumer server, a special entry is used to indicate that the supplier server has master level access to the consumer server. Master level access bypasses ACL and entry owner restrictions and allows updates to be made even when the server is a read-only consumer, cascading consumer, or under a quiesced replication context. If the supplier server authenticates to the consumer server with a simple bind, the DN specified by the `replicaBindDN` attribute value in the replication agreement entry is used as the bind DN. If the supplier server authenticates to the consumer server with a SASL EXTERNAL bind, the bind DN is extracted from the SSL certificate unless the `sslMapCertificate` configuration option's first value is set to `replace`. See page 105 for more information about certificate mapping.

If the `ibm-slapdMasterDN` attribute value in an `ibm-slapdReplication` entry matches the bind DN, the supplier server (or user) is allowed master level access to all replication contexts. If the `ibm-slapdMasterDN` attribute value in an `ibm-slapdSupplier` entry matches the bind DN, the supplier server (or user) is only allowed master level access to the replication contexts indicated by the multi-valued `ibm-replicaSubtree` attribute value.

**Note:** The consumer server credentials entry must be present on both the consumer and supplier servers and reside under the `cn=configuration` suffix in the CDBM backend. The topology entries are the only way for the servers to know their roles in the topology as a whole, therefore, are needed on all the servers in the topology.

<table>
<thead>
<tr>
<th>Attribute description and example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>cn</strong></td>
</tr>
<tr>
<td>A required attribute that specifies the common name of the consumer server credentials entry.</td>
</tr>
</tbody>
</table>

Example:

```
cn: master server
```
### Table 68. ibm-slapdReplication objectclass schema definition (required and optional attributes) (continued)

<table>
<thead>
<tr>
<th>Attribute description and example</th>
</tr>
</thead>
<tbody>
<tr>
<td>ibm-slapdMasterDN</td>
</tr>
<tr>
<td>Specifies the distinguished name (DN) that the supplier server uses to authenticate with the consumer server.</td>
</tr>
<tr>
<td>If the supplier server authenticates to the consumer server with a simple bind, this value should match the replicaBindDN attribute value in the simple bind supplier server credentials entry used by the replication agreement entry. If the supplier server authenticates to the consumer server with a SASL EXTERNAL bind, this value should match the bind DN extracted from the SSL certificate unless the sslMapCertificate configuration option's first value is set to replace. See page 105 for more information about certificate mapping.</td>
</tr>
<tr>
<td>Example:</td>
</tr>
<tr>
<td>ibm-slapdMasterDN: cn=supplier,cn=localhost</td>
</tr>
<tr>
<td>ibm-slapdMasterPW</td>
</tr>
<tr>
<td>Contains the simple bind authentication information needed for the replicating server to authenticate with the consumer server using the ibm-slapdMasterDN. This password value should match the replicaCredentials attribute value in the simple bind supplier server credentials entry used by the replication agreement entry.</td>
</tr>
<tr>
<td>This password value is encrypted if it is added or modified when the secretEncryption configuration option is set to AES or DES in the CDBM backend. If secretEncryption is set to AES or DES, this improves directory security because the password is no longer stored in the directory in clear text.</td>
</tr>
<tr>
<td>If a SASL EXTERNAL bind is used, this attribute value should not be specified.</td>
</tr>
<tr>
<td><strong>Note:</strong> This value is only used if the entry specified in the ibm-slapdMasterDN attribute value does not reside under a configured suffix in the LDAP server.</td>
</tr>
<tr>
<td>Example:</td>
</tr>
<tr>
<td>ibm-slapdMasterPW: secret</td>
</tr>
<tr>
<td>ibm-slapdMasterReferral</td>
</tr>
<tr>
<td>A single valued attribute that contains the LDAP URL of the supplier server. The LDAP URL syntax is completely documented in RFC 2255: The LDAP URL Format.</td>
</tr>
<tr>
<td>If an update operation is done by a user other than the supplier server or the LDAP administrator with Server Administration control, this value is returned as one of the referral values.</td>
</tr>
<tr>
<td>See Replication topology hints and tips for more information about referrals with advanced replication.</td>
</tr>
<tr>
<td>Example:</td>
</tr>
<tr>
<td>ibm-replicaReferralURL: ldap://master1.ibm.com:500</td>
</tr>
<tr>
<td>ibm-slapdNoReplConflictResolution</td>
</tr>
<tr>
<td>A boolean (true or false) indicating whether the consumer server participates in replication conflict resolution. If set to true, the consumer server does not participate in conflict resolution. If set to false or the attribute is not specified, the consumer server does participate in conflict resolution.</td>
</tr>
<tr>
<td>Conflict resolution is used to automatically attempt to resolve conflicts with entries that are no longer synchronized between a supplier and consumer server. The modifyTimestamp attribute value of the entry is used to detect a conflict between the two servers.</td>
</tr>
<tr>
<td>Example:</td>
</tr>
<tr>
<td>ibm-slapdNoReplConflictResolution: true</td>
</tr>
</tbody>
</table>

For the examples in Table 68, the supplier server located at master1.ibm.com on non-secure port 500 does a simple bind to the consumer server by binding with the cn=supplier,cn=localhost entry and specifying a password of secret. The consumer server is not configured for conflict resolution.
Table 69. *ibm-slapdSupplier* objectclass schema definition (required and optional attributes)

<table>
<thead>
<tr>
<th>Attribute description and example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>cn</strong></td>
</tr>
<tr>
<td>A required attribute that specifies the common name of the consumer server credentials entry.</td>
</tr>
<tr>
<td>Example:</td>
</tr>
<tr>
<td>cn: master server</td>
</tr>
</tbody>
</table>

| **ibm-slapdMasterDN** |
| Specifies the distinguished name (DN) that the supplier server uses to authenticate with the consumer server. |
| If the supplier server authenticates to the consumer server with a simple bind, this value should match the **replicaBindDN** attribute value in the simple bind supplier server credentials entry used by the replication agreement entry. |
| If the supplier server authenticates to the consumer server with a SASL EXTERNAL bind, this value should match the bind DN extracted from the SSL certificate unless the **sslMapCertificate** configuration option's first value is set to *replace*. See page [105](#) for more information about certificate mapping. |
| Example: |
| ibm-slapdMasterDN: cn=supplier,cn=localhost |

| **ibm-slapdMasterPW** |
| Contains the simple bind authentication information needed for the replicating server to authenticate with the consumer server using the **ibm-slapdMasterDN**. This password value should match the **replicaCredentials** attribute value in the simple bind supplier server credentials entry used by the replication agreement entry. |
| This password value is encrypted if it is added or modified when the **secretEncryption** configuration option is set to *AES* or *DES* in the CDBM backend. If **secretEncryption** is set to *AES* or *DES*, this improves directory security because the password is no longer stored in the directory in clear text. |
| If a SASL EXTERNAL bind is used, this attribute value should not be specified. |
| **Note**: This value is only used if the entry specified in the **ibm-slapdMasterDN** attribute value does not reside under a configured suffix in the LDAP server. |
| Example: |
| ibm-slapdMasterPW: secret |

| **ibm-slapdReplicaSubtree** |
| A multi-valued attribute that specifies the distinguished names of replication contexts that are subject to this consumer server credentials entry. |
| The bound user has master server level access to the replication contexts that are specified for this attribute. |
| Example: |
| ibm-slapdReplicaSubtree: o=ibm |

For the examples in Table 69, when the supplier server replicates updates to the o=ibm replication context on the consumer server, the supplier server performs a simple bind using the cn=supplier,cn=localhost entry and specifying a password of secret.
Things to consider before configuring advanced replication

Before setting up an advanced replication configuration, there are some administrative responsibilities that must be considered. 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 advanced 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.

Detailed status and error information is available to the LDAP administrator by querying the operational attributes in the replication agreement entries. See Monitoring and diagnosing advanced replication problems for a description of the information available. Configuring multiple master servers adds to the potential error cases that an LDAP 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 advanced replication topology. The advanced replication conflict resolution support is designed to detect and resolve conflicts that may occur. See Replication conflict resolution for more information about replication conflict resolution.

The following should be considered when planning an advanced replication environment:

1. Determine if an existing Directory Information Tree (DIT) subtree is to be introduced into a replication topology or if a new subtree is to be added to the server after the replication topology is established. It is recommended that all servers that serve as a supplier server are put into maintenance mode until the replication topology entries are loaded on all servers. This ensures that external updates to the subtree are not lost while configuring advanced replication. See Advanced replication maintenance mode for more information.
   a. If using an existing subtree for advanced replication:
      1) Modify the subtree to add the auxiliary objectclass ibm-replicationContext. If the ibm-replicationContext auxiliary objectclass is added to a non-suffix level entry in the directory, explicit aclEntry and entryOwner attribute values are required. The entryOwner attribute value must start with an access-id:
      2) Unload the entire subtree to an LDIF file by using the ds2ldif utility. See ds2ldif utility for additional information about the ds2ldif utility
      3) For each server participating in the replication topology, add the unloaded entries to the server by using the ldapadd or ldif2ds utilities. See IBM Tivoli Directory Server Client Programming for z/OS for more information about the ldapadd utility. See ldif2ds utility for additional information about the ldif2ds utility.
   b. If using a new subtree for advanced replication, verify that the subtree has an auxiliary objectclass of ibm-replicationContext. For each server participating in the replication topology, add the same entries to all servers by using the ldapadd or ldif2ds utilities. See IBM Tivoli Directory Server Client Programming for z/OS for more information about the ldapadd utility. See ldif2ds utility for additional information about the ldif2ds utility.

2. For each consumer server in the replication topology, load master bind and referral information under the cn=configuration suffix in the CDBM backend. Consumer server credential entries with an objectclass of ibm-slapdSupplier or ibm-slapdReplication must be added. If using a consumer server credentials entry with an objectclass of ibm-slapdSupplier, the replication context must be added to the ibm-slapdReplicaSubtree attribute value. See Consumer server entries for more information.

3. For each supplier server in the replication topology, supplier server credential entries must be added for each unique consumer server in the replication context. A supplier server credential entry enables the supplier server to authenticate with the consumer server by using a simple or SASL EXTERNAL bind. If the objectclass of the supplier server credentials entry is ibm-replicationCredentialsSimple, a simple bind is used. If the objectclass of the supplier server credentials entry is ibm-replicationCredentialsExternal, a SASL EXTERNAL bind is used.
Note: There are no requirements for placing supplier server credential entries within a specific subtree. If supplier server credential entries should not be replicated, use the `cn=localhost` subtree that does not allow the replication of entries. If supplier server credential entries should be replicated outside the scope of the replication context being configured, consider using the `cn=ibmpolicies` subtree. When the `cn=ibmpolicies` subtree is configured for advanced replication, schema modifications are also replicated. See Replication of schema updates for more information about schema replication.

The following steps are used to deploy the replication topology on all servers:

1. On a supplier server in the topology, use the `ldapadd` utility with the Server Administration and the Do Not Replicate controls to add the replication topology entries. The replication topology entries are the following:
   b. Replica group with an objectclass of `ibm-replicaGroup`. See Replica groups for more information.
   c. Replica subentry with an objectclass of `ibm-replicaSubentry`. See Replica subentries for more information.
   d. Replication agreement with an objectclass of `ibm-replicationAgreement`. See Replication agreements for more information.

2. On the replication context added in the previous step, use the Replication topology extended operation in the `ldapexop` utility. This synchronizes all replication topology entries for each consumer server in the replication context. See `ldapexop` utility for more information about the `ldapexop` utility.

When the above steps are complete, the supplier servers should be moved out of maintenance mode.

## Advanced replication configuration examples

This topic provides examples of the different replication topologies that can be configured. It provides example LDIF data that includes the hostnames, IP addresses, ports, server IDs and passwords.

### Suppliers and consumers

In advanced replication, updates are propagated from one LDAP server to another through a replication queue. The server that pushes updates into the replication queue is called a supplier. The server that absorbs these changes is called the consumer. The queue is maintained on the supplier.

- **Hostnames and ports:** Provide the supplier with enough information to connect to the consumer.
- **Server IDs:** Strings that enable one LDAP server to identify other LDAP servers in the topology.
- **Bind DNs and passwords:** The supplier connects to the consumer using the LDAP protocol. In LDAP terminology this is called a bind. The bind requires a bind Distinguished Name (DN) and a password.

The following examples demonstrate setup for a topology consisting of a maximum of three servers.

**Note:** The host name of the consumer should resolve correctly from the supplier. If not, the supplier cannot connect to the consumer and advanced replication fails.

<table>
<thead>
<tr>
<th>Server</th>
<th>Host name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server 1</td>
<td>server1.us.ibm.com</td>
</tr>
<tr>
<td>Server 2</td>
<td>server2.us.ibm.com</td>
</tr>
<tr>
<td>Server 3</td>
<td>server3.us.ibm.com</td>
</tr>
</tbody>
</table>

Each LDAP server in the above table is listening on port 389, which is the default LDAP port.
These examples assume that a simple bind is being done from the supplier server to the consumer server.
Each example assumes the following bind DN and password for all supplier-consumer agreements.

- **DN:** cn=bindtoconsumer
- **Password:** iamsupplier

**Server ID**

In the examples, the server ID of each server is the role of that server in the topology. That is, in the
Master-Replica topology, the master identifies the server ID as Master and the replica is identified as
Replica. In the Peer-to-Peer topology, one peer is Peer1 and the other is Peer2. In the
Master-Forwarder-Replica topology, the master is Master, the forwarder is Forwarder and the replica is
Replica. In the Gateway topology, the gateway servers are Gateway1 and the other is Gateway2 and the
replica is Replica.

It is recommended that the server ID for a server not change. When the z/OS LDAP server is first
configured with a CDBM backend, the server ID is generated as an IBM entry UUID value in the
**ibm-slapdServerID** attribute value in the **cn=configuration** entry. For convenience, the server ID is
published in the rootDSE entry's attribute **ibm-serverID**. The server ID is only allowed to be modified
when there are no replica subentries defined in the server. See **Table 10** for more information about the
**ibm-slapdServerID** attribute value.

**Advanced replication related entries summary**

For convenience, this section quickly summarizes the various types of entries that are used to build an
advanced replication topology.

**Supplier server entries**

- **Replication context:** This is the root entry for the subtree that is to be replicated. It must have an
auxiliary objectclass of **ibm-replicationContext**. To replicate a subtree **o=ibm,c=us**, the replication
context might be:

  - dn: o=ibm,c=us
  - objectclass: top
  - objectclass: organization
  - objectclass: ibm-replicationContext
  - o: ibm

All other replication entries except for the credential and schedule entries must be under the replication
context. The credential and schedule entries can be anywhere in the DIT.

- **Replica group:** This entry is not very important apart from the fact that all the advanced replication
related entries exist under this entry. It must have the **ibm-replicaGroup** objectclass. For example,

  - dn: ibm-replicaGroup=default, o=ibm,c=us
  - objectclass: top
  - objectclass: ibm-replicaGroup
  - ibm-replicaGroup: default

- **Replica subentry:** These types of entries declare the servers that are taking part in the advanced
replication topology. Each server participating in the topology has one subentry. If a server is
represented by more than one subentry under a replication context, unexpected behavior might result.
This is done by having more than one subentry under a replication context containing the same
**ibm-replicaServerId** attribute value. This entry has the **ibm-replicaSubentry** objectclass. For example,

  - dn: ibm-replicaServerId=Peer1, ibm-replicaGroup=default, o=ibm,c=us
  - objectclass: top
  - objectclass: ibm-replicaSubentry
  - ibm-replicaServerId: Peer1
  - ibm-replicaServerIsMaster: true
  - cn: Peer1
  - description: Peer1
As shown in the replica subentry example, the entry has the server ID of the participating server, Peer1. It has an attribute called `ibm-replicationServersIsMaster`. When this attribute is set to true, the server is a read-write copy.

- **Replication agreements**: These types of entries occur under replica subentries. When these entries appear under a specific server's replica subentry, they define a replication agreement from that server to some other server in the topology. For example,

```plaintext
dn: cn=Peer2, ibm-replicaServerId=Peer1,ibm-replicaGroup=default,o=ibm,c=us
objectclass: top
objectclass: ibm-replicationAgreement
cn: Peer2
ibm-replicaConsumerId: Peer2
ibm-replicaCredentialsDN: cn=Peer1BindCredentials, cn=localhost
description: Replication agreement from Peer1 to Peer2
```

The replication agreement example is from Peer1 to Peer2. The supplier is Peer1 as the agreement occurs under the subentry for Peer1. The consumer is Peer2. The server Peer2 is on server2.us.ibm.com and is listening on port 389. Peer1 binds to Peer2 using the credentials defined in the entry (cn=Peer1BindCredentials,cn=localhost).

- **Replication credentials**: If a simple bind is used by the supplier server to authenticate with the consumer server, this entry defines the bind DN and password that is used. This credential entry uses the `ibm-replicationCredentialsSimple` objectclass. If a SASL EXTERNAL bind is used by the supplier server to authenticate with the consumer server, see Credentials entries for information about the `ibm-replicationCredentialsExternal` objectclass. For example,

```plaintext
dn: cn=Peer1BindCredentials, cn=localhost
objectclass: ibm-replicationCredentialsSimple
cn: ReplicaBindCredentials
replicaBindDN: cn=bindtoconsumer
replicaCredentials: iamsupplier
description: Bind Credentials on Peer1 to be used to bind to other servers.
```

The replication credential example defines the `replicaBindDN` as cn=bindtoconsumer and the password as iamsupplier. Take note of the description. The same credentials entry can be used for multiple replication agreements.

**Consumer server entries**

The only replication related entry needed on the consumer server is the consumer server credentials entry. The consumer server credential entry identifies the distinguished name and optionally the password value that the supplier server uses to authenticate with the consumer server. There are two types of credential entries that can be used on the consumer.

**Type 1 example:**

```plaintext
dn: cn=Master server,cn=configuration
objectclass: ibm-slapdReplication
cn: master server
ibm-slapdMasterDN: cn=bindtoconsumer
ibm-slapdMasterPW: iamsupplier
ibm-slapdMasterReferral: ldap://localhost:1389
```

**Type 2 example:**

```plaintext
dn: cn=Supplier s1,cn=configuration
objectclass: ibm-slapdSupplier
cn: Supplier s1
ibm-slapdMasterDN: cn=bindtoconsumer
ibm-slapdMasterPW: iamsupplier
ibm-slapdReplicaSubtree: o=ibm,c=us
```

The use of the credential established by type 2 is limited to the `ibm-slapdReplicaSubtree` only. Therefore, suppliers binding with bind DN as cn=bindtoconsumer and password as iamsupplier supplies only to the
o=ibm,c=us subtree, unless another credential entry gives rights to another subtree. The type 1 credential entry is global across the LDAP server. When a type 2 entry is defined in the cn=configuration suffix in the CDBM backend, any subtree can be supplied to if a supplier authenticates with bind DN of cn=bindtoconsumer and password of iamsupplier.

Note: The consumer server credentials entry must be present on both the consumer and supplier servers and reside under the cn=configuration suffix in the CDBM backend. The topology entries are the only way for the servers to know their roles in the topology as a whole, therefore, are needed on all the servers in the topology.

Creating a master-replica topology
This example describes deploying the most basic of all the topologies, the master-replica topology. It has one read-write server and one read-only server.

![Master-replica topology](image)

Figure 48. Master-replica topology

- The first step when building a topology is to define:
  1. Replication context: o=ibm,c=us
  2. Supplier(s): LDAP server on server1.us.ibm.com:389 is the only supplier. The server ID is Master. It supplies updates to the LDAP server on server2.us.ibm.com:389.
  3. Consumer(s): LDAP server on server2.us.ibm.com:389 is the only consumer. The server ID is Replica. It consumes updates from the LDAP server on server1.us.ibm.com:389.
  4. Read-write server(s): LDAP server on server1.us.ibm.com:389, with ID Master is the only read-write server.
  5. Read-only server(s): LDAP server on server2.us.ibm.com:389, with ID Replica is the only read-only server.

Configuration changes: Because of these examples, some CDBM changes are needed for the master and the replica server for replication to work correctly.

Note: These are done here for the purpose of this example ONLY. The server ID is only allowed to be modified when there are no replica subentries defined in the server. See Table 10 for more information about the ibm-slapdServerID attribute value.

1. Server IDs:
   - On the master server, apply the following modify using the ldapmodify utility. See IBM Tivoli Directory Server Client Programming for z/OS for additional information about the ldapmodify utility.
     ```
     dn: cn=configuration
     changetype: modify
     replace: ibm-slapdserverid
     ibm-slapdserverid: Master
     ```
   - On the replica server, apply the following modify:
     ```
     dn: cn=configuration
     changetype: modify
     replace: ibm-slapdserverid
     ibm-slapdserverid: Replica
     ```

2. Consumer server credentials entry: Add this entry to the replica server using the ldapadd utility. See IBM Tivoli Directory Server Client Programming for z/OS for additional information about the ldapadd utility. For example:
The next step is to build the LDIF file for the topology. This LDIF file is called `masterreplica.ldif`. Copy each of these entries to `masterreplica.ldif` with the necessary changes in the subtree, server IDs, host names and ports.

1. **Replication context:**
   - If the subtree entry already exists, use the `ldapmodify` utility to modify the existing entry. For example,
     ```
     dn: o=ibm,c=us
     changetype: modify
     add: objectclass
     objectclass: ibm-replicationContext
     ```
   - If the subtree entry does not exist, add the entry with the `ibm-replicationContext` auxiliary objectclass by using the `ldapadd` utility. For example,
     ```
     dn: o=ibm,c=us
     changetype: add
     objectclass: top
     objectclass: organization
     objectclass: ibm-replicationContext
     o: ibm
     ```

2. Add the replica group entry using the `ldapadd` utility. For example:
   ```
   dn: ibm-replicaGroup=default, o=ibm,c=us
   changetype: add
   objectclass: top
   objectclass: ibm-replicaGroup
   ibm-replicaGroup: default
   ```

3. **Replica subentries:** Because there are two LDAP servers in the topology, you need to add two replica subentries; one for the master server and another one for the replica server by using the `ldapadd` utility. For example, for a subentry for the master:
   ```
   dn: ibm-replicaServerId=Master, ibm-replicaGroup=default, o=ibm,c=us
   changetype: add
   objectclass: top
   objectclass: ibm-replicaSubentry
   ibm-replicaServerId: Master
   ibm-replicationServerIsMaster: true
   cn: Master
   description: Master server of the topology
   ```
   For example, for a subentry for the replica:
   ```
   dn: ibm-replicaServerId=Replica, ibm-replicaGroup=default, o=ibm,c=us
   changetype: add
   objectclass: top
   objectclass: ibm-replicaSubentry
   ibm-replicaServerId: Replica
   ibm-replicationServerIsMaster: false
   cn: Replica
   description: Replica server of the topology
   ```

   Take note of the subentries carefully. The subentry for the master uses the server ID `Master` and has the server declared as a master server. This server receives updates from clients. The subentry for the replica has the server ID of `Replica` and has the server declared as a non-master server. This server cannot receive updates from LDAP clients that bind to it.

**Note:** The number of subentries is not dependent on the number of physical servers in the topology. Rather, it is dependent on the number of LDAP servers in the topology.
4. **Supplier server credentials entry:** This step defines the credentials that the master uses to bind to the replica. Add an entry with the `ldapadd` utility. For example,

```
dn: cn=ReplicaBindCredentials, o=ibm,c=us
changetype: add
objectclass: ibm-replicationCredentialsSimple
cn: ReplicaBindCredentials
replicaBindDN: cn=bindtoconsumer
replicaCredentials: iamsupplier
description: Bind Credentials on master to bind to replica
```

Note that the DN and password are the same as the pair that was added in the consumer server credentials entry section above. Add the entry with the `ldapadd` utility. As a result, updates to this object results in the server attempting to replicate.

5. **Replication agreements:** There is one supplier-consumer relationship in this advanced replication topology. The master supplies updates to the `o=ibm,c=us` subtree to the replica that consumes the changes. Therefore, there is only one agreement: From the master to the replica. Note that the number of agreements is dependent upon the number of supplier-consumer relationships in the topology. For example,

```
dn: cn=Replica, ibm-replicaServerId=Master,ibm-replicaGroup=default,o=ibm,c=us
changetype: add
objectclass: top
objectclass: ibm-replicationAgreement
cn: Replica
ibm-replicaConsumerId: Replica
ibm-replicaCredentialsDN: cn=ReplicaBindCredentials, o=ibm,c=us
description: Replication agreement from master to replica
```

The entry above is under the subentry for the master. It supplies to a consumer with an ID as Replica. The replica URL is `ldap://server2.us.ibm.com:389` meaning the replica is listening on server2.us.ibm.com on port 389. This agreement uses the credentials that were created in the last step for the master to bind to the replica. That means the master binds to the replica with a bind DN `cn=bindtoconsumer` and the password `iamsupplier`. Note that there is no agreement under the subentry for the replica. This is natural as the replica is a read-only copy and cannot receive any client updates, therefore, there is no point in having an agreement, because there are no updates to propagate.

- Now that the replication entries have been added, the `masterreplica.ldif` file is as follows:

```
dn: o=ibm,c=us
changetype: add
objectclass: top
objectclass: organization
objectclass: ibm-replicationContext
o: ibm

dn: ibm-replicaGroup=default, o=ibm,c=us
changetype: add
objectclass: top
objectclass: ibm-replicaGroup
ibm-replicaGroup: default

dn: ibm-replicaServerId=Master,ibm-replicaGroup=default, o=ibm,c=us
changetype: add
objectclass: top
objectclass: ibm-replicaSubentry
ibm-replicaServerId: Master
ibm-replicationServerIsMaster: true
cn: Master
description: Master server of the topology.

dn: ibm-replicaServerId=Replica,ibm-replicaGroup=default, o=ibm,c=us
```
Next, add the `masterreplica.ldif` file on the master server. Perform the `ldapadd` command on the master where the `masterreplica.ldif` file was created. For example:

```
ldapadd -h server1.us.ibm.com -p 389 -D adminDN -w adminPW -f masterreplica.ldif -k -L
```

The `-L` option on the `ldapadd` utility sends the Do Not Replicate control to the server that indicates not to replicate the topology now. The `-k` option sends the Server Administration control to the server so that the addition of entries continues even when the subtree becomes read-only because of a server ID mismatch.

Next add the replication topology to the replica also. Use the `ldapexop` command. For example,

```
ldapexop -h server1.us.ibm.com -p 389 -D adminDN -w adminPW -op repltopology -rc o=ibm,c=us
```

This is an example of the Replication topology extended operation. It propagates the advanced replication topology to all the consumers defined under the `o=ibm,c=us` replication context. See the `ldapexop` utility for additional information about the `ldapexop` utility.

After the `ldapadd` and `ldapexop` commands are performed successfully, the master-replica topology is ready. The master accepts updates on the `o=ibm,c=us` subtree and propagates them to the replica. The replica does not accept updates. It returns a referral to the master in case a client tries to update it, however, it can handle searches.
Creating a peer-to-peer replication topology

The peer-to-peer replication topology does not differ much from the master-replica topology. It also has two servers, but, both the servers are now read-write servers. They both supply changes to each other.

![Peer-to-peer topology diagram](image)

**Figure 49. Peer-to-peer topology**

- The first step when building a topology is to define:
  1. **Replication context:** `o=ibm,c=us`
  4. **Read-write server(s):** LDAP servers `Peer1` and `Peer2` on `server1.us.ibm.com` and `server2.us.ibm.com` are read-write servers.
  5. **Read-only server(s):** There are no read-only servers in this topology.

**Configuration changes:** Some CDBM changes need to be done to both the `Peer1` and `Peer2` servers for replication to work correctly. Configured servers should be current. If you are using the same servers that you used for the Master-Replica setup, undo the changes that you made in Creating a master-replica topology.

**Note:** These are done here for the purpose of this example ONLY. The server ID is only allowed to be modified when there are no replica subentries defined in the server. See Table 10 for more information about the `ibm-slapdServerID` attribute value.

1. **Server IDs:**
   - On the `Peer1` server (`server1.us.ibm.com`), apply the following modify using the `ldapmodify` utility. See IBM Tivoli Directory Server Client Programming for z/OS for additional information about the `ldapmodify` utility.
     ```
     dn: cn=configuration
     changetype: modify
     replace: ibm-slapdserverid
     ibm-slapdserverid: Peer1
     ```
   - On the `Peer2` server (`server2.us.ibm.com`), apply the following modify using the `ldapmodify` utility.
     ```
     dn: cn=configuration
     changetype: modify
     replace: ibm-slapdserverid
     ibm-slapdserverid: Peer2
     ```

2. **Consumer server credentials entry:** Add this first entry to the `Peer1` server and the second entry to the `Peer2` server using the `ldapadd` utility. See IBM Tivoli Directory Server Client Programming for z/OS for additional information about the `ldapadd` utility.

**Note:** An alternative is to add entries with an `ibm-slapdSupplier` objectclass. Also, each peer uses the other peer as the referral. This is useful if a peer must become a read only replica or a
forwarding server.

For example, for a consumer side credentials entry for the Peer1:

dn: cn=Master server,cn=configuration
changetype: add
objectclass: ibm-slapdReplication
cn: master server
ibm-slapdMasterDN: cn=bindtouser
ibm-slapdMasterPW: iamsupplier

For example, for a credential subentry for the Peer2:

dn: cn=Master server,cn=configuration
changetype: add
objectclass: ibm-slapdReplication
cn: master server
ibm-slapdMasterDN: cn=bindtouser
ibm-slapdMasterPW: iamsupplier

- The next step is to build the LDIF file for the replication topology. This LDIF file is called peer2peer.ldif. Copy each of these entries to peer2peer.ldif with the necessary changes in the subtree, server IDs, host names and ports.

1. **Replication context:**
   - If the subtree entry already exists, use the *ldapmodify* utility to modify the existing entry. For example,
   
   ```
   dn: o=ibm,c=us
   changetype: modify
   add: objectclass
   objectclass: ibm-replicationContext
   ```

   - If the subtree entry does not exist, add the entry with the *ibm-replicationContext* auxiliary objectclass by using the *ldapadd* utility. For example,

   ```
   dn: o=ibm,c=us
   changetype: add
   objectclass: top
   objectclass: organization
   objectclass: ibm-replicationContext
   o: ibm
   ```

2. **Add the replica group entry using the *ldapadd* utility.** For example:

   ```
   dn: ibm-replicaGroup=default, o=ibm,c=us
   changetype: add
   objectclass: top
   objectclass: ibm-replicaGroup
   ibm-replicaGroup: default
   ```

3. **Replica subentries:** Because there are two LDAP servers in the topology, you need to add two replica subentries; one for Peer1 and another one for the Peer2 server by using the *ldapadd* utility. For example, for a subentry for Peer1:

   ```
   dn: ibm-replicaServerId=Peer1,ibm-replicaGroup=default, o=ibm,c=us
   changetype: add
   objectclass: top
   objectclass: ibm-replicaSubentry
   ibm-replicaServerId: Peer1
   ibm-replicationServerIsMaster: true
   cn: Peer1
description: Subentry for Peer1
   ```

   For example, for a subentry for Peer2:

   ```
   dn: ibm-replicaServerId=Peer2,ibm-replicaGroup=default, o=ibm,c=us
   changetype: add
   objectclass: top
   objectclass: ibm-replicaSubentry
   ```
4. **Supplier server credentials entry:** This step defines the credentials that Peer1 and Peer2 use to bind to each other. Add the entry with the `ldapadd` utility. For example,

```
dn: cn=ReplicaBindCredentials, o=ibm,c=us
changetype: add
objectclass: ibm-replicationCredentialsSimple
 cn: ReplicaBindCredentials
 replicaBindDN: cn=bindtoconsumer
 replicaCredentials: iamsupplier
 description: Bind Credentials on Peer1 and Peer2 to bind to each other.
```

5. **Replication agreements:** There are two supplier-consumer relationships in this topology. Peer1 supplies updates made to the `o=ibm,c=us` subtree to Peer2, that consumes the changes. Peer2 also accepts updates from clients on the `o=ibm,c=us` subtree and sends them to Peer1, that consumes the changes. Therefore, there are two agreements:

   a. from Peer1 to Peer2
   b. from Peer2 to Peer1

   **Note:** The number of agreements is dependent upon the number of supplier-consumer relationships in the topology.

   For example, a replication agreement from Peer1 to Peer2:

```
dn: cn=Peer2, ibm-replicaServerId=Peer1,ibm-replicaGroup=default,o=ibm,c=us
changetype: add
objectclass: top
objectclass: ibm-replicationAgreement
 cn: Peer2
 ibm-replicaConsumerId: Peer2
 ibm-replicaCredentialsDN: cn=ReplicaBindCredentials, o=ibm,c=us
 description: Replication agreement from Peer1 to Peer2
```

   For example, a replication agreement from Peer2 to Peer1:

```
dn: cn=Peer1, ibm-replicaServerId=Peer2,ibm-replicaGroup=default,o=ibm,c=us
changetype: add
objectclass: top
objectclass: ibm-replicationAgreement
 cn: Peer1
 ibm-replicaConsumerId: Peer1
 ibm-replicaCredentialsDN: cn=ReplicaBindCredentials, o=ibm,c=us
 description: Replication agreement from Peer2 to Peer1
```

   The two agreements for this topology are shown above. The first one is the agreement from Peer1 to Peer2. It is under the Peer1 subentry. The second one is from Peer2 to Peer1 and it is under the Peer2 subentry. Note the use of the same credentials entry for both agreements. This is acceptable. The credentials entry is added to both Peer1 and Peer2.

   - Now that the replication entries have been added, the `peer2peer.ldif` file is as follows:
   ```
   dn: o=ibm,c=us
   changetype: add
   objectclass: top
   objectclass: organization
   objectclass: ibm-replicationContext
   ```
Next, add the `peer2peer.ldif` file on the Peer1 server. Perform the `ldapadd` command on the Peer1 where the `peer2peer.ldif` file was created. For example,

```bash
ldapadd -h server1.us.ibm.com -p 389 -D adminDN -w adminPW -f peer2peer.ldif -k -L
```

The `-L` option on the `ldapadd` utility sends the Do Not Replicate control to the server that indicates not to replicate the topology now. The `-k` option sends the Server Administration control to the server so that the addition of entries continues even when the subtree becomes read-only because of a server ID mismatch.

Next add the replication topology to the Peer2 also. Use the `ldapexop` command. For example,
This is an example of the Replication topology extended operation. It propagates the advanced replication topology to all the consumers defined under the o=ibm,c=us replication context. See [ldapexop utility] for additional information about the ldapexop utility.

- After the ldapadd and ldapexop commands are performed successfully, the peer-to-peer topology is ready. Both peers accept updates on and send them to the other peer.

## Creating a master-forwarder-replica (cascading) topology

Another advanced replication topology is the master-forwarder-replica or cascading replication topology.

![Master-forwarder-replica topology](image)

The forwarder server is a specialized replica server. As previously stated, replica servers are read-only as is a forwarder server. Replica servers do not transmit changes that are consumed by them. However, forwarder servers replicate changes that they have consumed. To supply changes further down the topology, forwarders have agreements under their subentries.

**Note:** Gateway replication topologies are similar, however, forwarders are specialized replicas while gateways are specialized masters.

This topology needs to have one more server included: server3.us.ibm.com

- The first step when building a topology is to define:
  1. **Replication context:** o=ibm,c=us
  4. **Read-write server(s):** LDAP server on server1.us.ibm.com:389, with ID Master is the only read-write server.
  5. **Read-only server(s):** LDAP server on server1.us.ibm.com:389, with IDs, Forwarder and Replica are read-only.

**Configuration changes:** Some configuration changes need to be made to the Master, Forwarder, and Replica servers for replication to work correctly. Configured servers should be current. If reusing the same servers that were used for the Master-Replica setup, undo the changes that were made in [Creating a master- replica topology] if reusing the same servers that were used for the Peer-Peer setup, undo the changes that were made in [Creating a peer-to-peer replication topology].
Note: These are done here for the purpose of this example ONLY. The server ID is only allowed to be modified when there are no replica subentries defined in the server. See Table 10 for more information about the ibm-slapdServerID attribute value.

1. Server IDs:
   - On the Master server (server1.us.ibm.com), apply the following modify using the ldapmodify utility. See [IBM Tivoli Directory Server Client Programming for z/OS] for additional information about the ldapmodify utility.
     - dn: cn=configuration
     - changetype: modify
     - replace: ibm-slapdserverid
     - ibm-slapdserverid: Master
   - On the Forwarder server (server2.us.ibm.com), apply the following modify using the ldapmodify utility.
     - dn: cn=configuration
     - changetype: modify
     - replace: ibm-slapdserverid
     - ibm-slapdserverid: Forwarder
   - On the Replica server (server3.us.ibm.com), apply the following modify using the ldapmodify utility.
     - dn: cn=configuration
     - changetype: modify
     - replace: ibm-slapdserverid
     - ibm-slapdserverid: Replica

2. Consumer server credentials entry: Add this first entry to the Forwarder server and the second entry to the Replica server using the ldapadd utility. See [IBM Tivoli Directory Server Client Programming for z/OS] for additional information about the ldapadd utility.

   Note: An alternative is to add entries with an ibm-slapdSupplier objectclass.
   For example, for a consumer server credentials entry for the Forwarder:
   - dn: cn=Master server,cn=configuration
   - changetype: add
   - cn: master server
   - ibm-slapdMasterDN: cn=bindtoconsumer
   - ibm-slapdMasterPW: iamsupplier
   - objectclass: ibm-slapdReplication

   For example, for a consumer server credentials entry for the Replica:
   - dn: cn=Master server,cn=configuration
   - changetype: add
   - cn: master server
   - ibm-slapdMasterDN: cn=bindtoconsumer
   - ibm-slapdMasterPW: iamsupplier
   - objectclass: ibm-slapdReplication

• The next step is to build the LDIF file for the replication topology. This LDIF file is called mfr.ldif. Copy each of these entries to mfr.ldif with the necessary changes in the subtree, server IDs, host names and ports.

1. Replication context:
   - If the subtree entry already exists, use the ldapmodify utility to modify the existing entry. For example,
     - dn: o=ibm,c=us
     - changetype: modify
     - add: objectclass
     - objectclass: ibm-replicationContext
   - If the subtree entry does not exist, add the entry with the ibm-replicationContext auxiliary objectclass by using the ldapadd utility. For example,
2. Add the replica group entry using the `ldapadd` utility. For example:

dn: ibm-replicaGroup=default, o=ibm,c=us
changetype: add
objectclass: top
objectclass: ibm-replicaGroup
ibm-replicaGroup: default

3. **Replica subentries:** Because there are three LDAP servers in the topology, you need to add three replica subentries; one for Master, one for Forwarder, and another one for the Replica server by using the `ldapadd` utility. For example, for a subentry for Master:

dn: ibm-replicaServerId=Master,ibm-replicaGroup=default, o=ibm,c=us
changetype: add
objectclass: top
objectclass: ibm-replicaSubentry
ibm-replicaServerId: Master
ibm-replicationServerIsMaster: true
cn: Master
description: Subentry for Master

For example, for a subentry for Forwarder:

dn: ibm-replicaServerId=Forwarder,ibm-replicaGroup=default, o=ibm,c=us
changetype: add
objectclass: top
objectclass: ibm-replicaSubentry
ibm-replicaServerId: Forwarder
ibm-replicationServerIsMaster: false
cn: Forwarder
description: Subentry for the Forwarder

For example, for a subentry for Replica:

dn: ibm-replicaServerId=Replica,ibm-replicaGroup=default, o=ibm,c=us
changetype: add
objectclass: top
objectclass: ibm-replicaSubentry
ibm-replicaServerId: Replica
ibm-replicationServerIsMaster: false
cn: Replica
description: Subentry for the Replica

The subentry for Master identifies the server ID as Master and has the server declared as a master server. The server can receive updates from clients. The subentry for the Forwarder identifies the server ID as Forwarder, and is declared as a non-master server; it cannot get updates from clients. The subentry for the Replica has the server ID mentioned as Replica and is declared as a non-master server; it cannot get updates from clients.

4. **Supplier server credentials entry:** This step defines the credentials that the Master uses to bind to the Forwarder. Add the entry with the `ldapadd` utility. For example,

dn: cn=ReplicaBindCredentials, o=ibm,c=us
changetype: add
objectclass: ibm-replicationCredentialsSimple
cn: ReplicaBindCredentials
replicaBindDN: cn=bindtoconsumer
replicaCredentials: iamsupplier
description: Bind Credentials on master to bind to forwarder

5. **Replication agreements:** There are two supplier-consumer relationships in this topology. Master supplies updates made to the `o=ibm,c=us` subtree to Forwarder, that consumes the changes and then supplies these changes to the Replica. Therefore, there are two agreements:
a. from Master to Forwarder
b. from Forwarder to Replica

**Note:** The number of replication agreements is dependent upon the number of supplier-consumer relationships in the topology.

For example, a replication agreement from Master to Forwarder:
```plaintext
dn: cn=Forwarder, ibm-replicaServerId=Master,ibm-replicaGroup=default,o=ibm,c=us
changetype: add
objectclass: top
objectclass: ibm-replicationAgreement
cn: Forwarder
ibm-replicaConsumerId: Forwarder
ibm-replicaCredentialsDN: cn=ReplicaBindCredentials, o=ibm,c=us
description: Replication agreement from Master to Forwarder
```

For example, a replication agreement from Forwarder to Replica:
```plaintext
dn: cn=Replica, ibm-replicaServerId=Forwarder,ibm-replicaGroup=default,o=ibm,c=us
changetype: add
objectclass: top
objectclass: ibm-replicationAgreement
cn: Replica
ibm-replicaConsumerId: Replica
ibm-replicaCredentialsDN: cn=ReplicaBindCredentials, o=ibm,c=us
description: Replication agreement from Forwarder to Replica
```

The two agreements for this topology are shown above. The first one is the agreement from Master to Forwarder. It is under the Master subentry. The second one is from Forwarder to Replica and it is under the Forwarder subentry. Note the use of the same credentials entry for both agreements. This is acceptable. The consumer server credentials entry is added to both Master and Forwarder servers.

- Now that the replication entries have been added, the **mfr.ldif** file is as follows:
  ```plaintext
dn: o=ibm,c=us
changetype: add
objectclass: top
objectclass: organization
objectclass: ibm-replicationContext
o: ibm

  dn: ibm-replicaGroup=default, o=ibm,c=us
  changetype: add
  objectclass: top
  objectclass: ibm-replicaGroup
ibm-replicaGroup: default

  dn: ibm-replicaServerId=Master,ibm-replicaGroup=default, o=ibm,c=us
  changetype: add
  objectclass: top
  objectclass: ibm-replicaSubentry
ibm-replicaServerId: Master
ibm-replicationServerIsMaster: true
cn: Master
description: Subentry for Master.

  dn: ibm-replicaServerId=Forwarder,ibm-replicaGroup=default, o=ibm,c=us
  changetype: add
  objectclass: top
  objectclass: ibm-replicaSubentry
ibm-replicaServerId: Forwarder
ibm-replicationServerIsMaster: false
```
Next, add the mfr.ldif file on the Master server. Perform the ldapadd command on the Master where the mfr.ldif file was created. For example,
```
ldapadd -h server1.us.ibm.com -p 389 -D adminDN -w adminPW -f mfr.ldif -k -L
```

Next, add the mfr.ldif file on the Forwarder server. Perform the ldapadd command on the Forwarder server. For example,
```
ldapadd -h server2.us.ibm.com -p 389 -D adminDN -w adminPW -f mfr.ldif -k -L
```

The -L option on the ldapadd utility sends the Do Not Replicate control to the server that indicates not to replicate the topology now. The -k option sends the Server Administration control to the server so that the addition of entries continues even when the subtree becomes read-only because of a server ID mismatch.

Next, add the replication topology to the Replica also. Use the ldapexop command. For example,
```
ldapexop -h server2.us.ibm.com -p 389 -D adminDN -w adminPW -op repltopology -rc o=ibm,c=us
```

This is an example of the Replication topology extended operation. It propagates the advanced replication topology to the Replica defined under the o=ibm,c=us replication context. See ldapexop utility for additional information about the ldapexop utility.

After the ldapadd and ldapexop commands are performed successfully, the master-forwarder-replica topology is ready. The Master accepts updates, that go to the Forwarder and then to the Replica.
intend to add another replica to the topology, under the forwarder, you need to add another subentry for
the replica, and add an agreement from the forwarder to that replica.

Creating a gateway topology
A gateway topology requires at least two gateway servers. Gateway topologies are created in a similar
manner to a master-forwarder-replica (cascading) topology. A gateway topology includes gateway master
servers that forward all replication traffic from the local replication site where it resides to other gateway
servers in the replicating network. A gateway server also receives replication traffic from other gateway
servers within the replication network and forwards updates to all servers on its local replication site.

Gateway1

Gateway2

Replica

Figure 51. Gateway topology
In a gateway topology, gateway servers are distinguished from normal master servers by including the
ibm-replicaGateway objectclass for the replica subentry in the replication context. As previously stated, a
server is a master if the server ID in the replica subentry within the replication context equals the server’s
server ID and the subentry’s ibm-replicationServerIsMaster attribute is set to true. See Replica
subentries for more information about replica subentries.

The first step when building a topology is to define:
1. Replication context: cn=ibm,c=us
2. Supplier(s): LDAP server on server1.us.ibm.com:389 with server ID Gateway1 supplies updates to
   the LDAP server with server ID Gateway2 on server2.us.ibm.com:389. LDAP server on
   server2.us.ibm.com:389 with server ID Gateway2 supplies updates to the LDAP server with server
   ID Gateway1 on server1.us.ibm.com:389 and to the LDAP server with server ID Replica on
3. Consumer(s): LDAP server with server ID Gateway2 on server2.us.ibm.com:389 consumes updates
   from LDAP server with server ID Gateway1 on server1.us.ibm.com:389. LDAP server with Server ID
   Gateway1 on server1.us.ibm.com:389 consumes updates from LDAP server with server ID Gateway2
   consumes updates from LDAP server with server ID Gateway2 on server2.us.ibm.com:389.
4. Read-write server(s): LDAP servers Gateway1 and Gateway2 on server1.us.ibm.com and
   server2.us.ibm.com are read-write servers.
5. Read-only server(s): LDAP server Replica on server3.us.ibm.com is a read-only server.
Configuration changes: Some configuration changes need to be made to the Gateway1, Gateway2, and
Replica servers for replication to work correctly. Configured servers should be current. If reusing the
same servers that were used for the Master-Replica setup, undo the changes that were made in
Creating a master-replica topology. If reusing the same servers that were used for the Peer-Peer setup, undo the changes that were made in Creating a peer-to-peer replication topology. If reusing the same servers that were used for the Master-forwarder-replica setup, undo the changes that were made in Creating a master-forwarder-replica (cascading) topology.

Note: These are done here for the purpose of this example ONLY. The server ID is only allowed to be modified when there are no replica subentries defined in the server. See Table 10 for more information about the \texttt{ibm-slapdServerID} attribute value.

1. **Server IDs:**
   - On the Gateway1 server (server1.us.ibm.com), apply the following modify using the \texttt{ldapmodify} utility. See \textit{IBM Tivoli Directory Server Client Programming for z/OS} for additional information about the \texttt{ldapmodify} utility.
     
     \begin{verbatim}
     dn: cn=configuration
     changetype: modify
     replace: ibm-slapdserverid
     ibm-slapdserverid: Gateway1
     \end{verbatim}
   
   - On the Gateway2 server (server2.us.ibm.com), apply the following modify using the \texttt{ldapmodify} utility.
     
     \begin{verbatim}
     dn: cn=configuration
     changetype: modify
     replace: ibm-slapdserverid
     ibm-slapdserverid: Gateway2
     \end{verbatim}
   
   - On the Replica server (server3.us.ibm.com), apply the following modify using the \texttt{ldapmodify} utility.
     
     \begin{verbatim}
     dn: cn=configuration
     changetype: modify
     replace: ibm-slapdserverid
     ibm-slapdserverid: Replica
     \end{verbatim}

2. **Consumer server credentials entry:** Add the first entry to the Gateway1 server, the second entry to the Gateway2 server, and the third entry to the Replica server using the \texttt{ldapadd} utility. See \textit{IBM Tivoli Directory Server Client Programming for z/OS} for additional information about the \texttt{ldapadd} utility.

   \begin{verbatim}
   Note: An alternative is to add entries with an \texttt{ibm-slapdSupplier} objectclass.
   \end{verbatim}

   For example, for a consumer server credentials entry for Gateway1:
   
   \begin{verbatim}
   dn: cn=Master server,cn=configuration
   changetype: add
   cn: master server
   ibm-slapdMasterDN: cn=bindtoconsumer
   ibm-slapdMasterPW:iamsupplier
   objectclass: ibm-slapdReplication
   \end{verbatim}

   For example, for a consumer server credentials entry for Gateway2:
   
   \begin{verbatim}
   dn: cn=Master server,cn=configuration
   changetype: add
   cn: master server
   ibm-slapdMasterDN: cn=bindtoconsumer
   ibm-slapdMasterPW:iamsupplier
   objectclass: ibm-slapdReplication
   \end{verbatim}

   For example, for a consumer server credentials entry for Replica:
   
   \begin{verbatim}
   dn: cn=Master server,cn=configuration
   changetype: add
   cn: master server
   \end{verbatim}
The next step is to build the LDIF file for the replication topology. This LDIF file is called gateway.ldif.

Copy each of these entries to gateway.ldif with the necessary changes in the subtree, server IDs, host names and ports.

1. **Replication context:**
   - If the subtree entry already exists, use the **ldapmodify** utility to modify the existing entry. For example,
     ```
     dn: o=ibm,c=us
     changetype: modify
     add: objectclass
     objectclass: ibm-replicationContext
     ```
   - If the subtree entry does not exist, add the entry with the **ibm-replicationContext** auxiliary objectclass by using the **ldapadd** utility. For example,
     ```
     dn: o=ibm,c=us
     changetype: add
     objectclass: top
     objectclass: organization
     objectclass: ibm-replicationContext
     o: ibm
     ```

2. **Add the replica group entry** using the **ldapadd** utility. For example:
   ```
   dn: ibm-replicaGroup=default, o=ibm,c=us
   changetype: add
   objectclass: top
   objectclass: ibm-replicaGroup
   ibm-replicaGroup: default
   ```

3. **Replica subentries**: Because there are three LDAP servers in the topology, you need to add three replica subentries, one for Gateway1, one for Gateway2, and another one for the Replica server by using the **ldapadd** utility. For example, for a subentry for Gateway1 (note that an objectclass of **ibm-replicaGateway** has been added to this entry to indicate that it is a gateway server):
   ```
   dn: ibm-replicaServerId=Gateway1,ibm-replicaGroup=default, o=ibm,c=us
   changetype: add
   objectclass: top
   objectclass: ibm-replicaSubentry
   objectclass: ibm-replicaGateway
   ibm-replicaServerId: Gateway1
   ibm-replicationServerIsMaster: true
   cn: Gateway1
   description: Subentry for Gateway1.
   ```

   For example, for a subentry for Gateway2 (note that an objectclass of **ibm-replicaGateway** has been added to this entry to indicate that it is a gateway server):
   ```
   dn: ibm-replicaServerId=Gateway2,ibm-replicaGroup=default, o=ibm,c=us
   changetype: add
   objectclass: top
   objectclass: ibm-replicaSubentry
   objectclass: ibm-replicaGateway
   ibm-replicaServerId: Gateway2
   ibm-replicationServerIsMaster: true
   cn: Gateway2
   description: Subentry for Gateway2.
   ```

   For example, for a subentry for Replica:
   ```
   dn: ibm-replicaServerId=Replica,ibm-replicaGroup=default, o=ibm,c=us
   changetype: add
   objectclass: top
   objectclass: ibm-replicaSubentry
   ```
The subentry for Gateway1 identifies the server ID as Gateway1 and has the server declared as a gateway server. The Gateway1 server can receive updates from clients. The subentry for Gateway2 identifies the server ID as Gateway2 and has the server declared as a gateway server. The Gateway2 server can receive updates from clients. The subentry for Replica has the server ID as Replica and is declared as a non-master server; it cannot get updates from clients.

4. **Supplier server credentials entry:** This step defines the credentials that Gateway1 uses to bind to Gateway2, Gateway2 uses to bind to Gateway1, and Gateway2 uses to bind to Replica. Add the entry with the `ldapadd` utility. For example,

```bash
dn: cn=ReplicaBindCredentials, o=ibm,c=us
changetype: add
objectclass: ibm-replicationCredentialsSimple
cn: ReplicaBindCredentials
replicaBindDN: cn=bindtoconsumer
replicaCredentials: iamsupplier
description: Bind Credentials used on the gateway servers.
```

5. **Replication agreements:** There are three supplier-consumer relationships in this topology. Gateway1 supplies updates made to the o=ibm,c=us subtree to Gateway2. Gateway2 supplies updates made to the o=ibm,c=us subtree to Gateway1 and to Replica. Therefore, there are three agreements:

   a. from Gateway1 to Gateway2
   b. from Gateway2 to Gateway1
   c. from Gateway2 to Replica

**Note:** The number of replication agreements is dependent upon the number of supplier-consumer relationships in the topology.

For example, a replication agreement from Gateway1 to Gateway2:

```bash
dn: cn=Gateway2,ibm-replicaServerId=Gateway1,ibm-replicaGroup=default,o=ibm,c=us
changetype: add
objectclass: top
objectclass: ibm-replicationAgreement
cn: Gateway2
ibm-replicaConsumerId: Gateway2
ibm-replicaCredentialsDN: cn=ReplicaBindCredentials, o=ibm,c=us
description: Replication agreement from Gateway1 to Gateway2.
```

For example, a replication agreement from Gateway2 to Gateway1:

```bash
dn: cn=Gateway1,ibm-replicaServerId=Gateway2,ibm-replicaGroup=default,o=ibm,c=us
changetype: add
objectclass: top
objectclass: ibm-replicationAgreement
cn: Gateway1
ibm-replicaConsumerId: Gateway1
ibm-replicaCredentialsDN: cn=ReplicaBindCredentials, o=ibm,c=us
description: Replication agreement from Gateway2 to Gateway1.
```

For example, a replication agreement from Gateway2 to Replica:

```bash
dn: cn=Replica, ibm-replicaServerId=Gateway2,ibm-replicaGroup=default,o=ibm,c=us
changetype: add
objectclass: top
objectclass: ibm-replicationAgreement
cn: Replica
ibm-replicaConsumerId: Replica
ibm-replicaCredentialsDN: cn=ReplicaBindCredentials, o=ibm,c=us
description: Replication agreement from Gateway2 to Replica.
```
The three agreements for this topology are shown above. The first one is the agreement from Gateway1 to Gateway2. It is under the Gateway1 subentry. The second one is from Gateway2 to Gateway1 and it is under the Gateway2 subentry. The third one is from Gateway2 to Replica and it is also under the Gateway2 subentry. Note the use of the same credentials entry for all three agreements. This is acceptable. The consumer server credentials entry is added to Gateway1, Gateway2, and Replica servers.

Now that the replication entries have been added, the gateway.ldif file is as follows:

```ldif
dn: o=ibm,c=us
changetype: add
objectclass: top
objectclass: organization
objectclass: ibm-replicationContext
o: ibm

dn: ibm-replicaGroup=default, o=ibm,c=us
changetype: add
objectclass: top
objectclass: ibm-replicaGroup
ibm-replicaGroup: default

dn: ibm-replicaServerId=Gateway1,ibm-replicaGroup=default, o=ibm,c=us
changetype: add
objectclass: top
objectclass: ibm-replicaSubentry
objectclass: ibm-replicaGateway
ibm-replicaServerId: Gateway1
ibm-replicationServerIsMaster: true
cn: Gateway1
description: Subentry for Gateway1.

dn: ibm-replicaServerId=Gateway2,ibm-replicaGroup=default, o=ibm,c=us
changetype: add
objectclass: top
objectclass: ibm-replicaSubentry
objectclass: ibm-replicaGateway
ibm-replicaServerId: Gateway2
ibm-replicationServerIsMaster: true
cn: Gateway2
description: Subentry for Gateway2.

dn: ibm-replicaServerId=Replica,ibm-replicaGroup=default, o=ibm,c=us
changetype: add
objectclass: top
objectclass: ibm-replicaSubentry
objectclass: ibm-replicaGateway
ibm-replicaServerId: Replica
ibm-replicationServerIsMaster: false
cn: Replica
description: Subentry for Replica.

dn: cn=replicaBindCredentials, o=ibm,c=us
changetype: add
objectclass: ibm-replicationCredentialsSimple
cn: replicaBindCredentials
replicaBindDN: cn=bindtoconsumer
replicaCredentials: iamsupplier
description: Bind Credentials used on the gateway servers.

dn: cn=Gateway2,ibm-replicaServerId=Gateway1,ibm-replicaGroup=default,o=ibm,c=us
changetype: add
objectclass: top
objectclass: ibm-replicationAgreement
cn: Gateway2
ibm-replicaConsumerId: Gateway2
ibm-replicaCredentialsDN: cn=ReplicaBindCredentials, o=ibm,c=us
```

Chapter 24. Advanced replication 419
description: Replication agreement from Gateway1 to Gateway2.
dn: cn=Gateway1,ibm-replicaServerId=Gateway2,ibm-replicaGroup=default,o=ibm,c=us
changetype: add
objectclass: top
objectclass: ibm-replicationAgreement
cn: Gateway1
ibm-replicaConsumerId: Gateway1
ibm-replicaCredentialsDN: cn=ReplicaBindCredentials, o=ibm,c=us
description: Replication agreement from Gateway2 to Gateway1.
dn: cn=Replica, ibm-replicaServerId=Gateway2,ibm-replicaGroup=default,o=ibm,c=us
changetype: add
objectclass: top
objectclass: ibm-replicationAgreement
cn: Replica
ibm-replicaConsumerId: Replica
ibm-replicaCredentialsDN: cn=ReplicaBindCredentials, o=ibm,c=us
description: Replication agreement from Gateway2 to Replica.

- Next, add the gateway.ldif file on the Gateway2 server. Perform the ldapadd command on Gateway2
  where the gateway.ldif file was created. For example,
  ```
  ldapadd -h server2.us.ibm.com -p 389 -D adminDN -w adminPW -f gateway.ldif -k -L
  ```
  The -L option on the ldapadd utility sends the Do Not Replicate control to the server that indicates not to
  replicate the topology now. The -k option sends the Server Administration control to the server so that the addition of entries continues even when the subtree becomes read-only because of a server ID mismatch.
- Next, add the replication topology to the Gateway1 and Replica servers. Use the ldapexop command. For example,
  ```
  ldapexop -h server2.us.ibm.com -p 389 -D adminDN -w adminPW -op repltopology -rc o=ibm,c=us
  ```
  This is an example of the Replication topology extended operation. It propagates the advanced replication topology to the Gateway1 and Replica servers defined under the o=ibm,c=us replication context. See ldapexop utility for more information about the ldapexop utility.
- After the ldapadd and ldapexop commands are performed successfully, the gateway topology is ready. The Gateway1 server accepts updates that go to the Gateway2 server. The Gateway2 server also accepts updates which then forwards them to the Gateway1 and Replica servers.

### Replication topology hints and tips
Below is information that might be helpful to you for replication topology.

1. When setting up the advanced replication topologies, all master servers should be in maintenance mode until all the topology entries have been loaded on all servers participating in the topology. The Do Not Replicate and Server Administration controls should also be used when adding entries for configuring advanced replication topologies. This precludes updates to master servers that would be lost if they are received before all server responsibilities (replica, master, forwarder, gateway) and relationships (replication agreement) being established.
2. All peers in a topology need to supply to every other server in the topology unless they are separated by gateways. If they are separated by gateways, all the peers under a gateway need to supply to all other servers including the gateway. This is because peers do not replicate changes supplied by other peers. That leads to peers receiving the updates they initiated.
3. All gateways in a topology need to supply to each other. There has to be at least two gateways in a topology for them to be useful.
4. Read-only servers do not accept updates that clients send. When an update is attempted against a read-only server, the referral list returned to the client is established from the following:
a. The `ibm-replicationContext` objectclass allows for an optional attribute, `ibm-replicaReferralURL`. As stated previously, the `ibm-replicationContext` auxiliary objectclass must be added to the root of the subtree. This objectclass identifies the subtrees that are replication contexts. The `ibm-replicaReferralURL` attribute can hold a space delimited list of LDAP URLs. The URLs specified appears first in the list of referrals returned to the client. See [Replication contexts](#) for more information about replication contexts.

b. The `cn=configuration` subtree in the CDBM backend allows a consumer server credentials entry with an objectclass of `ibm-slapdReplication` to be stored. If this object exists and contains a value for the `ibm-slapdMasterReferral` attribute, the value is appended to referral list set by the replication context. If the replication context does not define a referral list with the `ibm-replicaReferralURL` attribute, this is the only value sent to the client. See [Consumer server entries](#) for more information about consumer server entries.

c. If the LDAP server configuration file has a `referral` configuration option specified and there are no consumer server credentials entries in the `cn=configuration` subtree with an `ibm-slapdMasterReferral` value, the `referral` option values are appended to the referral list set by the replication context. If the replication context does not have a referral list specified with the `ibm-replicaReferralURL` attribute and the consumer server credentials entry is not providing a referral list, the `referral` option is the only value sent to the client. See [Chapter 8, “Customizing the LDAP server configuration”](#) for more information about the `referral` configuration option.

---

### Replication of schema updates

Advanced replication allows the replication of schema updates to consumer servers. Schema updates can be replicated by configuring a replication topology under the `cn=ibmpolicies` suffix in the CDBM backend. By default, schema updates are not replicated unless a replication topology is configured in the `cn=ibmpolicies` suffix.

Before configuring schema replication, verify that the schema between the servers are already synchronized by using the `ldapdiff` utility. See [ldapdiff utility](#) for more information about the `ldapdiff` utility. If using the `ldapdiff` utility for schema comparison, the `-S` option must be specified. It is also strongly suggested that the `-L` option is specified so that schema differences are stored in an output LDIF file. The `ldapdiff` utility does not automatically fix schema differences on the consumer server. The schema on the consumer server must be manually modified with the output schema LDIF file generated by the `ldapdiff` utility.

Schema replication configuration is the same as configuring advanced replication in the LDBM or TDBM backends. See [Advanced replication configuration examples](#) for information about configuring advanced replication, however, change the suffix used in those examples with `cn=ibmpolicies`. Once the advanced replication entries are properly configured in the CDBM backend, the server performs schema replication.

---

### Protecting replication topology entries

The default propagating ACLs inherited from a suffix or root entry might be inappropriate for controlling access to the replication topology entries. To protect access to all replication topology entries in the server, it is recommended that the `ibm-slapdReplRestrictedAccess` attribute value be set to true in the `cn=replication,cn=configuration` entry. When the `ibm-slapdReplRestrictedAccess` attribute is true, only the LDAP administrator and the master server DN for the replication context is allowed access to the replication topology entries. See [Table 11](#) for more information about the `ibm-slapdReplRestrictedAccess` attribute.

---

### Unconfiguring advanced replication

If advanced replication is no longer needed, perform the following steps on the supplier server to remove the replication topology entries:
1. Ensure that the supplier server is in maintenance mode. See Advanced replication maintenance mode for more information about advanced replication maintenance mode.

2. Bind to the supplier server as the LDAP administrator.

3. Delete the replication agreement entry.

4. Delete the replica subentry.

5. Delete the replica group.

6. Delete the `ibm-replicationContext` attribute value from the objectclass attribute type from the replication context.

7. Move the supplier server out of maintenance mode. See Advanced replication maintenance mode for more information about advanced replication maintenance mode.

On the consumer server, perform the following steps to remove the replication topology entries:

1. Ensure that the consumer server is in maintenance mode. See Advanced replication maintenance mode for more information about advanced replication maintenance mode.

2. Bind to the consumer server as the LDAP administrator.

3. Delete the replication agreement entry.

4. Delete the replica subentry.

5. Delete the replica group.

6. Delete the `ibm-replicationContext` attribute value from the objectclass attribute type from the replication context.

7. If the server is no longer a consumer server for any other replication contexts, delete the consumer server credentials entry with an objectclass of `ibm-slapdReplication`.

8. If there are any consumer server credentials entries with an objectclass of `ibm-slapdSupplier` that have only one `ibm-slapdReplicaSubtree` attribute value equal to the replication context that is being deleted, the entry can be deleted, or else, delete the `ibm-slapdReplicaSubtree` attribute value from the consumer server credentials entry.

9. Move the consumer server out of maintenance mode. See Advanced replication maintenance mode for more information about advanced replication maintenance mode.

At this point, advanced replication is no longer configured between these two servers and each server is distinctly managing the data in the subtree that made up the replication context.

---

**Advanced replication maintenance mode**

Maintenance mode allows the LDAP administrator to setup the server for advanced replication. This mode restricts access to all entries in an LDAP server. This allows the advanced replication topology to be fully configured on all servers participating in advanced replication.

While an LDAP server is in maintenance mode, the adminDN has unrestricted access to all entries in the server and can update operational attributes, such as `ibm-entryuuid` and `modifyTimestamp`, in the server. The operational attribute values in the `replicateOperationalAttributes` control are allowed to be updated when bound as the adminDN in maintenance mode. When the server is not in maintenance mode, the adminDN must specify the Server Administration control for the server to honor the operational attribute values in the `replicateOperationalAttributes` control. See Appendix C, “Supported server controls” for more information about the `replicateOperationalAttributes` and Server Administration controls.

Consumer servers under maintenance mode will continue to accept updates from supplier servers.

Pending replication entries are replicated to consumer servers, but, updates performed when in maintenance mode are not replicated.
Other users can bind to the LDAP server, but cannot access any entries within the server.

Specify the -m option on the server startup command to start the LDAP server in maintenance mode. The LDAP server MAINTMODE operator modify command can be used to change from maintenance mode to normal mode while the LDAP server is running. It can also be used to put a running server into maintenance mode.

The following command can be sent to the LDAP server from the SDSF or the operator’s console. If the command is entered from SDSF, it must be preceded by a slash (/).

```
f dssrv,maintmode state
```

where state can be on to turn maintenance mode on or off to turn maintenance mode off (and turn normal mode on).

---

### Partial replication

Partial replication is an advanced 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. With partial replication support, the LDAP administrator can allow entries that have certain object class values to be replicated to a consumer server. For example, entries that have an objectclass of `person` and only the `cn`, `sn`, `userPassword` attributes are allowed to be replicated but not the `description` attribute.

The replication filter entry can be specified in each individual replication agreement entry in the `ibm-replicationFilterDN` attribute value. See [Replication agreements](#) for more information. A replication filter entry has a structural objectclass of `ibm-replicationFilter`. See [Table 71](#) for the required attribute values for the `ibm-replicationFilter` objectclass. If an `ibm-replicationFilterDN` attribute value is not a valid replication filter entry or does not exist, replication for the replication agreement is suspended. The replication filter entry must be added to the directory or the `ibm-replicationFilterDN` attribute value must be removed from the replication agreement entry. When replication is suspended for the replication agreement, it can be resumed by using the Control replication extended operation in the `ldapexop` utility. See [ldapexop utility](#) for more information about the `ldapexop` utility.

The attributes that are to be replicated are specified using a replication filter in the `ibm-replicationFilterAttr` attribute. 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 are selected for replication while an exclusion list is list of attributes that are not selected for replication. See [Table 71](#) for the required attribute values for the format of the `ibm-replicationFilterAttr` attribute values.

---

**Table 71. ibm-replicationFilter objectclass schema definition (required attributes)**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description and Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>cn</code></td>
<td>Common name of the replication filter entry. This attribute does not affect advanced replication configuration. Example: cn: filter1</td>
</tr>
</tbody>
</table>
Table 71. ibm-replicationFilter objectclass schema definition (required attributes) (continued)

<table>
<thead>
<tr>
<th>Attribute description and example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ibm-replicationFilterAttr</strong></td>
</tr>
<tr>
<td>A multi-valued attribute that specifies a replication filter. A replication filter is based on the object class values of entries that are replicated. The filter can be an inclusion or exclusion filter.</td>
</tr>
<tr>
<td>The following is the replication filter format:</td>
</tr>
<tr>
<td>( (\text{objectclass}=\text{objclass}):(\text{attr1},\text{attr2}))</td>
</tr>
<tr>
<td>where,</td>
</tr>
<tr>
<td>\text{objclass} Specifies a valid objectclass in the server's schema. If an * is specified, then all other objectclasses not specified by other ibm-replicationFilterAttr attribute values (if any) in the replication filter entry, are subject to this replication filter.</td>
</tr>
<tr>
<td>! If specified, indicates that the attribute type list is an exclusion list, otherwise, it is an inclusion list.</td>
</tr>
<tr>
<td>\text{attr1, attr2} Specifies a list of valid attribute types in the server's schema. If an exclusion list, the attribute types in this list are not replicated for entries that have an objectclass value of \text{objclass}. If an inclusion list, the attribute types in this list are replicated for entries that have an objectclass value of \text{objclass}. An * can be specified to indicate all attribute types for the \text{objclass}.</td>
</tr>
<tr>
<td>The following attributes are always replicated, irrespective of their presence in the exclusion list:</td>
</tr>
<tr>
<td>• Object class attributes of an entry</td>
</tr>
<tr>
<td>• Naming attribute</td>
</tr>
<tr>
<td>• All operational attributes (for example, ibm-entryuuid attribute values)</td>
</tr>
<tr>
<td>Notes:</td>
</tr>
<tr>
<td>1. If an attribute type is present in both an inclusion and an exclusion list, the exclusion takes precedence.</td>
</tr>
<tr>
<td>2. If there is not an ibm-replicationFilterAttr attribute value with \text{objclass} equal to *, no replication with entries that have an objectclass other than the ones explicitly specified is done. This acts as if an ibm-replicationFilterAttr attribute value of ( (\text{objectclass}=*)!:!(*)) is specified on the replication filter entry.</td>
</tr>
<tr>
<td>Example:</td>
</tr>
<tr>
<td>ibm-replicationFilterAttr: (objectclass=person): (cn, sn)</td>
</tr>
</tbody>
</table>

The example in Table 71 allows the following replication to occur:

• Entries with an objectclass of **person** only have their **cn** and **sn** attribute values replicated.
• Entries with other objectclasses are not replicated.

**Replication filter examples**

The following are examples that explain the usages of replication filters:

Example 1

dn: cn=replicationfilter, cn=localhost
objectclass: ibm-replicationFilter
ibm-replicationFilterAttr: (objectclass=person):(*)
ibm-replicationFilterAttr: (objectclass=person):!(*)

The first ibm-replicationFilterAttr filter value indicates entries with an objectclass of **person** have all their attributes replicated. The second ibm-replicationFilterAttr filter value indicates entries with an objectclass other than **person** are not replicated. This means that only entries with an objectclass of **person** are replicated and no other entries are replicated.

Example 2
The first `ibm-replicationFilterAttr` filter value indicates entries with an objectclass of `javaObject` have their `javaClassName` and `description` attributes replicated. The second `ibm-replicationFilterAttr` filter value indicates entries with an objectclass of `javaNamingReference` has the `javaReferenceAddress` attribute replicated. The third `ibm-replicationFilterAttr` filter value indicates the `javaReferenceAddress` attribute is not replicated for any entries other than `javaObject` and `javaNamingReference`.

Therefore, if an entry has an objectclass of `javaObject`, the `javaClassName` and `description` attributes are replicated. If an entry has an objectclass of `javaObject` and an auxiliary objectclass of `javaNamingReference`, the `javaClassName`, `description`, and `javaReferenceAddress` attributes are replicated. If an entry has an objectclass other than `javaObject` or `javaNamingReference`, all attributes except `javaReferenceAddress` are replicated.

Example 3

The first `ibm-replicationFilterAttr` filter value indicates entries with an objectclass of `person` have their `cn`, `sn`, and `userPassword` attribute values replicated. The second `ibm-replicationFilterAttr` filter value indicates that entries with an objectclass of `inetOrgPerson` do not have their `userPassword` and `employeeNumber` attribute values replicated. The third `ibm-replicationFilterAttr` filter value indicates that no other attributes are replicated if the objectclass is something other than `person` or `inetOrgPerson`.

Therefore, if an entry has an objectclass of `person`, the attributes `cn`, `sn`, and `userPassword` are replicated. If an entry has objectclasses of `person` and `inetOrgPerson`, only the `cn` and `sn` attributes are replicated. Because the `userPassword` attribute is present in both the inclusion and exclusion list, the `userPassword` attribute is eliminated because exclusion takes precedence over inclusion. If any other entry has an objectclass other than `person` or `inetOrgPerson`, no attributes are replicated.

SSL/TLS and advanced replication

SSL/TLS can be used to communicate between a replicating server (supplier, gateway, forwarder, or peer) and a replica server (consumer, gateway, forwarder, or peer).

Replica server with SSL/TLS enablement

Set the replica server up for SSL/TLS by updating the LDAP server configuration file if it is not already configured for SSL/TLS. An LDAP URL with a prefix of `ldaps://` is required in the `listen` configuration option in the replica server so that a secure connection can be configured. See Setting up for SSL/TLS for more information.

If a SASL EXTERNAL bind is performed between the replicating and replica servers, the replica server must be configured to use server and client authentication. The `sslAuth` configuration option must be set to `serverClientAuth`. The replica server must have the replicating server's certificate in its key database file, RACF key ring, or PKCS #11 token. See Setting up for SSL/TLS for more information.

Replicating server with SSL/TLS enablement

The replicating server acts as an SSL/TLS client to the replica server. To set up the replicating server to use simple binds, you must:
1. Run the gskkyman utility or the RACDCERT command as if you were the client. For more information, see z/OS Cryptographic Services System SSL Programming for the gskkyman utility or z/OS Security Server RACF Command Reference for the RACDCERT command. The key database file, RACF key ring, or PKCS #11 token must contain the replicating server’s key pair and certificate. Receive the replica’s self-signed certificate and mark it as trusted.

2. In the LDAP server configuration file on the replicating server:
   - Set the sslKeyRingFile configuration option to the replica key database file, RACF keyring, or PKCS #11 token created above.
   - If a key database file is used, set sslKeyRingFilePW to the password for the key database file, or set sslKeyRingPWStashFile to the file name where the password is stashed.

3. The ibm- replicaURL attribute value in the replication agreement entry must use an LDAP URL with a prefix of ldaps://. This indicates that an SSL connection is used between the replicating and replica servers. See Table 62 for more information about the ibm-replicaURL attribute value.

The above procedure can also be used to set up the replicating server to use SASL EXTERNAL binds. The SSL related configuration options in the LDAP server configuration file, if specified, represent the default values for the related optional attributes in ibm-replicationCredentialsExternal objects. These defaults can be overridden by specifying the optional attributes. See Table 65 for more information about SASL EXTERNAL credentials entries.

Because the replicating server acts as an SSL/TLS client to the replica server, the replicating server binds with the replica server.

Command line tasks for managing replication

This topic discusses the use of the tools that are at your disposal to check the current advanced replication status for each of your configured replication agreements. These procedures can be used to recover from out of sync conditions between servers participating in advanced replication.

Advanced replication related extended operations

A set of extended operations is provided to allow administrators the opportunity to manage all aspects of advanced replication. Specifically, the extended operations allow an administrator to do the following:

- Propagate any replication topology entries to all consumers
- Manage the replication queue, especially in the context of replication scheduling
- Manage any replication related errors
- Manage the quiesce state of the replication context
- Suspend or resume replication processing

The extended operations also allow for an administrator to manage a specific server and then have the server cascade the management operation to any of its consumers. These extended operations require the user to bind with the LDAP server administrator’s DN and password. The ldapexop is provided to invoke the advanced replication extended operations. This utility provides a command line interface to all of the advanced replication extended operations. See ldapexop utility for more information about the ldapexop utility.

Table 72 summarizes the extended operations with their ldapexop operation value.

Note: The term, cascade, is used to describe the process of a supplier server issuing an extended operation it just processed to one or all of its consumers.
<table>
<thead>
<tr>
<th>ldapexop operation</th>
<th>ldapexop description</th>
<th>Overview</th>
</tr>
</thead>
<tbody>
<tr>
<td>cascrepl</td>
<td><strong>Cascading control replication</strong> extended operation</td>
<td>This extended operation can quiesce, unquiesce, or force immediate replication of all pending changes (even if scheduling is dictated another way). When the extended operation is performed on the supplier that this extended operation was issued against, it proceeds to cascade the extended operation to one or all of its consumers. In the case of forcing immediate replication of all pending changes, two variants of the extended operation can be issued. If the replication of all pending changes must complete before the cascade step, the administrator can use the <code>wait</code> option. If there are no dependencies on the completion of the replication operations before cascading, the <code>replnow</code> option can be used. See <a href="#">Cascading control replication</a> for more information about the <strong>Cascading control replication</strong> extended operation.</td>
</tr>
<tr>
<td>controlqueue</td>
<td><strong>Control replication queue</strong> extended operation</td>
<td>This extended operation can skip one or all pending changes in the advanced replication queue. See <a href="#">Control replication queue</a> for more information about the <strong>Control replication queue</strong> extended operation.</td>
</tr>
<tr>
<td>controirepl</td>
<td><strong>Control replication</strong> extended operation</td>
<td>This extended operation suspends or resumes all advanced replication related activity. Also, given any replication schedule objects that might exist, resuming replication will not necessarily cause the immediate replication of any pending changes. Instead, in addition to resume, this extended operation can be used to begin the immediate replication of any pending changes, even if replication schedule objects have deferred replication. See <a href="#">Control replication</a> for more information about the <strong>Control replication</strong> extended operation.</td>
</tr>
<tr>
<td>ldapexop operation</td>
<td>ldapexop description</td>
<td>Overview</td>
</tr>
<tr>
<td>--------------------</td>
<td>----------------------</td>
<td>----------</td>
</tr>
<tr>
<td>controlreplerr</td>
<td>Control replication error log extended operation</td>
<td>This extended operation can delete, retry or show, any replication operations that resulted in an unsuccessful return code returned to the supplier from the consumer. The show options returns an LDIF representation of the unsuccessful replication operation. See <a href="#">Control replication error log</a> for more information about the Control replication error log extended operation.</td>
</tr>
<tr>
<td>quiesce</td>
<td>Quiesce or unquiesce context extended operation</td>
<td>This extended operation can quiesce or unquiesce a replication context. A quiesced replication context typically cannot accept any LDAP update operations. To perform LDAP update operations on a quiesced context the update operations must be done by the administrator with the Server Administration control. See <a href="#">Quiesce or unquiesce context</a> for more information about the Quiesce or unquiesce context extended operation.</td>
</tr>
<tr>
<td>repltopology</td>
<td>Replication topology extended operation</td>
<td>This extended operation propagates advanced replication topology related entries to all consumers. It then cascades this extended operation to all the consumers. This results in a cascading of the topology related entries to all servers that participate in replication for a given replication context. See <a href="#">Replication topology</a> for more information about the Replication topology extended operation.</td>
</tr>
</tbody>
</table>

### 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.

Use the following command:

```
ldapsearch -D adminDN -w adminPW -p port -b baseDN objectclass=ibm-repl*
```

The objects returned includes the replica group itself, plus the following:

- Entries with an objectclass of `ibm-replicaSubentry` for each server that replicates data within this replication context. These entries are replica subentries that contain a server ID attribute and the role that servers plays in the replication topology. See [Replica subentries](#) for more information about replica subentries.
For each replica subentry, there is a replication agreement entry for each consumer server that receives replication updates from the server described by the replica subentry. See Replication agreements for more information about replication agreement entries.

Monitoring and diagnosing advanced replication problems

An LDAP administrator can monitor the state of advanced replication processing and troubleshoot problems by using LDAP search requests to retrieve operational attributes available for the roots of the replication contexts (entries with an objectclass of ibm-replicationContext) and replication agreements (entries with an objectclass of ibm-replicationAgreement). Because these are operational attributes, they must be specifically requested on a search request in order to be returned and cannot be specified in search filters.

The following tables describe the operational attributes for the replication context and replication agreement entries. Replication context entries use the auxiliary objectclass of ibm-replicationContext and replication agreement entries use the structural objectclass ibm-replicationAgreement. See Table 73 for the operational attributes for the ibm-replicationContext objectclass. See Table 74 for the operational attributes for the ibm-replicationAgreement objectclass.

When retrieved for a replication context or replication agreement entry, the operational attributes provide information concerning that entry. It is very important to take notice of attributes that have values that contain failureId or changeId values. These IDs are often required when working with the Control replication error log and the Control replication queue extended operations with the ldapexop utility. See ldapexop utility for more information about the ldapexop utility.

Table 73. ibm-replicationContext operational attributes

<table>
<thead>
<tr>
<th>Attribute and description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ibm-replicationThisServerIsMaster</td>
</tr>
<tr>
<td>A boolean (true or false) indicating whether the server is the master of the replication context. If set to true, the server is the master of the replication context. If set to false, the server is not the master of the replication context.</td>
</tr>
<tr>
<td>ibm-replicationsQuiesced</td>
</tr>
<tr>
<td>A boolean (true or false) indicating whether the replication context is quiesced. If set to true, the replication context is quiesced. If set to false, the replication context is not quiesced.</td>
</tr>
</tbody>
</table>

Updates under a quiesced replication context are restricted to the LDAP administrator, if using the Server Administration control (OID 1.3.18.0.2.10.15), and any replication master DNs with authority under this context. Advanced replication continues for a quiesced context. If the server is restarted, all replication contexts are then unquiesced.

See Table 59 for the optional non-operational attribute for the ibm-replicationContext objectclass.

Table 74. ibm-replicationAgreement operational attributes

<table>
<thead>
<tr>
<th>Attribute and description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ibm-replicationChangeLDIF</td>
</tr>
<tr>
<td>The LDIF representation of the next pending change that has not yet been replicated and has resulted in advanced replication being stalled to the consumer server. If there is not a stalled replication change, the value is N/A.</td>
</tr>
</tbody>
</table>

Examples of when an advanced replication queue may be stalled include:

1. A replication change failed because of an LDAP_TIMEOUT return code.
2. The backend replication table has reached the maximum number of errors allowed on the supplier server within this backend while attempting to replicate a change to a consumer server. See Table 11 for more information about the ibm-slapdReplMaxErrors attribute value.
Table 74.  **ibm-replicationAgreement** operational attributes (continued)

<table>
<thead>
<tr>
<th>Attribute and description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ibm-replicationFailedChangeCount</strong></td>
</tr>
<tr>
<td>Specifies the number of advanced replication operations that have failed in this replication agreement. This number is shared among all replication agreement entries on the backend level by the <code>ibm-slapdReplMaxErrors</code> attribute in the CDBM backend configuration entry <code>cn=Replication, cn=Configuration</code>. See Table 11 for more information about the <code>ibm-slapdReplMaxErrors</code> attribute value.</td>
</tr>
</tbody>
</table>

| **ibm-replicationFailedChanges** |
| A multi-valued attribute that lists all of the logged replication operations that have failed. The number of attribute values is shared among all replication agreement entries on the backend level by the `ibm-slapdReplMaxErrors` attribute in the CDBM backend configuration entry `cn=Replication, cn=Configuration`. See Table 11 for more information about the `ibm-slapdReplMaxErrors` attribute value. |
| A string value of the form: `failureId timestamp returnCode numOfAttempts changeld operation entryDn` |
| The `failureId` identifies the update that has failed to replicate to the consumer server. The `failureId` is used with the Control replication error log extended operation to display, delete, or retry the failing replication update. The `ldapexop` utility supports the `Control replication error log` extended operation. See `ldapexop` utility for more information about the `ldapexop` utility. |
| The `timestamp` is the time in Zulu format when this operation was last attempted to be replicated to the consumer server. |
| The `returnCode` is the LDAP return code from the consumer server. |
| The `numOfAttempts` is the number of times the error has been retried on the consumer server. |
| The `changeld` is the ID that this `failureId` had when it was in the pending replication queue. |
| Example: |

```none
ibm-replicationfailedchanges: 1 20050407202221Z 68 1 170814 add cn=entry-85,o=IBM,c=US
```
| `failureId`: 1 |
| `timestamp`: 20050407202221Z |
| `returnCode`: 68 |
| `numOfAttempts`: 1 |
| `changeld`: 170814 |
| `operation`: add |
| `entryDn`: `cn=entry-85,o=IBM,c=US` |

| **ibm-replicationLastActivationTime** |
| Specifies the Zulu format timestamp when advanced replication actively began replicating queued updates. |

| **ibm-replicationLastChangeID** |
| Specifies the replication change ID of the last successfully completed advanced replication update. |

<p>| <strong>ibm-replicationLastFinishTime</strong> |
| Specifies the Zulu format timestamp when advanced replication updates in the queue were all attempted and the server awaits a new scheduled start time or more operations to appear in the advanced replication queue. See Schedule entries for more information about replication schedule entries. |</p>
<table>
<thead>
<tr>
<th>Attribute and description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ibm-replicationLastResult</strong></td>
</tr>
<tr>
<td>A description of the result from the last advanced replication operation or connection attempt to a consumer server.</td>
</tr>
<tr>
<td>A string value of the form: timestamp changeld returnCode operation entryDn</td>
</tr>
<tr>
<td>The timestamp is the time in Zulu format when this operation was last attempted to be replicated to the consumer server.</td>
</tr>
<tr>
<td>The changeld is the ID of the last replication update.</td>
</tr>
<tr>
<td>The returnCode is the LDAP return code from the consumer server.</td>
</tr>
<tr>
<td>The operation indicates the last LDAP operation. It has one of the following values: add, connect, delete, modify, or modifydn</td>
</tr>
<tr>
<td>The entryDn indicates the distinguished name of the entry that was last added, deleted, modified, or renamed. If operation is connect, entryDn is set to NULL.</td>
</tr>
<tr>
<td>Example:</td>
</tr>
<tr>
<td>ibm-replicationLastResult: 20050412140436Z 19 81 add cn=testpendingchange,o=ibm,c=us</td>
</tr>
<tr>
<td>timestamp: 20050412140436Z</td>
</tr>
<tr>
<td>changeld: 19</td>
</tr>
<tr>
<td>returnCode: 81</td>
</tr>
<tr>
<td>operation: add</td>
</tr>
<tr>
<td>entryDn: cn=testpendingchange,o=ibm,c=us</td>
</tr>
<tr>
<td><strong>ibm-replicationLastResultAdditional</strong></td>
</tr>
<tr>
<td>The descriptive reason code message text that supplements the return code message with the purpose of providing additional information from the last replication attempt.</td>
</tr>
<tr>
<td><strong>ibm-replicationNextTime</strong></td>
</tr>
<tr>
<td>Specifies the Zulu format timestamp of the next time advanced replication would begin if pending changes existed. When this value is set to 19000101000000z, replication begins immediately once a change is ready to be replicated if the ibm-replicationState operational attribute is set to active.</td>
</tr>
<tr>
<td><strong>ibm-replicationPendingChangeCount</strong></td>
</tr>
<tr>
<td>The number of replication operations that are waiting to be replicated to a consumer server.</td>
</tr>
<tr>
<td><strong>ibm-replicationPendingChanges</strong></td>
</tr>
<tr>
<td>A multi-valued attribute that lists all changes waiting to be replication to a consumer server.</td>
</tr>
<tr>
<td>A string value of the form: changeld operation entryDn</td>
</tr>
<tr>
<td>The changeld is the ID of the pending replication update.</td>
</tr>
<tr>
<td>The operation indicates the LDAP operation that is pending. It has one of the following values: add, delete, modify, or modifydn</td>
</tr>
<tr>
<td>The entryDn indicates the distinguished name of the entry that is to be added, deleted, modified, or renamed.</td>
</tr>
<tr>
<td>Example:</td>
</tr>
<tr>
<td>ibm-replicationpendingchanges: 19 add cn=test1,o=ibm,c=us</td>
</tr>
<tr>
<td>changeld: 19</td>
</tr>
<tr>
<td>operation: add</td>
</tr>
<tr>
<td>entryDn: cn=test1,o=ibm,c=us</td>
</tr>
</tbody>
</table>
Table 74. `ibm-replicationAgreement` operational attributes (continued)

<table>
<thead>
<tr>
<th>Attribute and description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ibm-replicationState</code></td>
</tr>
<tr>
<td>Identifies the current state of the advanced replication queue. It has one of the following values:</td>
</tr>
<tr>
<td>• <strong>active</strong> - Indicates that advanced replication is occurring from this replication agreement.</td>
</tr>
<tr>
<td>• <strong>binding</strong> - Indicates that the replication agreement is in the process of authenticating with the consumer server.</td>
</tr>
<tr>
<td>• <strong>connecting</strong> - Indicates that the replication agreement is attempting to contact the consumer server.</td>
</tr>
<tr>
<td>• <strong>on hold</strong> - Indicates that the replication agreement is on hold. Replication updates to the consumer server are queued until the replication agreement is resumed.</td>
</tr>
<tr>
<td>• <strong>ready</strong> - Indicates immediate replication mode, ready to send updates as they occur.</td>
</tr>
<tr>
<td>• <strong>retrying</strong> - Indicates that the server retries the current change every 60 seconds until it succeeds. The retrying state occurs when a consumer server is restarted, the replication backend table is full, the current replicated update is failing, or when there is an <code>LDAP_TIMEOUT</code> return code from the consumer server. Retrying is a likely symptom that advanced replication may be stalled and LDAP administrator intervention is required to get it running again. See <a href="#">Recovering from advanced replication errors</a> for the steps on how to recover from out of sync conditions between supplier and consumer servers.</td>
</tr>
<tr>
<td>• <strong>suspended</strong> - Indicates that the replication agreement is suspended. No additional replication updates are sent to the consumer server by this agreement (until it returns to the ready state).</td>
</tr>
<tr>
<td>• <strong>waiting</strong> - Indicates that the replication agreement is currently waiting for the next scheduled replication to occur. See <a href="#">Schedule entries</a> for more information about replication schedule entries.</td>
</tr>
</tbody>
</table>

See Table 62 for the required non-operational attributes for the `ibm-replicationAgreement` object class. See Table 63 for the optional non-operational attributes for the `ibm-replicationAgreement` object class.

Recovering from advanced replication errors

Replication errors can be handled proactively, before they are allowed to accumulate, or reactively, after replication has already stalled. Replication stalls occur when the number of failures reaches the limit as specified by the `ibm-slapdReplMaxErrors` attribute value in the `cn=Replication,cn=configuration` entry. See Table 11 for more information about the `cn=Replication,cn=configuration` entry.

When replication is stalled, the latest failed change occupies the beginning of the pending changes queue. The latest failed change gets retried every minute until it succeeds or the failed change is removed from the queue by the LDAP administrator. When this failed change occupies the lead position in the pending replication queue, all other replication updates are blocked and replication is stalled.

The options for handling stalled replication are:

1. Increase the size of the `ibm-slapdReplMaxErrors` attribute in the `cn=Replication,cn=configuration` entry. This allows more replication failures to be stored in the backend where the replication agreement entry resides.
2. Delete or retry one or more failed replication changes.
3. Skip the latest failed replication change.
4. If the stalled replication problem is severe enough, the entire replication context where the replication agreement entry resides may need to be re-synchronized. In order to do this, you must:
   a. Quiesce the replication context
   b. Suspend replication for all replication agreements
   c. Delete all failed replication changes for all replication agreements
   d. Skip all pending changes for all replication agreements
   e. Re-synchronize the replication context
   f. Resume replication for the suspended replication agreements
   g. Unquiesce the replication context
The following operational attributes in the replication agreement entry can be queried to determine what to do:

1. The `ibm-replicationChangeLdif` operational attribute in the replication agreement entry shows the LDIF representation of the latest failure. The `ibm-replicationLastResult` and `ibm-replicationLastResultAdditional` operational attributes in the replication agreement have further detail for the reason the change failed.

2. The `ibm-replicationPendingChanges` operational attribute in the replication agreement shows the change ID, the operation type, and the target DN of the next changes to be replicated. The number of pending changes that are displayed is limited by the `ibm-slapdMaxPendingChangesDisplayed` attribute in the `cn=Replication,cn=configuration` entry. See Table 11 for more information about the `ibm-slapdMaxPendingChangesDisplayed` attribute. See Table 74 for more information about the `ibm-replicationPendingChanges` operational attribute.

3. The `ibm-replicationFailedChanges` operational attribute in the replication agreement shows each of the failed changes, including the failure ID. See Table 74 for more information about the `ibm-replicationFailedChanges` operational attribute.

4. The Control replication error log extended operation can be used to display information about a failure by providing the `failureId` obtained from the `ibm-replicationFailedChanges` operational attribute. The `controlreplerr` extended operation `-show` option in the `ldapexop` utility can be used to display the latest failure. See `ldapexop` utility for more information about the `ldapexop` utility.

Once the latest and all previous failures are understood, the LDAP administrator must decide whether to fix the replication failures individually or re-synchronize the entire replication context. The options are:

1. Increase the size of the `ibm-slapdReplMaxErrors` attribute in the `cn=Replication,cn=configuration` entry. This allows more replication failures to be stored in the backend where the replication agreement entry resides. See Table 11 for more information about the `ibm-slapdReplMaxErrors` attribute.

2. Delete or retry one or more failed changes for the replication agreement by using the Control replication error log extended operation with the `ldapexop` utility. The `-retry` option on the `controlreplerr` extended operation in the `ldapexop` utility allows a single failure (identified by its `failureId`) to be retried or all failures to be retried. The ability to retry all failures is especially useful when you have corrected the problem that caused a change to fail the first time. Once a failed change is retried successfully, it is removed from the list of failed changes and there is space for a new one. The `-delete` option on the `controlreplerr` extended operation in the `ldapexop` utility allows a single failure (identified by its `failureId`) to be deleted or all failures to be deleted. This delete option is especially useful when a change is deemed to be unnecessary, the problem has been fixed manually, or a synchronization tool such as the `ldiff` utility has been used to resynchronize the directories. Deleting a failed change frees space in the list of failed changes so that a new failure can be added. See `ldapexop` utility for more information about the `ldapexop` utility. See `ldiff` utility for more information about the `ldiff` utility.

3. Skip the latest failure for the replication agreement by using the Control replication queue extended operation. The `ldapexop` utility supports the Control replication queue extended operation that allows the next pending change (identified by its `changeId`) or all pending changes to be skipped. This extended operation is useful when the `ibm-slapdReplMaxErrors` attribute in the `cn=Replication,cn=configuration` entry is set to 0 in which case the replication failure is not allowed and replication stalls on the first failure. Also, the Control replication queue extended operation is useful when replication failures should not be deleted, the `ibm-slapdReplMaxErrors` attribute value should not be increased, or after using the `ldiff` utility to re-synchronize the replication context. See `ldapexop` utility for more information about the `ldapexop` utility.

4. If there are multiple failed and pending replication changes, the entire replication context where the replication agreement entry resides may need to be re-synchronized. In order to do this, you must:
   a. Quiesce the replication context on all servers in the replication topology by using the Cascading control replication extended operation on the `ldapexop` utility. The Cascading control replication extended operation should be targeted against the master server which in turn quiesces the replication context on all consumer servers. A quiesced replication context only accepts updates from the LDAP administrator when using the `Server Administration` control and...
replica master servers. See Cascading control replication extended operation. See ldapexop utility for more information about the ldapexop utility.

b. Suspend replication for all replication agreements in the replication context by using the Control replication extended operation on the ldapexop utility. A suspended replication agreement queues all replication changes updates until it is resumed. See Control replication extended operation for more information about the Control replication extended operation.

c. Use the ldapiutil utility with the -L option to compare the replication contexts on each of the servers within the replication context. The -L option allows the entry differences to be written to an output LDIF file. See ldapiutil utility for more information about the ldapiutil utility.

d. Delete all failed replication changes for all replication agreements by using the Control replication error log extended operation on the ldapexop utility. See Control replication error log extended operation for more information about the Control replication error log extended operation.

e. Skip all pending replication changes by using the Control replication queue extended operation on the ldapexop utility. See Control replication queue extended operation for more information about the Control replication queue extended operation.

f. Re-synchronize the replication context by using the fix option on the ldapiutil utility.

g. Resume replication for all suspended replication agreements by using the Control replication extended operation on the ldapexop utility.

h. Unquiesce the replication context on all servers in the replication topology by using the Cascading control replication extended operation on the ldapexop utility.

The other methodology for handling replication failures is to take a proactive, preventive approach. The LDAP administrator monitors the replication failure queue and resolves problems before the queue reaches capacity and replication stalls. The LDAP administrator can use the Control replication error log extended operation and the ibm-replicationFailedChanges and ibm-replicationState operational attributes in the replication agreement entry to monitor the current replication status.

**Advanced replication error recovery example**

This advanced replication error recovery example uses the master-replica topology that has been configured in Creating a master-replica topology. This example assumes the ibm-slapdReplMaxErrors attribute value in the cn=Replication,cn=configuration entry is set to one.

The LDAP administrator periodically monitors the replication status of the replication agreement in the o=ibm,c=us replication context by querying the replication agreement operational attribute values. See Table 74 for more information about the replication agreement operational attributes.

**Note:** Operational attributes must be specified so that they are returned on a search operation.

The current replication status from the master to the replica can be determined by using the following ldapsearch command to retrieve the replication agreement entry. See IBM Tivoli Directory Server Client Programming for z/OS for more information about the ldapsearch utility.

```
```

The ldapsearch command returns the following entry:

```
 cn=Replica, ibm-replicaServerId=Master, ibm-replicaGroup=default, o=ibm, c=us
 objectclass=top
 objectclass=ibm-replicationAgreement
```
The following analysis of the replication agreement entry can be performed:

1. The **ibm-replicationState** operational attribute value is set to *retrying* which indicates replication is currently stalled. Replication is stalled because the number of replication failures exceeds one. (The **ibm-slapdMaxReplErrors** attribute value has been set to one in the **cn=Replication,cn=configuration** entry).

2. The **ibm-replicationChangeLdif** operational attribute in the replication agreement shows the LDIF representation of the latest failure. The LDIF shows the last failure is a modify of the **ou=sub,o=ibm,c=us** entry on the consumer server. The **ibm-replicationLastResult** and **ibm-replicationLastResultAdditional** operational attributes in the replication agreement indicate that the modify failed on the consumer server because the **ou=sub,o=ibm,c=us** entry does not exist.

3. The **ibm-replicationPendingChanges** operational attribute in the replication agreement shows the **changeId** of the next pending update is 46. The next pending change is also the same modify operation of the **ou=sub,o=ibm,c=us** entry. It will be replicated to the consumer server after the add failure in the **ibm-replicationFailedChanges** operational attribute is resolved.

4. The **ibm-replicationFailedChanges** operational attribute in the replication agreement shows one failed replication update. The attribute value indicates that the **failureId** is 12, the LDAP return code from the consumer server is 32, it is an add operation of the **cn=entry,ou=sub,o=ibm,c=us** entry, and the supplier server has tried once to replicate the update. To determine why the add of the **cn=entry,ou=sub,o=ibm,c=us** entry failed, the *ldapexop* utility can be used to perform a **Control replication error log** extended operation to show the failed replication change. See [ldapexop utility](#) for more information about the *ldapexop* utility.

The following *ldapexop* command can be used to show the LDIF representation of failed replication change that has a **failureId** of 12.

```
ldapexop -p 389 -h server1.us.ibm.com -D adminDn -w adminPw -op controlreplerr -ra "cn=Replica, ibm-replicaServerId=Master, ibm-replicaGroup=default, o=ibm,c=us" -show 12
```

The **ldapexop** command returns the following:
The LDAP administrator can either fix the replication differences manually or use the `ldapdiff` utility to re-synchronize the replication contexts on all servers in the replication topology. The `ldapdiff` utility is a useful tool for comparing and verifying that the entries within a replication context on supplier and consumer server are synchronized. For the purposes of this example, the LDAP administrator has chosen to re-synchronize the replication context by using the `ldapdiff` utility. See `ldapdiff utility` for more information about the `ldapdiff` utility.

Before you use the `ldapdiff` utility to compare or fix entries within a replication context, quiesce the replication context on all servers within the replication topology by using the `Cascading control replication` extended operation quiesce option on the `ldapexop` utility. See `ldapexop utility` for more information about the `ldapexop` utility.

The following `ldapexop` command quiesces the `o=ibm,c=us` replication context on the master and replica server in the replication topology:

```
ldapexop -p 389 -h server1.us.ibm.com -D adminDn -w adminPw -op cascrepl -action quiesce -rc "o=ibm,c=us"
```

After the replication context is quiesced on all servers, the `Control replication` extended operation can be used to suspend replication for all replication agreements within the replication context.

The following `ldapexop` command suspends replication for all replication agreements in the replication context `o=ibm,c=us`. The `cn=Replica, ibm-replicaServerId=Master, ibm-replicaGroup=default, o=ibm, c=us` is the only replication agreement within the `o=ibm,c=us` replication context so that it is the only agreement that is suspended.

```
ldapexop -p 389 -h server1.us.ibm.com -D adminDn -w adminPw -op controlrep1 -action suspend -rc "o=ibm,c=us"
```

The following `ldapdiff` command is run to compare the entries within the replication context `o=ibm,c=us` on the master server on `server1.us.ibm.com` and the replica server on `server2.us.ibm.com`. If there are any differences between the two servers, they are written to the output LDIF file called `differences.ldif`. The `ldapdiff -a` option is specified to write the `Server Administration` control to the output LDIF for each entry that is different between the two servers. See `Server Administration` for more information about the `Server Administration` control.

```
ldapdiff -a -b "o=ibm,c=us" -L differences.ldif -sh server1.us.ibm.com -sp 389 -sD adminDn -sw adminPw -ch server2.us.ibm.com -cp 389 -cD adminDn -cw adminPw
```

where `differences.ldif` contains:

```
[PLAIN]

dn: ou=sub,o=ibm,c=us
control: 1.3.18.0.2.10.15 true
control: 1.3.18.0.2.10.19 false::

MIgMBAAKAwgQMoZ3J1XVrcnNOYW11M0eE2CNgwJ1uYMDAQA2WkQP
Y3J1XVrxG1tZVNOYW1wMRgEFj1wMDkMwMjA2MTQ0D0Li4n4MgFvIAoB
ADAbA1tb2RzZmllcNOYW11M0eE2CNgwJ1uYMDAQA2WkQP
VG1tZVNOYW1wMRgEFj1wMDkMwMjA2MTQ0D0Li4n4MgFvIAoB

changeType: add
```
The contents of the differences.ldif file indicates that the ou=sub,o=ibm,c=us entry does not exist on the consumer server. This explains why the add of the child entry cn=entry,ou=sub,o=ibm,c=us failed on the consumer server.

Before synchronizing entries within a replication context on the master and replica servers, all replication failures should be deleted and all pending replication changes should be skipped. Replication failures are deleted by using the Control replication error log extended operation on the ldapexop utility. Pending replication changes are skipped by using the Control replication queue extended operation on the ldapexop utility.

The following ldapexop command deletes all failed replication failures from the backend replication table,

```
ldapexop -p 389 –h server1.us.ibm.com -D adminDn -w adminPw -op controlreplerr -delete all -ra "cn=Replica, ibm-replicaServerId=Master, ibm-replicaGroup=default, o=ibm, c=us"
```

The following ldapexop command skips (deletes) all pending replication changes from the replication queue:

```
ldapexop -p 389 –h server1.us.ibm.com -D adminDn -w adminPw -op controlqueue –skip all -ra "cn=Replica, ibm-replicaServerId=Master, ibm-replicaGroup=default, o=ibm, c=us"
```

To synchronize the o=ibm,c=us replication context on the master and replica servers, run the ldapdiff utility again with the -F (Fix) option specified or use the ldapmodify command to add the entries in the differences.ldif file to the consumer server.

Because the master and replica servers are now synchronized, the replication agreement can now be resumed and the replication context unquiesced. The replication agreement is resumed by using the Control replication extended operation on the ldapexop utility. The replication context is unquiesced on all servers in the replication topology by using the Cascading control replication extended operation on the ldapexop utility.

The following ldapexop command resumes replication for the replication agreement, cn=Replica, ibm-replicaServerId=Master, ibm-replicaGroup=default, o=ibm, c=us:

```
ldapexop -p 389 –h server1.us.ibm.com -D adminDn -w adminPw -op controlrepl –action resume -ra "cn=Replica, ibm-replicaServerId=Master, ibm-replicaGroup=default, o=ibm, c=us"
```

The following ldapexop command unquiesces the replication context o=ibm,c=us on all servers in the replication topology:
The current replication status from the master to the replica can be determined by using the following `ldapsearch` command to retrieve the replication agreement entry:

```bash
```

The `ldapsearch` command returns the following entry:

```
cn=Replica, ibm-replicaServerId=Master, ibm-replicaGroup=default, o=ibm, c=us
objectclass=top
objectclass=ibm-replicationAgreement
ibm-replicaconsumerid=Replica
ibm-replicaurl=ldap://server2.us.ibm.com:389
ibm-replicacredentialsdn=cn=ReplicaBindCredentials,o=ibm, c=us
description=Replication agreement from master to replica
cn=Replica
ibm-replicationonhold=FALSE
ibm-replicationstate=ready
ibm-replicationpendingchangecount=0
ibm-replicationnexttime=19000101000000
ibm-replicationlastfinishtime=20090206165454Z
ibm-replicationlastchangeid=46
ibm-replicationlastactivationtime=20090206144354Z
ibm-replicationfailedchangecount=0
ibm-replicationchangeid=N/A
```

Because the `ibm-replicationState` operational attribute value in the replication agreement entry is set to `ready`, replication from the master to the replica is now no longer stalled.
Chapter 25. Alias

Alias support provides a means for a TDBM, LDBM, or CDBM directory entry to point to another entry in the same directory. An alias entry can also be used to create a convenient public name for an entry or subtree, hiding the more complex actual name of the entry or subtree.

Alias support involves:

- Creating an alias entry which points to another entry
- Dereferencing during search: when a distinguished name contains an alias, the alias is replaced by the value it points to and search continues using the new distinguished name.

For example, you can create an alias entry to provide a simple name for the z/OS LDAP department:

"ou=LDAPZOS,o=IBM"

The alias entry points to the actual z/OS LDAP department:

"ou=DEPTC8NG,ou=Poughkeepsie,o=IBM_US,o=IBM"

This provides easier access to the entries of the z/OS LDAP developers, using public names such as:

"cn=kmorg,ou=LDAPZOS,o=IBM"

This name is dereferenced during search to:

"cn=kmorg,ou=DeptC8NG,ou=Poughkeepsie,o=IBM_US,o=IBM"

and the information for that entry is returned.

Impact of aliasing on search performance

Usage of aliases in a directory can cause a large increase in the amount of processing that takes place during search, even if no alias entries are actually involved in the particular search that was requested. To minimize the impact to search performance:

- Do not add aliases to the directory if they are not needed. There is no impact on search if there are no aliases in the directory.
- Only perform a search with dereferencing when aliases are involved in the search. Again, the impact on search is avoided if no dereferencing is requested.

Note: The search request from the LDAP client specifies whether to do dereferencing. The default value for dereferencing varies between different LDAP clients. If the default is to do dereferencing (this is the case with some Java clients), make sure to specifically reset this value to do no dereferencing when you issue search requests for which you do not want to do dereferencing.

- If you do want to use aliases in a directory, use them efficiently to minimize the number of alias entries. For example, use an alias entry for the root of a subtree (such as the alias for a department entry in the example above) rather than creating an alias entry for each individual entry within the subtree.

Alias entry

An alias entry contains:

- one of two object classes:
  - aliasObject - AUXILIARY object class
  - alias - STRUCTURAL object class
Note: This requires an object class such as extensibleObject to allow the naming attributes for the entry.

- aliasedObjectName attribute
  - its value is the distinguished name that the alias points to

These object classes and attributes are always part of the LDAP server schema.

Below is an example of an alias entry:

dn: ou=LDAPZOS,o=IBM
objectclass: organizationalUnit
objectclass: aliasObject
ou: LDAPZOS
aliasedobjectname: ou=DeptC8NG,ou=Poughkeepsie,o=IBM_US,o=IBM

or

dn: ou=LDAPZOS,o=IBM
objectclass: alias
objectclass: extensibleobject
ou: LDAPZOS
aliasedobjectname: ou=DeptC8NG,ou=Poughkeepsie,o=IBM_US,o=IBM

Alias entry rules

An alias is a directory entry containing either the alias structural object class or the aliasObject auxiliary object class. Both of these object classes require the aliasedObjectName attribute (the aliasedEntryName alternate name can also be used). The extensibleObject object class should also be specified if the alias object class is used in order to add the RDN attributes for the alias entry.

An alias entry must be a leaf entry. This means that no ancestor of an entry can be an alias entry. In addition, an alias entry cannot also be a referral entry. A suffix entry can be an alias entry. In this case, the suffix will have no entries below it.

The value of the aliasedObjectName attribute does not have to be an existing entry. However, an error will be returned when dereferencing the alias if the value of the aliasedObjectName attribute does not refer to an entry in the same backend as the alias entry. The value cannot be the distinguished name of the alias entry; in other words, an alias entry cannot dereference to itself.

Dereferencing an alias

All or part of a distinguished name (DN) can be an alias. Dereferencing a DN consists of the systematic replacement of an alias within the DN by the value of the aliasedObjectName attribute of the alias. This creates a new DN that must then be checked to see if it contains an alias that needs to be dereferenced. This process continues until the final dereferenced DN contains no alias within its name. An error will be returned if a circular chain is detected, that is, when a particular alias entry is encountered more than once. The final dereferenced DN must be the DN of an entry in the same backend as the original DN. This entry must either exist or be under a referral entry.

Alias dereferencing is performed only during search operations. Alias entries are not dereferenced for any other LDAP operation.

- Aliases are not dereferenced when performing a null-based subtree search since all entries in all LDBM, TDBM, and CDBM backends are included in the search scope.

Duplicate objects will not be returned by a search operation. Duplicate objects can be encountered during a search if an alias points to an entry higher in the tree or if two aliases point to the same entry.
Dereferencing is only used to determine the entries that will be included in the search. The entries actually returned as search results must match the search filter. The DN of returned entries is the dereferenced DN. Using the above example, a search for "cn=John Doe, ou=LDAPZOS,o=IBM" will return an entry with DN "cn=John Doe,ou=DeptC8NG,ou=Poughkeepsie,o=IBM,c=US" if the "cn=John Doe,ou=DeptC8NG,ou=Poughkeepsie,o=IBM,c=US" entry matches the search filter.

Access checking is not performed when dereferencing an alias entry. Normal access checking will be performed for the dereferenced entry. Therefore, a search can dereference aliases even though the requester might not have any permissions to those alias entries.

**Dereferencing during search**

**Dereference options**

A flag value controls what alias dereferencing will be done during a search operation. This flag is sent by the client on the search request. The flag can have one of four values:

**LDAP_DEREF_NEVER (0)**
- do not dereference any alias entries. Alias entries encountered during the search operation are processed as 'normal' entries and are returned if they match the search filter.

**LDAP_DEREF_SEARCHING (1)**
- dereference alias entries within the scope of the search but do not dereference the search base entry (if it contains an alias). The search base is processed as a 'normal' entry (even if it is an alias entry) and is returned if it matches the search filter and is in the search scope.

**LDAP_DEREF_FINDING (2)**
- dereference the search base entry (if it contains an alias) but do not dereference any other alias entries within the search scope. Alias entries within the search scope of the dereferenced base are processed as 'normal' entries and are returned if they match the search filter.

**LDAP_DEREF_ALWAYS (3)**
- dereference the search base entry (if it contains an alias) and dereference alias entries within the scope of the search. All alias entries encountered during the search operation are dereferenced.

**Dereferencing during finding the search base**

In a search request with **LDAP_DEREF_FINDING** or **LDAP_DEREF_ALWAYS**, dereferencing the search base just establishes a new search base. The results are equivalent to those from a search request that specifies the new base is its base.

**Dereferencing during searching in subtree searches**

In a search request with **LDAP_DEREF_SEARCHING** or **LDAP_DEREF_ALWAYS** and subtree scope, dereferencing each entry under the base produces additional bases of subtrees to be searched. The aliases under each additional base are also dereferenced during search to find yet more subtree bases, and so on. When all the additional subtrees have been identified, the search filter is applied to all the non-alias entries in all the subtrees and the entries that match the filter are returned.

**Dereferencing during searching in one-level searches**

In a search request with **LDAP_DEREF_SEARCHING** or **LDAP_DEREF_ALWAYS** and one-level scope, dereferencing each alias entry that is one level below the search base yields additional entries to search (even though they are no longer one level below the search base). The search filter is then applied to these additional entries and to the non-alias entries that are one level below the search base and the entries that match the filter are returned.

**Dereferencing and root DSE subtree search**

Aliases are never dereferenced when performing a subtree search starting at the root DSE (this is also known as a null-based subtree search). All alias entries are processed like 'normal' entries, as if **LDAP_DEREF_NEVER** was specified.
Errors during dereferencing

The common dereferencing errors and the resulting return codes are:

- loop detected during dereferencing: **LDAP_ALIAS_PROBLEM** (x’21’)
- no entry in this backend for dereferenced DN: **LDAP_ALIAS_DEREF_PROBLEM** (x’24’)

Alias examples

The following figure shows the directory structure used in the examples. The dashed lines indicate aliases. The dashed oval indicates the position of an aliased entry in the directory hierarchy, but the aliased entry does not actually exist.

**Note:** Fictitious attributetypes are used in the figure.

![Diagram of directory structure with aliases](image)

The following search examples show the entries that are returned for various combinations of search base, search scope, and dereference option. The filter in each example is "objectclass=*". Cases that are affected by alias dereferencing are indicated with an "*".

**Example 1:** Perform a search from the base "sw=SGProds, o=IBM, c=US".

- **scope = base**
- Returned entries with **LDAP_DEREF_NEVER, LDAP_DEREF_SEARCHING, LDAP_DEREF_FINDING**, or **LDAP_DEREF_ALWAYS** specified:
"sw=SGProds, o=IBM, c=US"

**scope = one-level**
- Returned entries with LDAP_DEREF_NEVER, LDAP_DEREF_SEARCHING, LDAP_DEREF_FINDING, or LDAP_DEREF_ALWAYS specified:
  "product=ZOSLDAP, sw=SGProds, o=IBM, c=US"

**scope = subtree**
- Returned entries with LDAP_DEREF_NEVER or LDAP_DEREF_FINDING specified:
  1. "sw=SGProds, o=IBM, c=US"
  2. "product=ZOSLDAP, sw=SGProds, o=IBM, c=US"
  3. "group=test, product=ZOSLDAP, sw=SGProds, o=IBM, c=US"
  4. "group=development, product=ZOSLDAP, sw=SGProds, o=IBM, c=US"
  5. "subgroup=Unit Test, group=test, product=ZOSLDAP, sw=SGProds, o=IBM, c=US"
- Returned entries with LDAP_DEREF_SEARCHING or LDAP_DEREF_ALWAYS specified:
  1. "sw=SGProds, o=IBM, c=US"
  2. "product=ZOSLDAP, sw=SGProds, o=IBM, c=US"
  3. "group=test, product=ZOSLDAP, sw=SGProds, o=IBM, c=US"
  4. "group=development, product=ZOSLDAP, sw=SGProds, o=IBM, c=US" (returned only once)

**Example 2:** Perform a search from the base "site=Pok, o=IBM, c=US".
- **scope = base**
  - Returned entries with LDAP_DEREF_NEVER or LDAP_DEREF_SEARCHING specified:
    "site=Pok, o=IBM, c=US"
  - Returned entries with LDAP_DEREF_FINDING or LDAP_DEREF ALWAYS specified:
    "sw=SGProds, o=IBM, c=US"
- **scope = one-level**
  - Returned entries with LDAP_DEREF_NEVER or LDAP_DEREF_SEARCHING specified:
    No entries returned
  - Returned entries with LDAP_DEREF_FINDING or LDAP_DEREF ALWAYS specified:
    "product=ZOSLDAP, sw=SGProds, o=IBM, c=US"
- **scope = subtree**
  - Returned entries with LDAP_DEREF_NEVER or LDAP_DEREF_SEARCHING specified:
    "site=Pok, o=IBM, c=US"
  - Returned entries with LDAP_DEREF_FINDING specified:
    1. "sw=SGProds, o=IBM, c=US"
    2. "product=ZOSLDAP, sw=SGProds, o=IBM, c=US"
    3. "group=test, product=ZOSLDAP, sw=SGProds, o=IBM, c=US"
    4. "group=development, product=ZOSLDAP, sw=SGProds, o=IBM, c=US"
    5. "subgroup=Unit Test, group=test, product=ZOSLDAP, sw=SGProds, o=IBM, c=US"
  - Returned entries with LDAP_DEREF_ALWAYS specified:
    1. "sw=SGProds, o=IBM, c=US"
    2. "product=ZOSLDAP, sw=SGProds, o=IBM, c=US"
    3. "group=test, product=ZOSLDAP, sw=SGProds, o=IBM, c=US"
    4. "group=development, product=ZOSLDAP, sw=SGProds, o=IBM, c=US" (returned only once)

**Example 3:** Perform a search from the base "product=ZOSLDAP, o=IBM, c=US".
- **scope = base**
  - Returned entries with LDAP_DEREF_NEVER or LDAP_DEREF_SEARCHING specified:
    "product=ZOSLDAP, o=IBM, c=US"
  - Returned entries with LDAP_DEREF_FINDING or LDAP_DEREF ALWAYS specified:
    "product=ZOSLDAP, sw=SGProds, o=IBM, c=US"
• Returned entries with `LDAP_DEREF_NEVER` or `LDAP_DEREF_SEARCHING` specified:
  No entries returned
• Returned entries with `LDAP_DEREF_FINDING` or `LDAP_DEREF_ALWAYS` specified:
  1. "group=test, product=ZOSLDAP, sw=SGProds, o=IBM, c=US"
  2. "group=development, product=ZOSLDAP, sw=SGProds, o=IBM, c=US"

Scope = subtree
• Returned entries with `LDAP_DEREF_NEVER` or `LDAP_DEREF_SEARCHING` specified:
  "product=ZOSLDAP, o=IBM, c=US"
• Returned entries with `LDAP_DEREF_FINDING` specified:
  1. "product=ZOSLDAP, sw=SGProds, o=IBM, c=US"
  2. "group=test, product=ZOSLDAP, sw=SGProds, o=IBM, c=US"
  3. "group=development, product=ZOSLDAP, sw=SGProds, o=IBM, c=US"
  4. "subgroup=Unit Test, group=test, product=ZOSLDAP, sw=SGProds, o=IBM, c=US"
• Returned entries with `LDAP_DEREF_ALWAYS` specified:
  1. "product=ZOSLDAP, sw=SGProds, o=IBM, c=US"
  2. "group=test, product=ZOSLDAP, sw=SGProds, o=IBM, c=US"
  3. "group=development, product=ZOSLDAP, sw=SGProds, o=IBM, c=US" (returned only once)

Example 4: Perform a search from the base "group=test, product=ZOSLDAP, o=IBM, c=US".
Scope = base
• Returned entries with `LDAP_DEREF_NEVER` or `LDAP_DEREF_SEARCHING` specified:
  Error - LDAP_NO_SUCH_OBJECT
• Returned entries with `LDAP_DEREF_FINDING` or `LDAP_DEREF_ALWAYS` specified:
  "group=test, product=ZOSLDAP, sw=SGProds, o=IBM, c=US"
Scope = one-level
• Returned entries with `LDAP_DEREF_NEVER` or `LDAP_DEREF_SEARCHING` specified:
  Error - LDAP_NO_SUCH_OBJECT
• Returned entries with `LDAP_DEREF_FINDING` specified:
  "subgroup=Unit Test, group=test, product=ZOSLDAP, sw=SGProds, o=IBM, c=US"
• Returned entries with `LDAP_DEREF_ALWAYS` specified:
  "group=development, product=ZOSLDAP, sw=SGProds, o=IBM, c=US"
Scope = subtree
• Returned entries with `LDAP_DEREF_NEVER` or `LDAP_DEREF_SEARCHING` specified:
  Error - LDAP_NO_SUCH_OBJECT
• Returned entries with `LDAP_DEREF_FINDING` specified:
  1. "group=test, product=ZOSLDAP, sw=SGProds, o=IBM, c=US"
  2. "subgroup=Unit Test, group=test, product=ZOSLDAP, sw=SGProds, o=IBM, c=US"
• Returned entries with `LDAP_DEREF_ALWAYS` specified:
  1. "group=test, product=ZOSLDAP, sw=SGProds, o=IBM, c=US"
  2. "group=development, product=ZOSLDAP, sw=SGProds, o=IBM, c=US"
Chapter 26. Change logging

The change log is a set of entries in the directory that contain information about changes to objects. Depending on configuration options, information about a change to a TDBM, LDBM, or CDBM entry, to the LDAP server schema entry (cn=schema), or to an object controlled by an application (for example, a RACF user, group, user-group connection, or general resource profile) can be saved in a change log entry. An LDAP search operation can be used to retrieve change log entries to obtain information about what changes have taken place.

Each LDAP server contains one change log. The change log entries are created in the same order as the changes are made and each change log entry is identified by a change number value, beginning with 1, that is incremented each time a change number is assigned to a change log entry. Therefore, the change number of a new change log entry is always greater than all the change numbers in the existing change log entries.

The change log is implemented in the GDBM backend. The change log uses a hard-coded suffix, cn=changelog. This suffix is a semi-reserved name when the GDBM backend is configured. The change log root (cn=changelog) must not overlap any suffix in any TDBM, SDBM, or LDBM backend and the change log suffix cannot be the source or target of a rename operation. If GDBM is not configured, the user can use cn=changelog as a ‘normal’ suffix in a TDBM, SDBM, or LDBM backend, however, we do not recommend this because that suffix will have to be renamed to avoid an overlap if GDBM is configured in the future.

Change logging is enabled by configuring GDBM in the LDAP server configuration file. Change log processing is controlled by configuration options in the GDBM backend. The changeLogging configuration option turns change logging on or off. The changeLogMaxEntries and changeLogMaxAge configuration options determine when removal of old change log entries takes place. See Chapter 8, “Customizing the LDAP server configuration” for more information. If none of these configuration options is specified in the GDBM section, the default is to start change logging with no limits on the size of the change log.

The changeLoggingParticipant configuration option can be used to specify if an LDBM, TDBM, or CDBM backend wants change log entries to be created for changes to entries in its backend. Similarly, the configuration option can be specified in the GDBM backend to determine if a change log entry should be created for a change to the LDAP server schema. If the option is not specified for a TDBM, LDBM, CDBM or GDBM backend, the default is to create change log entries for changes to that TDBM, LDBM, or CDBM backend or to the LDAP server schema.

If the GDBM backend is configured and the cn=changelog root entry does not exist in the GDBM backend when the server is started, the LDAP server generates the root entry. The root entry is created with an ACL that allows only the administrator to access the change log. The ACL is propagated to the change log entries. The user needs to use an LDAP modify operation to change this ACL to an appropriate ACL for his usage of the change log. The aclEntry and entryOwner attributes are the only attributes that can be modified. The aclPropagate and ownerPropagate attributes will always be TRUE.

Modifications to the change log are not logged. This means that no change sequence number will be returned for a persistent search request issued for the change log (cn=changelog).

Configuring the GDBM backend

Note: You can use the LDAP configuration utility, dsconfig, to configure GDBM.

The GDBM backend is configured in one of two ways: DB2-based (like TDBM) or file-based (like LDBM). In either configuration:

1. There can be at most one GDBM backend in the configuration file.
2. The **suffix** option cannot be specified in the GDBM backend.

3. If the **changeLoggingParticipant** option is specified, it controls whether a change log entry is created for a change to the LDAP server schema. Change log entries are never created for any changes to GDBM entries, including the suffix entry.

### Configuring a DB2-based GDBM backend

When configuring a DB2-based GDBM backend, the following configuration file options are required:

```markdown
- database GDBM GLDBGD31 [name]
- dbuserid
- dbowner
```

The `aclSourceCacheSize`, `attrOverflowSize`, `dnToEidCacheSize`, `dsnaoini`, `entryCacheSize`, `entryOwnerCacheSize`, `filterCacheBypassLimit`, `filterCacheSize`, `include`, `multiserver`, `persistentSearch`, `readonly`, `serverName`, `sizeLimit`, and `timeLimit` options can also be specified in the GDBM configuration section. The `changeLogging`, `changeLoggingParticipant`, `changeLogMaxAge`, and `changeLogMaxEntries` configuration options can be specified to control change logging activity. See Chapter 8, “Customizing the LDAP server configuration” for more information on these options.

When using DB2 to store its entries, the GDBM database is identical to a TDBM database and is created in the same way using the same SPUFI script. A DB2-based GDBM backend cannot share a database with a TDBM backend. Like TDBM, a DB2-based GDBM backend cannot run in 64-bit mode.

### Configuring a file-based GDBM backend

When configuring a file-based GDBM backend, the following configuration file options are required:

```markdown
- database GDBM GLDBGD31/GLDBGD64 [name]
```

The `commitCheckpointEntries`, `commitCheckpointTOD`, `databaseDirectory`, `fileTerminate`, `filterCacheBypassLimit`, `filterCacheSize`, `include`, `multiserver`, `persistentSearch`, `readOnly`, `sizeLimit`, and `timeLimit` are options that can also be specified in the GDBM configuration section. The `changeLogging`, `changeLoggingParticipant`, `changeLogMaxAge`, and `changeLogMaxEntries` configuration options can be specified to control change logging activity. See Chapter 8, “Customizing the LDAP server configuration” for more information on these options.

When using files to store its entries, the GDBM database is identical to an LDBM database and is created in the same way. Like LDBM, a file-based GDBM backend can run in 64-bit mode.

### Additional required configuration

Additional configuration is required for RACF to be able to log changes to a RACF user, group, connection, or resource profile:

- The SDBM backend must be configured. The SDBM suffix is needed to create a DN for the change log entry for a modification to a RACF user, group, connection, or resource profile. SDBM is also needed to retrieve the RACF user’s new password or other changed fields. The following option must be specified in the SDBM section of the configuration file to allow change log entries to be created for changes to resource profiles:
  ```
  enableResources on
  ```

- LDAP Program Call support must be enabled in the LDAP server containing the change log. To do this, add the following option to either the global section of the configuration file or to the command used to start the LDAP server:
  ```
  listen ldap://:pc
  ```

**Note:** This listen parameter for LDAP Program Call support is in addition to any other listen parameters you have specified.
There is no additional configuration needed to log changes to a TDBM, LDBM, or CDBM entry or to the LDAP server schema entry. If you do not want to create change log entries for changes to entries within a TDBM, LDBM, or CDBM backend, add the following configuration option to that backend section. You can add the same option to the GDBM section of the configuration file to stop the creation of change log entries for changes to the LDAP server schema entry:

```
changeLoggingParticipant off
```

---

**When changes are logged**

Change log records can be created when the change logging is activated and the GDBM backend is not in read-only mode.

**RACF changes**

An extended operation, `changeLogAddEntry`, is provided to allow an application to log changes to data that it controls. The initial use of this interface is by RACF to log changes to a RACF user, group, user-group connection, or general resource profile when the profile is added, modified, or deleted. The RACF changes can be driven through the LDAP server or be made directly to RACF. For a user password or password phrase change, RACF includes information that the password or password phrase changed in the change log entry. For other user changes, RACF does not provide specific field information at this time.

The creation of a change log entry when using this interface is entirely separate from the change to RACF, even if the RACF change is made using LDAP. The result is that a RACF change can occur without a change log entry being created (for example, if the LDAP server is not running or if the change log entry creation fails).

**TDBM, LDBM, CDBM, and schema changes**

If change logging is activated, each add, modify, delete, or modify DN operation of an entry in any TDBM, LDBM, or CDBM backend or modify of the LDAP server schema entry results in the creation of a change log entry, with the exception of the following:

- Change log entries are not created for entries that are added to a TDBM backend when using the load utility, `ldif2ds`.
- If the `changeLoggingParticipant off` option in the LDAP server configuration file is specified for this backend, then no changes in this backend are logged. The option can be specified for the GDBM backend to stop logging changes to the LDAP server schema entry.

The change log entry is created after the change to the TDBM, LDBM, or CDBM backend entry or the LDAP server schema has been committed. This change is not rolled back if the change log record can not be created.

**Change log schema**

The following object classes and their attributes define a change log entry. These object classes and attributes are always in the LDAP server schema.

- `objectclass: changeLogEntry`
  - `changeNumber`
    - an integer assigned to this change log entry
  - `targetDN`
    - the DN to which the change was applied. For RACF, this DN is created from a user, group, class, and/or resource name passed in by RACF and the SDBM suffix.
  - `changeType`
    - `add` | `modify` | `delete` | `modrdn`
changeTime
the time stamp of when the change is made (not when this entry is created)

changes
the added entry or the modifications, in LDIF format. This is fully supported for change log entries created by TDBM, LDBM, CDBM, and the LDAP server schema. However, the values for the userPassword, secretKey, replicaCredentials, ibm-slapdMasterPw, and ibm-replicaKeyPwd attributes are replaced with *ComeAndGetIt* in the change log entry. For change log entries created by RACF, this attribute is only present when a RACF user password or password phrase is changed, and contains either *ComeAndGetIt* or *NoEnvelope*, for example:

replace: racfPassword
racfPassword: *ComeAndGetIt*

newRDN
the new RDN specified in a TDBM, LDBM, or CDBM modify DN operation

deleteOldRdn
a boolean indicating if the old RDN was deleted in a TDBM, LDBM, or CDBM modify DN operation

newSuperior
the new superior distinguished name specified in a TDBM, LDBM, or CDBM modify DN operation

• objectclass: ibm-changelog

ibm-changelogInitiatorsName
the DN of the entity that initiated the change. For RACF, this DN is created from a userid passed in by RACF and the SDBM suffix.

Note: If a RACF user’s password or password phrase is changed using the currentvalue/newvalue support on a bind to the SDBM backend or on a bind using native authentication, the ibm-changelogInitiatorsName value is created from the userid under which the LDAP server is running (and not the bound user).

The change log root entry and change log entries also have the standard operational attributes: the ACL attributes, creatorsname, createtimestamp, modifiersname, modifytimestamp, and ibm-entryuuid (change log root only).

Change log entries
The change log consists of:
• One root (suffix) entry, named cn=changelog
• One or more leaf entries, named changenumber=nnn,cn=changelog

root entry
The change log root entry is generated by the LDAP server, when change logging is first enabled. The root entry cannot be created, renamed, or deleted by the user. The generated root entry contains a propagated ACL that allows only the administrator to access the change log. An appropriately authorized user can modify the root entry to change the ACL. Operations on the change log root are not replicated and do not result in the creation of a change log entry.

The generated root entry is:

dn: cn=changelog
objectclass: container
cn: changelog
ac1entry: group:cn=Anybody
ac1Propagate: TRUE
entryowner: access-id:adminDN
ownerPropagate: TRUE
The change log root entry should be modified using the modify operation to set access control for the change log. Only the aclEntry and entryOwner attributes can be modified. When GDBM is configured to be file-based, the aclEntry and entryOwner attributes can be entirely deleted, in which case the default ACL is used. See [Default ACLs with LDBM or TDBM] for more information. When GDBM is configured to be DB2-based, these attributes cannot be entirely deleted. The root entry ACL is always propagated to provide access control to the change log entries because change log entries are not created with their own ACL. The change log root entry can be modified as long as change logging is enabled (the GDBM backend is configured), even if change logging is not on.

**leaf entry**

Each change log entry is created as a leaf entry directly under the change log root entry, using the changeLogEntry and ibm-changelog objectclasses and attributes as described above.

- Change log entries are only created by the LDAP server. The user cannot directly add a change log entry. Also, the user cannot modify or rename a change log entry. Change log entries inherit the ACL of the change log root entry.

- Change log entries are deleted by the LDAP server when the change log is trimmed due to reaching a limit specified by the changeLogMaxEntries and changeLogMaxAge options in the configuration file. Change log entries can also be deleted by the user through a normal delete operation.

- User operations (search, compare, delete) on change log entries are allowed as long as change logging is enabled (the GDBM backend is configured), even if change logging is off. Add and trim operations by the LDAP server are not performed when change logging is off.

- If the GDBM backend is in read-only mode, delete and modify operations are not allowed. Add and trim operations by the LDAP server are not performed.

- Operations on change log entries are not replicated and do not result in the creation of change log entries.

The following is an example of a change log entry created by RACF:

dn: CHANGENUMBER=1815,CN=CHANGELOG
objectclass: CHANGELOGENTRY
objectclass: IBM-CHANGELOG
objectclass: TOP
changenumber: 1815
targetdn: RACFID=KEN,PROFILETYPE=USER,CN=MYRACF
changetime: 20030611161820.374472Z
changetype: MODIFY
changes: replace: racfPassPhrase
racfPassPhrase: *ComeAndGetIt*

ibm-changeinitiatorsname: RACFID=SUADMIN,PROFILETYPE=USER,CN=MYRACF

---

**Searching the change log**

The change log can be searched using the standard LDAP search APIs or command utilities.

- You can use any attribute in the search filter. A common search is with a "changenumber >= nnn" filter, where nnn is the largest changenumber value that was retrieved the previous time the search was done (the changenumber=nnn entry is retrieved again to ensure that the next part of the change log has not been trimmed).

- The change log entries matching the search filter are returned in increasing changenumber order.

- You cannot depend on there being change log entries for all consecutive change numbers. Some change numbers might be skipped.

- The change log (including the root entry) can be searched as long as change logging is enabled (the GDBM backend is configured), even if change logging is off.
Passwords in change log entries

To avoid including passwords in the changes attribute of a change log entry, the values of the userpassword, secretkey, replicacredentials, ibm-replicakeypwd, ibm-slapdmasterpw, racfpassword, and racfpassphrase attributes are replaced by *ComeAndGetIt*. You can use a search command to retrieve the password. For a RACF password or password phrase, see Chapter 16, “Accessing RACF information,” on page 269 for more information.

Unloading and loading the change log

The unload utility (ds2ldif) cannot be used to unload the contents of the change log. You should use the search operation to do this. Change log entries cannot be loaded into the change log. Both the add operation and the load utility (ldif2ds) fail when processing change log entries.

Trimming the change log

When change logging is on, the LDAP server periodically trims the change log based on the limits set in the LDAP server configuration file.

If a change log entry exceeds the age limit set using the changeLogMaxAge configuration option, it is removed from the log.

If the number of change log entries exceeds the limit set using the changeLogMaxEntries configuration option, the change log entries with the lowest changenumber values are removed. The number of entries that are removed depends on how GDBM is configured. When GDBM is file-based, entries are removed until the number of entries remaining is at the limit. When GDBM is DB2-based, entries are removed until the number of entries remaining is about 95% of the limit. For example, if changeLogMaxEntries is 1000 and the number of entries in the change log reaches 1001, the 51 lowest entries are deleted to reduce the number of entries to 950.

The change log is checked for trimming when the server is started (for a DB2-based change log only) and when change log entries are added. When DB2-based, the change log is also periodically trimmed, with a frequency determined by the server based on the change log limits and contents. The frequency cannot be directly modified. No trimming is performed when the GDBM backend is in read-only mode. Trimming is performed when the LDAP server is in maintenance mode (and GDBM is not read-only).

Change log information in the root DSE entry

The following attributes in the root DSE entry allow applications to determine the location of the change log and effectively use it. The attributes appear whenever change logging is enabled (the GDBM backend is configured), whether or not change logging is currently on.

changelog=cn=changelog
  the location of the change log

firstchangenumber=nnn
  the lowest change number currently in use in the change log. A zero indicates no change log entries.

lastchangenumber=nnn
  the highest change number currently in use in the change log. A zero indicates no change log entries.
Multi-server considerations

When running GDBM in multi-server mode, the change log is shared by all of the LDAP servers in the same cross-system group. Each LDAP server must have an identical GDBM backend configured to avoid the possibility of not logging a change that should be logged (or the reverse). Note that each TDBM, LDBM, and CDBM backend must run in the same mode as GDBM. They must all run in multi-server mode or all not run in multi-server mode.

How to set up and use the LDAP server for logging changes

1. Update the LDAP server configuration file:
   a. Add the GDBM backend section, including a change log size and age limit if desired. GDBM can be configured to be DB2-based or file-based. The following example starts change logging using a DB2-based change log with a maximum size of 1000 entries. Entries are automatically deleted when they become a day old.

   ```
   database gdbm GLDBGD31
dbuserid dbu1
changeLogging on
changeLogMaxEntries 1000
changeLogMaxAge 86400
   ```

   b. If you plan to log changes to RACF users, groups, user-group connections, and general resource profiles, you must also:

   Add the SDBM backend section, including the `suffix` and, optionally, the `enableResources` configuration options. The `enableResources` configuration option is only needed when logging changes to resource profiles. Following is an example:

   ```
   database sdbm GLDBSD31/GLDBSD64
suffix cn=myRacf
enableResources on
   ```

   Enable the PC Callable support (used by RACF to add change log entries to the LDAP server) by specifying the following option in the global section of the configuration file:

   ```
   listen ldap://:pc
   ```

   c. If you do not want to log changes to entries in a TDBM, LDBM, or CDBM backend or to the LDAP server schema entry, add the following option to the TDBM, LDBM, CDBM, or GDBM backend section (the GDBM backend controls change logging for the schema entry):

   ```
   changeLoggingParticipant off
   ```

2. If GDBM is DB2-based, create the DB2 database to be used by the change log. This involves updating and executing a SPUFI script. The database owner in the script must match the `dbuserid` value in the GDBM section of the configuration file. See Creating the DB2 database and table spaces for TDBM or GDBM for more information.

3. If you plan to log changes to RACF users, groups, connections, and resource profiles, perform the RACF configuration required to support creation of an LDAP change log entry for RACF changes to those profiles. If you plan to retrieve RACF password or password phrase envelopes, you need to perform the RACF configuration required to support creation and retrieval of the password or password phrase envelopes. See z/OS Security Server RACF Security Administrator's Guide for more information.

4. Restart the LDAP directory server. You will see the GDBM configuration options are displayed. For a DB2-based GDBM backend, this will look similar to the following.

   **Note:** The GDBM-0002 is a backend name assigned by the LDAP server or is the name specified on the `database` configuration option for the GDBM backend:

   ```
database GDBM GLDBGD31 GDBM-0002
aclSourceCacheSize: 100
attrOverflowSize: 255
changeLogging: on
changeLogMaxAge: 86400
```
changeLogMaxEntries: 1000
dbUserid: DBU1
dnToEidCacheSize: 1000
timeCacheSize: 5000
eidOwnerCacheSize: 100
timeCacheSize: 5000
multiserver: off
persistentSearch: off
readOnly: off
sizeLimit: 1000
suffix: CN=CHANGELOG
timeLimit: 3600

For a file-based GDBM backend, this will look similar to:

database GDBM GLDBG31/GLDBG64 GDBM-0002
changeLogging: on
changeLogMaxAge: 86400
changeLogMaxEntries: 1000
changeLoggingParticipant: on
commitCheckpointEntries: 10000
commitCheckpointTOD: 00:00
databaseDirectory: /var/ldap/gdbm
fileTerminate: recover
multiserver: off
persistentSearch: off
readOnly: off
sizeLimit: 1000
suffix: CN=CHANGELOG
timeLimit: 3600

If GDBM fails to start, the following message is issued:

GLD1106E GDBM-0002 backend initialization failed.

5. At this point, change logging is started. Depending on your configuration, a change to a RACF user,
group, connection, or resource profile, or to a TDBM, LDBM, or CDBM entry, or to the LDAP server
schema entry will result in the creation of a change log entry in the LDAP server.

6. If desired, modify the ACL on the change log root entry, cn=changelog, for your usage of the change
log. The initial ACL restricts client access to the change log to the LDAP administrator.

For example, to give read access to the change log to RACF user CLREADER, create an ldif file,
c1.ldif, similar to the following:

dn: cn=changelog
changetype: modify
add: aclentry
aclentry:access-id:racfid=clreader,profiletype=user,cn=myRacf:normal:rsc:
sensitive:rsc:critical:rsc:system:rsc

You should then modify the change log ACL by issuing a modify command similar to the following:

ldapmodify -h ldaphost -p ldapporT -D adminDn -w adminPw -f c1.ldif

7. You can search, delete, and compare change log entries using the LDAP client interfaces and
command line utilities. In particular, all change log entries can be viewed using a search similar to the
following:

ldapsearch -h ldaphost -p ldapporT -D adminDn -w adminPw -b "cn=changelog" "objectclass=*"

Part of the output from this search would look like:

cn=changelog
objectclass=top
objectclass=container
cn=changelog

CHANGENUMBER=1,CN=CHANGELOG
objectclass=CHANGELOGENTRY
8. If the changes attribute of a change log entry contains any of the following lines:
   - racfPassword: *NoEnvelope*
   - racfPassword: *ComeAndGetIt*
   - racfPassPhrase: *NoEnvelope*
   - racfPassPhrase: *ComeAndGetIt*
   - userpassword: *ComeAndGetIt*
   - replicacredentials: *ComeAndGetIt*
   - secretkey: *ComeAndGetIt*
   - ibm-slapdmasterpw: *ComeAndGetIt*
   - ibm-replicakeypwd: *ComeAndGetIt*

   then a password in the RACF user profile, TDBM, LDBM, or CDBM entry was changed. If the value is
   *ComeAndGetIt*, then you can try to retrieve the actual password value. See "Passwords in change log
   entries" on page 450 for information on retrieving passwords.

9. The LDAP root DSE entry contains useful information about the LDAP change log, including its suffix,
   and the lowest and highest change numbers currently in use. A command similar to the following one
   obtains this information:
   
   ```bash
   ldapsearch -h ldaphost -p ldapport -D adminDn -w adminPw -s base -b "" objectclass=*"
   ```

   Part of the output from this search would look like:
   
   ```
   changelog=cn=changelog
   firstchangenumber=1
   lastchangenumber=202
   ```

   **Note:** The LDAP server occasionally skips one or more change numbers, so it cannot be assumed
   that there is a change log entry for every number between 1 and 202. In addition, skips are
   created if you delete a change log entry that does not have the lowest number. Change
   numbers that are generated by the LDAP server are not guaranteed to be consecutive, but will
   always increase.
Chapter 27. Referrals

Referrals provide a way for servers to refer clients to additional directory servers. With referrals you can:
- Distribute namespace information among multiple servers
- Provide knowledge of where data resides within a set of interrelated servers
- Route client requests to the appropriate server

Following are some of the advantages of using referrals:
- 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.

This topic describes how to create referral entries in a TDBM or LDBM backend and how to configure a default referral for the LDAP server.

A referral entry can be added to a TDBM or LDBM backend to indicate that the backend does not contain that entry or any entries below it and to identify another LDAP server that may contain those entries. A referral entry returns referral information to the LDAP client if the target of a client operation is at or below the referral entry or if a search operation includes the referral entry within its search scope.

A default referral can be added to the LDAP server configuration file to identify another LDAP server that may contain entries that do not fall within any of the suffixes in this LDAP server. If the target of an operation is not at or below any suffix defined in the LDAP server, the LDAP server returns the default referral to the client.

This topic also discusses how to associate multiple servers using referrals and an example of associating a set of servers through referrals, basic replication (see Chapter 23, “Basic replication”), and advanced replication (see Chapter 24, “Advanced replication”).

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 in the referencing server. The value of the ref attribute points to the corresponding entry maintained in the referenced server. While the distinguished name (DN) in a value of the ref attribute is typically that of an entry in a naming context below the naming context held by the referencing server, it is permitted to be the distinguished name of any entry. A multi-valued ref attribute may be used to indicate different locations for the same resource. If the ref attribute is multi-valued, all the DNs in the values of the ref attribute should have the same value.

A referral entry must be a leaf entry. This means that no ancestor of an entry can be a referral entry. In addition, a referral entry cannot also be an alias entry.

Creating referral entries

Following is an example configuration that illustrates the use of the ref attribute.
In the example, Server A holds references to two entries:

- `o=ABC,c=US`
- `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 references to two equivalent servers, Server C and Server D.

The recommended 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 information and set the default referral to point back to its parent server.

### Associating servers with referrals

In order to associate servers through referrals:

- Use referral entries to point to other servers for subordinate references
- Define the default referral to point somewhere else, typically to the parent server

These steps are defined below.

### Pointing to other servers

Use referral entries to point to the other servers for subordinate references which are portions of the namespace below this server which are not serviced directly.

Referral entries are created in TDBM and LDBM backends. Referral entries consist of:

- **dn**: Specifies the distinguished name. It is the portion of the namespace served by the referenced server.
- **objectclass**: Specifies referral. Also include the object class extensibleObject.
- **ref**: Specifies the location of the referenced server. There is no required format for the value, however, the z/OS LDAP client can only follow a ref value which is in LDAP URL format. A LDAP URL has one of the following formats:
  - `ldap://hostname:port/DN`
  - `ldaps://hostname:port/DN`

  The default port (389 for a non-SSL connection or 636 for an SSL connection) is used if a port is not specified as part of the LDAP URL. The DN of the referral entry is used if a DN is not specified.
as part of the LDAP URL. The `ldap://` form is for a non-SSL connection while the `ldaps://` form is for an SSL connection. The `ldaps://` form is required if you are using non-standard ports and want to allow SSL connections to the referenced server. The DN value in the LDAP URL should match the DN of the referral entry. The `ref` attribute may be multi-valued, with each value specifying the LDAP URL of a different server. When multiple values are used, each LDAP URL should contain the same DN, and each server should hold equivalent information for the portion of the namespace represented by the DN. Note that you cannot specify a 0-length value for the `ref` attribute.

The z/OS LDAP server automatically adds the `extensibleObject` object class to a referral entry if it is not specified. This allows the RDN attributes to be added to the referral entry.

Following is an example:

```plaintext
dn: o=IBM,c=US
objectclass: referral
objectclass: extensibleObject
ref: ldap://Host1:389/o=IBM,c=US
ref: ldap://Host2:389/o=IBM,c=US
ref: ldap://Host3:1389/o=IBM,c=US
```

A TDBM or LDBM backend can contain any number of referral entries in its directory.

### Defining the default referral

Define the default referral to point to another server which services other portions of the namespace unknown to the referencing server. The 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 which possibly serves a disjoint portion of the namespace.

The default referral is specified using the `referral` option in the LDAP server configuration file and applies to all backends in the LDAP server. The value of the option must be an LDAP URL. Multiple default referrals may be specified. However, each one specified is considered equivalent; that is, each server referenced by a default referral should present the same view of the namespace to its clients.

The default referral LDAP URL does not include the DN portion and a DN, if specified, is ignored. The default port (389 for a non-SSL connection or 636 for an SSL connection) is used if a port is not specified as part of the LDAP URL. The `ldap://` form is for a non-SSL connection while the `ldaps://` form is for an SSL connection. The `ldaps://` form is required if you are using non-standard ports and want to allow SSL connections to the referenced server. Following is an example:

```plaintext
referral ldap://host3.ibm.com:999
```

**SSL/TLS note:** A non-secure client referral to a secure port is not supported. Also, a secure client referral to a non-secure port is not supported.

### Processing referrals

When LDAP clients request information from LDAP servers which do not hold the needed data, servers can pass back referral URLs which indicate one or more other servers to contact. The clients can then request the information from the referenced server. The z/OS client API, by default, chases referrals returned from servers. However, client applications can suppress referral chasing through the `ldap_set_option()` API. In this case, the application retrieves the referral from the LDAP client and processes it within the application. This option's scope is the LDAP handle, so a client could open multiple connections to one or more servers, some of which would chase referrals automatically, and some of which would not.
Servers present the referral URLs differently depending on the LDAP protocol version being used by the client. Referrals are presented to LDAP Version 2 clients in the error string, as the protocol does not provide a specific mechanism for indicating referrals. In LDAP Version 3, protocol elements are specifically defined to allow servers to present referral information to clients.

**Using LDAP Version 2 referrals**

Referrals are not supported by the LDAP Version 2 protocol. In order to provide referral information to LDAP Version 2 clients, the referral information is returned as part of the error string in the result message. Since clients do not generally examine the error string for results indicating LDAP_SUCCESS, the LDAP server returns LDAP_PARTIAL_RESULTS instead of LDAP_SUCCESS if referral information is present in the error string. Referral information may be present for any result other than LDAP_SUCCESS.

The referral information in the error string is returned as follows, where `\n` indicates a newline character:

```
Referral:\nldap://hostname:port/DN\n...
```

where Referral: is followed by a new line character (`\n`) and `ldap://hostname:port/DN\n` is an LDAP URL followed by a new line character. The ellipses (`...`) indicate a list of multiple referrals; that is, more LDAP URLs followed by new line characters.

**Limitations with LDAP Version 2 referrals**

Multiple referrals are only presented for partial search results when it is necessary to contact more than one additional server to complete the entire request. This would indicate that multiple referral entries were found in the referencing server that matched the search criteria. If chasing referrals, the client contacts every server presented in the list to continue the search request. For referral entries that have multi-valued ref attributes, the server sends only one of the LDAP URLs to a client using LDAP Version 2 protocol. This is because there is no provision for distinguishing between equivalent servers to contact (as indicated by multi-valued ref attributes) and multiple servers which must be contacted to complete a search request.

A second limitation of referrals in LDAP Version 2 is that operations can sometimes be ambiguous in their intent regarding whether the operation was targeted for “real” entries in the namespace, as opposed to the referral entries themselves. For searches, referral entries are only presented as referrals, since the usual intent of a search is to look at the real entries in the namespace. Server administrators must therefore use other means to examine existing referral entries, such as examining the database, or reviewing `ds2ldif` output. For update operations, default referrals for upward references are presented as referrals, so that read-only replica servers can forward update operations to the master replica. However, subordinate references indicated by a referral entry are not followed for update operations, rather they operate on the referral entry itself. This is necessary to allow an administrator the ability to delete or modify existing referral entries. Erroneous changes caused by misdirected update operations are generally avoided through access protection and schema rules.

**Using LDAP Version 3 referrals**

In LDAP Version 3, referrals are defined as part of the protocol. The LDAP Version 2 limitations mentioned above are overcome by elements of the protocol and extensions to the protocol. There are two methods of passing back referral information in the LDAP Version 3 protocol: referrals and search continuation references.

If the target of a request is a referral entry or is below a referral entry, or if the target does not fall within any of the suffixes in the LDAP server and a default referral is configured, then a result code of LDAP_REFERRAL is presented by the server to indicate that the contacted server does not hold the target entry of the request. The referral field is present in the result message and indicates another server (or set of servers) to contact. Referrals can be returned in response to any operation request except unbind and abandon which do not have responses. When multiple URLs are present in a given referral response, each one must be equally capable of being used to progress the operation.
If the target of a search is found in the directory but a referral entry is encountered during the rest of a one-level or subtree search, a referral is not returned. Instead, one or more search continuation references are returned. Search continuation references are intermixed with returned search entries. Each one contains a URL to another server (or set of servers) to contact, and represents an unexplored subtree of the namespace which potentially satisfies the search criteria. When multiple URLs are present in a given search continuation reference, each one must be equally capable of being used to progress the operation.

As mentioned earlier, the other limitation in LDAP Version 2 referral processing is related to the inability of a client to specify whether a request was targeted for a normal entry or a referral entry. For LDAP Version 3, this difficulty is overcome with a protocol extension in the form of the manageDsaiT control. Appendix C, “Supported server controls” describes manageDsaiT in detail. For typical client requests where the control is absent, whenever the server encounters an applicable referral entry while processing the request, either a referral or search continuation reference is presented. When the client request includes this control, the server does not present any referrals or search continuation references, but instead treats the referral entries as normal entries. In this case, even superior references through the use of default referrals are suppressed. The z/OS LDAP client operations utilities support the -M option to indicate that the requester is managing the namespace, and therefore wishes to examine and manipulate referral entries as if they were normal entries. See IBM Tivoli Directory Server Client Programming for z/OS for more information. An exception to the processing described above is that referral entries are always treated as normal entries during the second phase of a persistent search, even if the manageDsaiT control is not specified on the persistent search request. See PersistentSearch for more information.

Bind considerations for referrals
When LDAP clients chase referrals from one server to another, they typically need to bind to the referenced server before redirecting the original request. If you distribute your directory across multiple servers connected by referrals, you must consider the capabilities of the applications which access your directory, how they chase referrals, and how they can bind to the referenced servers.

For example, the z/OS LDAP operation utilities like ldapsearch and ldapmodify use the bind DN and password specified on the utility invocation, both when binding to the original target server and also when chasing referrals to other servers. If you want the LDAP operation utilities to automatically chase referrals across servers, then the same bind DN and password must be accepted on each of the servers connected by referrals.

If you use an approach where there are no common bind identities, then your applications will either be limited to unauthenticated access or they will require the ability to bind appropriately to each server when chasing referrals.

Consider the following approaches:
1. Use unauthenticated access for reading information to avoid the need to bind with a common identity. This makes sense if the data in the directory is general reference information that does not need to be protected.
2. Establish an ‘authentication’ backend for identity information that is the same on each server. This could be an SDBM backend, where the common authentication identities are in RACF, or a TDBM or LDBM backend that is the same on each server (replication could be used to ensure this). Access control over the other entries in the referral servers uses the distinguished names from the authentication backend to control access to the entries.
3. If you use the LDAP administrator DN to access the entries, configure the administrator DN and password identically in each of the referral servers.
Example: associating servers through referrals and basic replication

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.

   - Server A
     - Perhaps just a server used to locate other servers in the US. With no other knowledge, clients can come here first to locate information for anyone in the US.
   - Server B1
     - A hub for all data pertaining to IBM in the US. Holds all HQ information directly. Holds all knowledge (referrals) of where other IBM data resides.
   - Server B2
     - A replica of Server B1.
   - Server C
     - Holds all IBM Austin information.
   - Server D
     - Holds all IBM Endicott information.
   - Server E
     - Holds all Lotus® information.

3. Set up referral entries to point to the descendents in other servers.

Figure 54. Setting up the servers

Following is a description of each server:

   - Server A
     - Perhaps just a server used to locate other servers in the US. With no other knowledge, clients can come here first to locate information for anyone in the US.

   - Server B1
     - A hub for all data pertaining to IBM in the US. Holds all HQ information directly. Holds all knowledge (referrals) of where other IBM data resides.

   - Server B2
     - A replica of Server B1.

   - Server C
     - Holds all IBM Austin information.

   - Server D
     - Holds all IBM Endicott information.

   - Server E
     - Holds all Lotus® information.
4. Servers can also define one or more default referrals which point to “more knowledgeable” servers for anything that is not underneath them in the namespace. The default referrals go in the configuration file, not the backend.

**Note:** The default referral LDAP URLs do not include the DN portion.

```
# General Section
referral ldap://ibm.com:389
referral ldap://ibm.com:390
listen ldap://:789
.
.
# tdbm database definitions
database tdbm GLDBTD31
suffix "ou=Endicott,o=IBM,c=US"
```

**Figure 56. Server D configuration file**

5. Putting it all together. [Figure 57](#), [Figure 58](#) and [Figure 60](#) show these same six servers, showing the referral entries in the database as well as the default referrals which are used for superior references. Also included in Servers B1 and B2 are sample definitions for replication, setting up Server B2 as a replica of Server B1. This ensures that these two servers remains identical. Servers B1 and B2 are located on the same system, but use different ports.
Server A: Services "c=US"
host name "US.white.pages.com"

Configuration File
listen ldap://:/1234
database tdbm GLDBTD31
dbuserid userA
suffix "c=US"

directory
dn: c=US
objectClass: country
dn: o=IBM,c=US
objectClass: referral
objectClass: extensibleObject
ref: ldap://ibm.com:389/o=IBM,c=US
ref: ldap://ibm.com:390/o=IBM,c=US
dn: o=Lotus,c=US
objectClass: referral
objectClass: extensibleObject
ref: ldap://lotus.com:389/o=Lotus,c=US

Server E: Services "o=Lotus,c=US"
host name "lotus.com"

Configuration File
referral ldap://US.white.pages.com:1234
listen ldap:///:389
database tdbm GLDBTD31
dbuserid userE
suffix "o=Lotus,c=US"

Directory
dn: o=Lotus,c=US
objectClass: organization

Figure 57. Referral example summary (servers A and E)
Server B1: Services "o=IBM,c=US"
host name "ibm.com"

Configuration File

<table>
<thead>
<tr>
<th>referral</th>
<th>ldap://US.white.pages.com:1234</th>
</tr>
</thead>
<tbody>
<tr>
<td>listen</td>
<td>ldap://:389</td>
</tr>
<tr>
<td>database</td>
<td>tdbm GLDBTD31</td>
</tr>
<tr>
<td>dbuserid</td>
<td>userB1</td>
</tr>
<tr>
<td>suffix</td>
<td>&quot;o=IBM,c=US&quot;</td>
</tr>
<tr>
<td>suffix</td>
<td>&quot;cn=localhost&quot;</td>
</tr>
</tbody>
</table>

Directory

dn: cn=localhost
objectClass: container

dn: cn=ReplicaB2,cn=localhost
objectClass: replicaObject
replicaHost: ibm.com
replicaPort: 390
replicaBindDN: cn=Master
replicaCredentials: secret

dn: o=IBM,c=US
objectClass: organization

dn: ou=Austin,o=IBM,c=US
objectClass: referral
objectClass: extensibleObject
ref: ldap://austin.com:389/ou=Austin,o=IBM,c=US

dn: ou=Endicott,o=IBM,c=US
objectClass: referral
objectClass: extensibleObject
ref: ldap://endicott.com:789/ou=Endicott,o=IBM,c=US

dn: ou=HQ,o=IBM,c=US
objectClass: organizationalUnit

Figure 58. Referral example summary (server B1)
Server B2: Services "o=IBM,c=US"
host name "ibm.com"

Configuration File
referral ldap://US.white.pages.com:1234
listen ldap://:390
database tdbm GLDBTD31
dbuserid userB2
suffix "o=IBM,c=US"
masterServer ldap://ibm.com:389
masterServerDN cn=Master
masterServerPW secret

Directory
dn: o=IBM,c=US
objectClass: organization

dn: ou=Austin,o=IBM,c=US
objectClass: referral
objectClass: extensibleObject
ref: ldap://austin.com:389/ou=Austin,o=IBM,c=US

dn: ou=Endicott,o=IBM,c=US
objectClass: referral
objectClass: extensibleObject
ref: ldap://endicott.com:789/ou=Endicott,o=IBM,c=US

dn: ou=HQ,o=IBM,c=US
objectClass: organizationalUnit

Figure 59. Referral example summary (server B2)
Figure 60. Referral example summary (servers C and D)
Chapter 28. Client considerations

When an LDAP application is communicating with an LDAP server, you should consider the following special topics:
- Root DSE
- Monitor Support
- CRAM-MD5 authentication support
- UTF-8 data over the LDAP Version 2 protocol
- Attribute types stored and retrieved in lowercase
- Abandon behavior
- Changed return codes
- Reason codes

Root DSE

The root DSE is the entry at the top of the LDAP server directory information tree. All the namingcontexts (suffixes) in the LDAP server are directly below the root DSE. The root DSE contains information about the LDAP server, including the namingcontexts that are configured and the capabilities of the server.

The root DSE can be searched by specifying a zero-length base distinguished name. The search scope can be either base or subtree (the one-level scope is not supported).

Root DSE search with base scope

A root DSE search with base scope returns the contents of the root DSE. The root DSE attributes describe the LDAP server. The only search filter supported is objectclass=*. There is no access control checking for the root DSE, but an anonymous bind will fail if allowAnonymousBinds off is specified in the LDAP server configuration file. The supportedcontrol, supportedextension, and namingcontexts attributes may contain values that are contributed by plug-in extensions configured in the LDAP server.

The following example uses the ldapsearch utility to request a base search of the root DSE and shows sample output for the search:

```
ldapsearch -h ldaphost -p ldapport -s base -b "" "objectclass=*"
```

Following is an example of the information that the LDAP server will report on a search of the root DSE. A subset of these values may appear in your root DSE based on the server configuration choices you have made.

```
supportedcontrol=1.3.18.0.2.10.20
supportedcontrol=2.16.840.1.113730.3.4.3
supportedcontrol=2.16.840.1.113730.3.4.2
supportedcontrol=1.3.18.0.2.10.10
supportedcontrol=1.3.18.0.2.10.11
supportedcontrol=1.3.18.0.2.10.15
supportedcontrol=1.3.18.0.2.10.18
supportedcontrol=1.3.18.0.2.10.19
supportedcontrol=1.3.18.0.2.10.2
supportedcontrol=1.3.18.0.2.10.23
supportedcontrol=1.3.18.0.2.10.27
supportedcontrol=1.3.18.0.2.10.24
supportedextension=1.3.18.0.2.12.62
supportedextension=1.3.18.0.2.12.48
supportedextension=1.3.18.0.2.12.15
supportedextension=1.3.18.0.2.12.16
supportedextension=1.3.18.0.2.12.17
supportedextension=1.3.18.0.2.12.19
supportedextension=1.3.18.0.2.12.54
supportedextension=1.3.18.0.2.12.56
```
Following are Object Identifiers (OIDs) for supported and enabled capabilities:

<table>
<thead>
<tr>
<th>Short name</th>
<th>Description</th>
<th>OID assigned</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advanced replication</td>
<td>Identifies that this server supports advanced replication which includes subtree and cascading replication.</td>
<td>1.3.18.0.2.32.1</td>
</tr>
<tr>
<td>Short name</td>
<td>Description</td>
<td>OID assigned</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>Entry Checksum</td>
<td>Indicates that this server supports the <code>ibm-entryCheckSum</code> and <code>ibm-entryCheckSumOp</code> operational attributes.</td>
<td>1.3.18.0.2.32.2</td>
</tr>
<tr>
<td>Entry UUID</td>
<td>Identifies that this server supports the <code>ibm-entryuuid</code> operational attribute.</td>
<td>1.3.18.0.2.32.3</td>
</tr>
<tr>
<td>System restricted ACL support</td>
<td>Indicates that the server supports specification and evaluation of ACLs on system and restricted attributes.</td>
<td>1.3.18.0.2.32.7</td>
</tr>
<tr>
<td><code>cn=ibmpolicies</code> advanced replication subtree</td>
<td>Indicates that this server supports the replication of the <code>cn=ibmpolicies</code> subtree. This support is only available when advanced replication is configured.</td>
<td>1.3.18.0.2.32.18</td>
</tr>
<tr>
<td>Max age ChangeLog entries</td>
<td>Specifies that the server is capable of retaining changelog entries based on age.</td>
<td>1.3.18.0.2.32.19</td>
</tr>
<tr>
<td>Monitor operation counts</td>
<td>The server provides new monitor operation counts for initiated and completed operation types.</td>
<td>1.3.18.0.2.32.24</td>
</tr>
<tr>
<td>Null-based subtree search</td>
<td>Indicates that the server supports null-based subtree search operations, which search all the LDBM, TDBM, and CDBM entries in the server.</td>
<td>1.3.18.0.2.32.26</td>
</tr>
<tr>
<td>TLS capabilities</td>
<td>Specifies that the server is capable of performing Transport Layer Security (TLS).</td>
<td>1.3.18.0.2.32.28</td>
</tr>
<tr>
<td>Non-blocking advanced replication</td>
<td>Indicates that this server is capable of ignoring some errors received from a consumer (replica) server that would normally cause an update to be retransmitted periodically until a successful return code is received.</td>
<td>1.3.18.0.2.32.29</td>
</tr>
<tr>
<td>Kerberos capability</td>
<td>Specifies that the server is capable of performing Kerberos authentication.</td>
<td>1.3.18.0.2.32.30</td>
</tr>
<tr>
<td><code>ibm-allMembers</code> and <code>ibm-allGroups</code> operational attributes</td>
<td>Indicates that a backend supports searching on the <code>ibm-allGroups</code> and <code>ibm-allMembers</code> operational attributes. The members of a static, dynamic or nested group can be obtained by performing a search on the <code>ibm-allMembers</code> operational attribute. The static, dynamic and nested groups that a member DN belongs to can be obtained by performing a search on the <code>ibm-allGroups</code> operational attribute.</td>
<td>1.3.18.0.2.32.31</td>
</tr>
<tr>
<td>Modify DN (subtree move)</td>
<td>Indicates that a subtree can be moved to another subtree, within a backend. This move uses a new superior. It can also use a new RDN.</td>
<td>1.3.18.0.2.32.33</td>
</tr>
<tr>
<td>Modify DN (subtree rename)</td>
<td>Indicates that a subtree can be renamed. The DN of each entry under the subtree will also be changed. This rename uses a new RDN but not a new superior.</td>
<td>1.3.18.0.2.32.34</td>
</tr>
<tr>
<td>Short name</td>
<td>Description</td>
<td>OID assigned</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>Advanced replication configuration</td>
<td>Indicates that this server supports configuration of supplier servers in an advanced replication environment.</td>
<td>1.3.18.0.2.32.43</td>
</tr>
<tr>
<td>Global updates support</td>
<td>Indicates that this server supports the advanced replication of global updates using the replication topology in the cn=ibmpolicies subtree in the CDBM backend.</td>
<td>1.3.18.0.2.32.44</td>
</tr>
<tr>
<td>Advanced replication conflict resolution</td>
<td>Indicates that this server supports the ibm-slapdReplConflictMaxEntrySize attribute on a CDBM entry with an objectclass of ibm-slapdReplConfiguration. This attribute value indicates the maximum number of bytes that an entry can contain and still be resent to a target server as a result of advanced replication conflict resolution.</td>
<td>1.3.18.0.2.32.51</td>
</tr>
<tr>
<td>Lost and found log</td>
<td>Indicates that this server supports the lost and found log for archiving replaced entries as a result of the advanced replication conflict resolution.</td>
<td>1.3.18.0.2.32.52</td>
</tr>
<tr>
<td>Updated ibm-entryCheckSumOp operational</td>
<td>Indicates that this server supports an updated algorithm for the checksum calculation of the ibm-entryCheckSumOp operational attribute.</td>
<td>1.3.18.0.2.32.56</td>
</tr>
<tr>
<td>attribute calculation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Filter replication</td>
<td>Identifies that this server supports filtered replication which allows only required entries and a subset of attributes to be replicated. This support is only available when advanced replication is configured.</td>
<td>1.3.18.0.2.32.65</td>
</tr>
<tr>
<td>Fine grained timestamps</td>
<td>Indicates that this server supports advanced replication with fine grained timestamps that include microseconds.</td>
<td>1.3.18.0.2.32.94</td>
</tr>
<tr>
<td>ibm-replicationWaitOnDependency attribute</td>
<td>Indicates that this server supports the replication of the ibm-replicationWaitOnDependency attribute from the advanced replication agreement entry.</td>
<td>1.3.18.0.2.32.95</td>
</tr>
<tr>
<td>replication</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Root DSE search with subtree scope (Null-based subtree search)**

A root DSE search with subtree scope returns all the entries that match the search filter in the LDBM, TDBM, and CDBM backends configured in the LDAP server. This search is commonly referred to as a null-based subtree search. Note that the search does not include the root DSE itself, the LDAP server schema entry, SDBM entries, and GDBM entries (change log records). Alias entries are not dereferenced during the search, they are processed like normal entries and returned if they match the search filter. Referral entries in LDBM, TDBM, and CDBM return referrals to the client. Any filter can be specified for the subtree search.

A null-based subtree is implemented as a series of searches to each LDBM, TDBM, and CDBM suffix. These individual searches are each limited by the timelim and sizelimit options specified in the LDAP server configuration file. If a time limit or size limit is specified on the root DSE search request, then the individual searches are also limited by the amount of time remaining and the number of entries left to
return when that individual search is started. See the descriptions of the sizelimit and timelimit options in Chapter 8, “Customizing the LDAP server configuration” for more information. Each individual LDBM, TDBM, and CDBM search is subject to the normal LDBM, TDBM, and CDBM access control checking.

The following example uses the ldapsearch utility to request a subtree search of the root DSE for entries that have a cn value that begins with ken and shows sample output for the search.

```
ldapsearch -h ldaphost -p ldapport -D binddn -w passwd -s sub -b "" "cn=ken*"
```

```
cn=ken,o=ldbm
objectclass=person
objectclass=top
cn=ken
sn=smith

  cn=kenx,o=tdbm
  objectclass=person
  objectclass=top
cn=kenx
  sn=jones
```

**Monitor support**

You can retrieve statistics from the server by issuing a search request with a search base of cn=monitor and a filter of (objectclass=*). For details, see Monitoring performance with cn=monitor.

**CRAM-MD5 authentication support**

CRAM-MD5 authentication is supported on the IBM Tivoli Directory Server client utilities on other platforms, such as AIX®, Windows®, and Linux®. However, the manner in which it has been implemented on the IBM Tivoli Directory Server on other platforms varies from the support that is available on the z/OS LDAP server.

In order to perform a CRAM-MD5 authentication bind with the IBM Tivoli Directory Server client utilities on other platforms to the z/OS LDAP server, you must specify the bindDN with the -D option. The IBM Tivoli Directory Server client utilities on other platforms do not support the specification of the username on a CRAM-MD5 bind.

**UTF-8 data over the LDAP Version 2 protocol**

The LDAP Version 3 Protocol allows UTF-8 attribute values outside of the IA5 character set to be stored in the directory. This information must be able to be returned in some format to LDAP Version 2 clients. An additional LDAP server configuration file option, sendV3stringsOverV2as, which has the possible values ISO8859-1 or UTF-8, can be used to indicate which format to use when sending this information over the Version 2 protocol.

**Note:** Different clients treat non-IA5 data differently over the Version 2 protocol. Refer to the documentation for the client APIs you are using for more information.

**Attribute types stored and returned in lowercase**

The LDAP server stores and returns attribute types in lowercase (normalized). For example, the attribute type “productName” is returned as “productname”.

Chapter 28. Client considerations  471
Abandon behavior

The LDAP server reads additional operations as they arrive as long as the connection is not a secure connection and the previous operation is not bind, unbind, or extended operation. This allows the LDAP server to process abandon operations as they are received and affect previously submitted operations.

Changed return codes

For information about changed LDAP return codes, see z/OS Migration or Migration considerations for applications.

Reason codes

The LDAPResult construct is used by the LDAP protocol to return success or failure indications from servers to clients. This construct contains an error message field. Servers can optionally provide “human-readable” diagnostic information in this field. Depending on the location in the LDAP server where errors are detected, error messages generated may have the following format:

\[ \text{R<numeric digits> <diagnostic information> <traceback information>} \]

where:

numeric digits

Represents a specific reason code.

diagnostic information

Provides details about the reason for the failure.

traceback information

Is of the form (file_identification) and will assist you in diagnosing application or configuration problems.

Note the following regarding this error information:

• It is intended to be “human-readable” to assist in identifying problems detected by the server.
• It is not translated (English text only).
• It is not intended to be used as an application programming interface (API).
• Data returned may be changed by service or new releases of the product. (Again, it is not intended to be an API.)
• The reason code returned for a particular error can change and the reason code text can change.

Following is the current list of reason codes and associated diagnostic information returned by the LDAP server.

<table>
<thead>
<tr>
<th>Reason Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>R000001</td>
<td>Unable to allocate storage</td>
</tr>
<tr>
<td>R000004</td>
<td>Internal server error encountered</td>
</tr>
<tr>
<td>R000005</td>
<td>Unable to translate value for attribute 'name' from source_codepage to target_codepage</td>
</tr>
<tr>
<td>R000100</td>
<td>The password has expired</td>
</tr>
<tr>
<td>R000101</td>
<td>The new password is not valid</td>
</tr>
<tr>
<td>R000102</td>
<td>The user id has been revoked</td>
</tr>
<tr>
<td>R000104</td>
<td>The password is not correct or the user id is not completely defined (missing password or uid)</td>
</tr>
<tr>
<td>R000105</td>
<td>A bind argument is not valid</td>
</tr>
<tr>
<td>R000114</td>
<td>The realm portion of the value of attribute 'name' is not the RACF default realm</td>
</tr>
</tbody>
</table>
R000115  There is no RACF default realm

R000116  Cannot specify a value when deleting attribute 'name'

R000117  Cannot delete attribute 'name'

R000118  Cannot replace attribute 'name'

R000119  Cannot add or replace attribute 'name'

R000120  Cannot specify multiple values for attribute 'name'

R000121  Value for attribute 'name' must be same as value for DN

R000122  The value for attribute 'name' must be the DN of a user

R000123  The value for attribute 'name' must be the DN of a group

R000124  The value for attribute 'name' must be the DN of a user or a group

R000125  Attribute 'name' is not supported

R000126  Filter 'filter' is not supported for this base

R000127  Filter 'filter' contains a type without a value

R000128  Filter is not supported

R000129  Value 'value' is not supported for filter 'filter'

R000131  'name' is not a valid RACF DN

R000132  Value for attribute 'name' cannot be more than size characters

R000133  Value for attribute 'name' must be an integer less than size

R000134  The RACF type command created to satisfy this request is too long, probably due to specifying a long filter or attribute value or too many attribute values

R000135  Cannot perform this request on a reserved SDBM DN, 'name'

R000137  'name' is not a valid RACF DN for bind, check that the syntax is correct for a RACF user DN

R000138  There is no RACF user DN in the alternate DN list for a kerberos bind

R000139  RACF 'type' command failed

R000140  Cannot parse RACF 'command' output

R000141  Routine 'name' failed, rc=return_code

R000142  Cannot obtain the password of a RACF user

R000143  Bound user does not have the authority to perform this operation

R000144  Cannot specify a binary attribute in a compare operation

R000145  Must specify a value when deleting attribute 'name'

R000146  Cannot access entry with DN 'name' because SDBM is not configured to support RACF resources

R000147  The value for attribute 'name' must be the DN of a class

R000148  DN 'name' is not supported as a target of an SDBM operation

R000149  Attribute 'attribute' is not supported for entry with DN 'name'

R000200  Change log not active
R000201  Cannot decode attribute from request, rc=return_code

R000202  Request did not come over PC interface

R000203  Value for attribute out of range

R000204  Required value for attribute is missing

R000205  Unable to convert userid (value), group (value), class (value), or resource (value) to DN, rc=return_code

R000206  PC caller must be in supervisor state

R000207  attribute cannot be specified with attribute

R001001  Generalized Time value 'value' is not valid

R001005  Duplicate value encountered: value

R001008  Value specified for attribute 'name' does not match attribute syntax

R001011  COLLECTIVE keyword is not supported for attribute type 'name'

R001012  Attribute type 'name' is not defined

R001015  Cycle detected in superior hierarchy for 'identifier'

R001017  Syntax/matching rule inconsistency for attribute type 'name'

R001018  Attribute type 'name' is obsolete

R001024  Abstract class 'name' may not be a base object class

R001025  Multiple base structural object classes specified for 'name'

R001026  No structural object class specified for 'name'

R001027  Base structural object class 'name' may not be changed

R001029  Entry does not contain MUST attribute 'name'

R001030  Entry contains attribute 'name' which is not allowed for object class

R001031  Missing left parenthesis in definition: definition

R001032  Missing right parenthesis in definition: definition

R001038  Numeric object identifier 'value' is not valid

R001046  Missing closing quote for value 'value'

R001047  Missing opening quote for value 'value'

R001048  Missing closing brace for value 'value'

R001052  Non-numeric character found in integer value 'value'

R001053  Integer value of length size exceeds maximum length of size

R001055  Attribute type 'name' is not valid for the directory schema

R001056  Object class 'name' is not valid for the directory schema

R001060  Object class 'name' is obsolete

R001067  keyword keyword missing in schema definition:

R001069  Reference attribute type not found for IBM attribute type 'name'

R001072  More than one object class type keyword found in schema definition: definition
R002006  Empty DN component is not supported
R002007  Incorrect syntax in aclEntry attribute value 'value'
R002008  Permissions missing in aclEntry attribute value 'value'
R002018  An extraneous colon was found in aclEntry attribute value 'value'
R002019  An unsupported extensible filter was specified
R002020  A decoding error has been encountered while base64-decoding attribute 'name'
R002021  An incorrectly formatted 'name' attribute value has been encountered
R003029  The aclPropagate attribute requires the aclEntry attribute
R003030  The 'name' attribute cannot be used in the entry distinguished name
R003032  The ownerPropagate attribute requires the entryOwner attribute
R003057  Access denied because user does not have 'add' permission for the parent entry
R003062  Access denied because user does not have 'write' permission for all attributes in the new entry
R003070  Access denied because user does not have 'write' permission for all modified attributes
R003076  Access denied because user does not have 'delete' permission for the entry
R003082  Access denied because user does not have 'write' permission for all attributes in the old name
R003095  Access denied because user does not have 'compare' permission for the attribute
R003119  Access denied because user does not have 'write' permission for all attributes in the new name
R003125  Access denied because user does not have 'add' permission for the new superior entry
R003128  Unable to realign DN attributes because user does not have 'write' permission for attributes in 'name'
R003129  Realigning DN attributes would result in duplicate values for attribute 'name' in 'name'
R004017  No attributes specified for entry 'name'
R004019  Entry data is missing required RDN components
R004020  RDN contains duplicate values for attribute 'name'
R004022  Parent not found for entry 'name'
R004026  Entry 'name' not found in database
R004028  Search size limit exceeded
R004031  Search time limit exceeded
R004035  Attribute type 'name' may not be added or modified by users
R004038  Operation not allowed because backend is in read-only mode
R004041  Entry 'name' is not a leaf and may not be deleted
R004051  Entry 'name' does not contain attribute 'name'
Chapter 28. Client considerations 477
<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>R004145</td>
<td>The new superior may not be a referral or alias object</td>
</tr>
<tr>
<td>R004153</td>
<td>Parent of new entry 'name' is an alias entry</td>
</tr>
<tr>
<td>R004154</td>
<td>Entry is not a leaf and cannot be modified to be an alias entry</td>
</tr>
<tr>
<td>R004155</td>
<td>Alias entry 'name' points to itself</td>
</tr>
<tr>
<td>R004158</td>
<td>Cycle detected while dereferencing alias 'name'</td>
</tr>
<tr>
<td>R004159</td>
<td>Dereferencing 'name' failed because the resulting DN does not exist in this backend</td>
</tr>
<tr>
<td>R004160</td>
<td>Entry 'name' cannot be both an alias and a referral</td>
</tr>
<tr>
<td>R004161</td>
<td>Persistent search terminated because search base entry has been deleted</td>
</tr>
<tr>
<td>R004162</td>
<td>Operation not allowed because backend is not the sysplex owner</td>
</tr>
<tr>
<td>R004163</td>
<td>Dynamic group URL 'url' is not valid</td>
</tr>
<tr>
<td>R004164</td>
<td>An unsupported value 'value' is specified for attribute 'name'</td>
</tr>
<tr>
<td>R004166</td>
<td>The LDAP server is shutting down</td>
</tr>
<tr>
<td>R004176</td>
<td>The _passwd() function failed with error 'error_code'</td>
</tr>
<tr>
<td>R004177</td>
<td>Parent entry must have an objectclass attribute value of 'value'</td>
</tr>
<tr>
<td>R004178</td>
<td>Entry of objectclass 'value' not permitted with this level of replication</td>
</tr>
<tr>
<td>R004179</td>
<td>Refusing request, subtree 'name' is quiesced</td>
</tr>
<tr>
<td>R004180</td>
<td>Refusing request, access to replication topology is restricted</td>
</tr>
<tr>
<td>R004181</td>
<td>An entry can not be moved into replication topology subtree 'name'</td>
</tr>
<tr>
<td>R004182</td>
<td>Timestamps indicate a modify conflict, requesting refresh of entry 'name'</td>
</tr>
<tr>
<td>R004183</td>
<td>Advanced replication is configured but not available</td>
</tr>
<tr>
<td>R005001</td>
<td>Requested operation is not supported by the GDBM backend</td>
</tr>
<tr>
<td>R005002</td>
<td>Only the base change log entry can be modified</td>
</tr>
<tr>
<td>R005003</td>
<td>The base change log entry cannot be deleted</td>
</tr>
<tr>
<td>R005004</td>
<td>Only the aclEntry and entryOwner attributes can be modified</td>
</tr>
<tr>
<td>R006001</td>
<td>LDAP Client API api_name has returned an error code=return_code with an error message='error_string'</td>
</tr>
<tr>
<td>R006003</td>
<td>A decoding error has been encountered while decoding attribute(s): attr_type, rc=return_code</td>
</tr>
<tr>
<td>R006004</td>
<td>An encoding error return_code has been encountered while encoding response</td>
</tr>
<tr>
<td>R006006</td>
<td>Unsupported or inappropriate critical control 'identifier'</td>
</tr>
<tr>
<td>R006009</td>
<td>The extended operation request with OID=oid requires the critical control with OID=oid</td>
</tr>
<tr>
<td>R006010</td>
<td>Unsupported extended operation 'identifier'</td>
</tr>
<tr>
<td>R006011</td>
<td>The extended operation request with OID=oid does not support the critical control with OID=oid</td>
</tr>
<tr>
<td>R006023</td>
<td>Required field (name) missing</td>
</tr>
<tr>
<td>Error Code</td>
<td>Message</td>
</tr>
<tr>
<td>------------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td>R006024</td>
<td>Connection to server (url) failed</td>
</tr>
<tr>
<td>R006025</td>
<td>Incorrect ldapURL specified (url)</td>
</tr>
<tr>
<td>R006026</td>
<td>ldap_search failed rc=return_code</td>
</tr>
<tr>
<td>R006027</td>
<td>Unsupported authorization type=type</td>
</tr>
<tr>
<td>R006028</td>
<td>Expected attribute name missing from entry</td>
</tr>
<tr>
<td>R006029</td>
<td>Empty sequence in extended operation request name</td>
</tr>
<tr>
<td>R006030</td>
<td>Extended operation request does not have an object identifier</td>
</tr>
<tr>
<td>R006031</td>
<td>The type backend is not defined</td>
</tr>
<tr>
<td>R006032</td>
<td>Persistent search is allowed only when bound as the LDAP administrator</td>
</tr>
<tr>
<td>R006033</td>
<td>Persistent search must specify LDAP_DEREF_NEVER or LDAP_DEREF_FINDING</td>
</tr>
<tr>
<td>R006034</td>
<td>Persistent search is not allowed using the Program Call interface</td>
</tr>
<tr>
<td>R006035</td>
<td>Persistent search is not allowed with paged or sorted results</td>
</tr>
<tr>
<td>R006036</td>
<td>Persistent search is not supported by the backend</td>
</tr>
<tr>
<td>R006037</td>
<td>Unload extended operation is allowed only when bound as the LDAP administrator</td>
</tr>
<tr>
<td>R006038</td>
<td>Unload extended operation found multiple LDBM, TDBM, or CDBM backends to unload</td>
</tr>
<tr>
<td>R006039</td>
<td>Unload extended operation cannot find subtree DN 'name' to unload</td>
</tr>
<tr>
<td>R006040</td>
<td>Unload extended operation cannot find backend name 'name' to unload</td>
</tr>
<tr>
<td>R006041</td>
<td>Unload extended operation unable to open file 'file_name', errno=error_code, errstring=error_string</td>
</tr>
<tr>
<td>R006042</td>
<td>Unload extended operation has both backend name and subtree DN specified</td>
</tr>
<tr>
<td>R006043</td>
<td>Unload extended operation cannot find any LDBM, TDBM, or CDBM backend in the LDAP server configuration file to unload</td>
</tr>
<tr>
<td>R006044</td>
<td>Unload extended operation with a filter DN is only supported when advanced replication is activated in the CDBM backend</td>
</tr>
<tr>
<td>R006045</td>
<td>Unload extended operation is not able to find valid filters in filter DN 'name'</td>
</tr>
<tr>
<td>R006046</td>
<td>Unload extended operation does not support filtering of the schema entry</td>
</tr>
<tr>
<td>R006047</td>
<td>Unload extended operation does not support subtreeDN with unload localhost set to TRUE</td>
</tr>
<tr>
<td>R006048</td>
<td>SASL authentication requires the LDAP Version 3 protocol</td>
</tr>
<tr>
<td>R006049</td>
<td>Unsupported SASL authentication method 'name'</td>
</tr>
<tr>
<td>R006050</td>
<td>Server is not configured for client authentication</td>
</tr>
<tr>
<td>R006051</td>
<td>Client certificate is not available</td>
</tr>
<tr>
<td>R006052</td>
<td>User password is not available with native authentication</td>
</tr>
<tr>
<td>R006053</td>
<td>TLS is not supported on the connection</td>
</tr>
<tr>
<td>R006054</td>
<td>SSL/TLS is already active on the connection</td>
</tr>
<tr>
<td>R006055</td>
<td>Other operations are outstanding for the connection</td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>R007030</td>
<td>Multiple 'name' attributes found in DIGEST-MD5 response</td>
</tr>
<tr>
<td>R007031</td>
<td>Required 'name' attribute not found in DIGEST-MD5 response</td>
</tr>
<tr>
<td>R007032</td>
<td>Syntax error in DIGEST-MD5 response</td>
</tr>
<tr>
<td>R007033</td>
<td>Authorization DN in DIGEST-MD5 response does not match DN associated with user name</td>
</tr>
<tr>
<td>R007034</td>
<td>BIND DN 'name' is not the same as authentication DN 'name'</td>
</tr>
<tr>
<td>R007035</td>
<td>The value of DIGEST-MD5 response attribute 'name' is not valid</td>
</tr>
<tr>
<td>R007036</td>
<td>The DIGEST-MD5 authorization identifier is not a distinguished name</td>
</tr>
<tr>
<td>R007037</td>
<td>DIGEST-MD5 response attribute 'name' is not the same as the challenge value</td>
</tr>
<tr>
<td>R007038</td>
<td>Maximum DIGEST-MD5 buffer size must be at least 256 bytes</td>
</tr>
<tr>
<td>R007047</td>
<td>SASL EXTERNAL bind using the system identity requires the SDBM backend</td>
</tr>
<tr>
<td>R007051</td>
<td>DIGEST-MD5 response URL 'url' is incorrect or cannot be verified</td>
</tr>
<tr>
<td>R007052</td>
<td>LDAP server in maintenance mode; operations restricted to adminDN, masterServerDN and peerServerDN</td>
</tr>
<tr>
<td>R007060</td>
<td>SASL bind is in progress</td>
</tr>
<tr>
<td>R007061</td>
<td>No SASL mechanism specified</td>
</tr>
<tr>
<td>R007062</td>
<td>The EXTERNAL SASL mechanism is not available for the connection</td>
</tr>
<tr>
<td>R007063</td>
<td>Client credentials may not be specified for the EXTERNAL SASL mechanism</td>
</tr>
<tr>
<td>R007064</td>
<td>Concurrent BIND requests are not supported</td>
</tr>
<tr>
<td>R007065</td>
<td>No SASL BIND credentials</td>
</tr>
<tr>
<td>R007066</td>
<td>Unable to accept GSSAPI security context: Major 0xstatus, Minor 0xstatus - error_string</td>
</tr>
<tr>
<td>R007067</td>
<td>Unexpected security token received for GSSAPI continuation</td>
</tr>
<tr>
<td>R007068</td>
<td>Unable to wrap GSSAPI response: Major 0xstatus, Minor 0xstatus - error_string</td>
</tr>
<tr>
<td>R007069</td>
<td>A GSSAPI authorization identity may not be specified</td>
</tr>
<tr>
<td>R007070</td>
<td>Unable to unwrap GSSAPI response: Major 0xstatus, Minor 0xstatus - error_string</td>
</tr>
<tr>
<td>R007071</td>
<td>Requested GSSAPI security layer number is not supported</td>
</tr>
<tr>
<td>R007072</td>
<td>Maximum GSSAPI receive length size is too small</td>
</tr>
<tr>
<td>R007073</td>
<td>Unable to get GSSAPI wrap size limit: Major 0xstatus, Minor 0xstatus - error_string</td>
</tr>
<tr>
<td>R007074</td>
<td>Unable to obtain GSSAPI source name: Major 0xstatus, Minor 0xstatus - error_string</td>
</tr>
<tr>
<td>R007075</td>
<td>Unexpected SASL BIND credentials</td>
</tr>
<tr>
<td>R007076</td>
<td>No digest realm name is available</td>
</tr>
<tr>
<td>R007077</td>
<td>No user name specified for SASL BIND request</td>
</tr>
<tr>
<td>R007078</td>
<td>HMAC digest in SASL BIND request is not valid</td>
</tr>
<tr>
<td>R007079</td>
<td>The local Program Call interface supports just the EXTERNAL SASL mechanism</td>
</tr>
</tbody>
</table>

480  z/OS V1R11.0 IBM Tivoli Directory Server Administration and Use for z/OS
<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>R007080</td>
<td>A bind DN has been specified without a password</td>
</tr>
<tr>
<td>R007081</td>
<td>Anonymous binds are not allowed and no bind distinguished name exists</td>
</tr>
<tr>
<td>R007082</td>
<td>An internal SSL error has been encountered</td>
</tr>
<tr>
<td>R007083</td>
<td>Authentication with a reserved bind DN is not allowed</td>
</tr>
<tr>
<td>R008001</td>
<td>LDBM backend database is disabled</td>
</tr>
<tr>
<td>R008002</td>
<td>Entry 'name' contains multiple password values</td>
</tr>
<tr>
<td>R008003</td>
<td>Multiple entries contain uid 'name'</td>
</tr>
<tr>
<td>R008004</td>
<td>Clear password is not available</td>
</tr>
<tr>
<td>R008005</td>
<td>Nested group recursion detected for group 'name'</td>
</tr>
<tr>
<td>R008006</td>
<td>Dynamic group search filter 'filter' is not valid</td>
</tr>
<tr>
<td>R008008</td>
<td>No base entry specified in dynamic group URL 'url'</td>
</tr>
<tr>
<td>R008009</td>
<td>An internal LDBM backend error has occurred</td>
</tr>
<tr>
<td>R008101</td>
<td>TDBM backend database is disabled</td>
</tr>
<tr>
<td>R008102</td>
<td>Entry 'name' contains multiple password values</td>
</tr>
<tr>
<td>R008103</td>
<td>Multiple entries contain uid 'name'</td>
</tr>
<tr>
<td>R008104</td>
<td>Clear password is not available</td>
</tr>
<tr>
<td>R008105</td>
<td>Nested group recursion detected for group 'name'</td>
</tr>
<tr>
<td>R008106</td>
<td>Dynamic group search filter 'value' is not valid</td>
</tr>
<tr>
<td>R008107</td>
<td>Non-numeric object identifier 'identifier' is not allowed when using a TDBM database with DB_VERSION less than 4.0</td>
</tr>
<tr>
<td>R008108</td>
<td>No base entry specified in dynamic group URL 'url'</td>
</tr>
<tr>
<td>R008109</td>
<td>Unable to connect to DB2 subsystem 'name'</td>
</tr>
<tr>
<td>R008110</td>
<td>Unable to read an entry from the DB2 database</td>
</tr>
<tr>
<td>R008111</td>
<td>Unable to update an entry in the DB2 database</td>
</tr>
<tr>
<td>R008112</td>
<td>Unable to add an entry to the DB2 database</td>
</tr>
<tr>
<td>R008113</td>
<td>Unable to delete an entry from the DB2 database</td>
</tr>
<tr>
<td>R008114</td>
<td>Unable to commit the changes to the DB2 database</td>
</tr>
<tr>
<td>R008115</td>
<td>An internal TDBM backend error has occurred</td>
</tr>
</tbody>
</table>
R008116 DB2 subsystem 'name' is not available

R008117 Attribute object identifier 'identifier' is longer than 200 characters

R008118 Object class name 'name' is longer than 200 characters

R008119 DN 'name' exceeds the maximum length of size

R008120 Subtree move is not supported by the replica servers

R008121 Subtree rename is not supported by the replica servers

R008122 New superior is not supported by the replica servers

R008123 DN attribute realignment is not supported by the replica servers

R008124 Changelog root must have an explicit and propagating ACL

R008125 Matching rule 'rule' is not supported for syntax 'syntax_oid (syntax_desc)' when using a TDBM database with DB_VERSION less than 4.0

R008126 Exhausted all unique keys

R008127 DB2 database is no longer in a consistent state

R010001 Invalid character in descriptor 'descriptor'

R010002 Missing attribute type in DN component 'component'

R010003 Missing attribute value in DN component 'component'

R010004 No equality matching rule for DN attribute 'attribute'

R010005 No matching rule defined for string value 'value'

R010006 UTC Time value 'value' is not valid

R010007 Invalid IA5 character found in string value 'value'

R010008 Bit string value 'value' is not valid

R010009 Boolean value 'value' is not valid

R010010 Octet string value 'value' is not valid

R010011 Telephone number value 'value' is not valid

R010012 UUID value 'value' is not valid

R010013 Undefined LDAP syntax syntax

R010014 Country string value 'value' is not valid

R010015 No backend for DN 'name'

R010016 Backend initialization failed for DN 'name'

R010017 operation is not supported by the type backend

R010018 Search with null base DN requires either scope=base (for root DSE search) or scope=subtree (for null based subtree search)

R010019 Search with null base DN requires filter (objectclass=*)

R010020 Schema search requires scope=base

R010021 Schema search requires an object class presence or equality filter

R010022 Binary option is not supported by the type backend
<p>| R010023 | LDAP protocol version 3 is required for server controls |
| R010024 | Unable to decode value for control 'identifier' |
| R010025 | No value provided for control 'identifier' |
| R010026 | Attribute type 'identifier' already specified for a sort key |
| R010027 | Control 'identifier' is specified multiple times |
| R010028 | Critical control 'identifier' cannot be processed |
| R010029 | Maximum of size result sets has been reached |
| R010030 | Unable to compute search message digest |
| R010031 | Page size of zero is not valid for initial request |
| R010032 | Paged search results not found |
| R010033 | Continuation search request not same as initial request |
| R010034 | Unknown LDAP message type type |
| R010035 | Binary attribute type 'name' not allowed in DN |
| R010036 | No value provided for attribute 'name' |
| R010037 | Binary transfer is not supported for non-binary attribute type 'name' |
| R010038 | Embedded string delimiter found in value for attribute 'name' |
| R010039 | Incorrect ASN.1 encoding in DN component 'component' |
| R010040 | Unsupported ASN.1 type in DN component 'component' |
| R010041 | Server control does not have an object identifier |
| R010042 | Definition has no components: definition |
| R010043 | Substring filter for attribute 'name' has no value |
| R010044 | Substring filter type type is used incorrectly |
| R010045 | type filter has an empty filter set |
| R010046 | No equality matching rule for attribute type 'name' |
| R010047 | The new entry DN must exist in the same backend |
| R010048 | The specified permissions are not allowed for the access class in aclEntry attribute value 'value' |
| R010049 | routine failed with return code return_code, reason code reason_code |
| R010050 | Label 'name' is not defined |
| R010051 | ICSF services are not available |
| R010052 | Incorrect key length for label 'name' |
| R010053 | Incorrect key parity for label 'name' |
| R010054 | Encryption type type is not supported |
| R010055 | Encryption tag 'value' is not supported |
| R010056 | Encrypted data length is not a multiple of number |
| R010057 | Incorrect key value for label 'name' |</p>
<table>
<thead>
<tr>
<th>Message ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>R010058</td>
<td>Old and new password values were not supplied</td>
</tr>
<tr>
<td>R010060</td>
<td>LDAP protocol version 3 is required for extended operations</td>
</tr>
<tr>
<td>R010061</td>
<td>Only GetDnForUserid and GetPrivileges extended operations are supported</td>
</tr>
<tr>
<td>R010062</td>
<td>Unable to communicate with cross-system group owner</td>
</tr>
<tr>
<td>R010063</td>
<td>cn=monitor search requires scope=base</td>
</tr>
<tr>
<td>R010064</td>
<td>cn=monitor search requires filter (objectclass=*)</td>
</tr>
<tr>
<td>R010065</td>
<td>Unable to write attribute type 'name'</td>
</tr>
<tr>
<td>R010066</td>
<td>Unable to write to file 'file_name': error_code/reason_code -'error_string'</td>
</tr>
<tr>
<td>R010067</td>
<td>No referrals defined for read only replica, unable to update entry 'name'</td>
</tr>
<tr>
<td>R010050</td>
<td>Unable to retrieve normalized values for attribute 'name'</td>
</tr>
<tr>
<td>R010051</td>
<td>Unable to retrieve the next change ID for replication context with DN 'DN'</td>
</tr>
<tr>
<td>R010069</td>
<td>More than one replication agreement exists for DN 'DN' and consumer URL 'URL'</td>
</tr>
<tr>
<td>R010053</td>
<td>The 'name' and 'name' attributes are not allowed on DN 'DN'</td>
</tr>
<tr>
<td>R010054</td>
<td>DN 'DN' cannot be a replication context</td>
</tr>
<tr>
<td>R010055</td>
<td>Unable to find replication context for DN 'DN'</td>
</tr>
<tr>
<td>R010056</td>
<td>Gateway server must be a Master server</td>
</tr>
<tr>
<td>R010057</td>
<td>Credential objectclass not found</td>
</tr>
<tr>
<td>R010058</td>
<td>No objectclass attribute found in entry</td>
</tr>
<tr>
<td>R010059</td>
<td>Credential DN 'DN' is in use and cannot be deleted</td>
</tr>
<tr>
<td>R0100510</td>
<td>Filter DN 'DN' is in use and cannot be deleted</td>
</tr>
<tr>
<td>R0100511</td>
<td>Adding an ibm-replicationContext to DN 'DN' is not allowed</td>
</tr>
<tr>
<td>R0100512</td>
<td>Modification of the replication subentry's 'name' attribute is not allowed</td>
</tr>
<tr>
<td>R0100513</td>
<td>Modification is not valid. Replication context entry is no longer a replication context entry</td>
</tr>
<tr>
<td>R0100514</td>
<td>Cannot rename a replication topology entry</td>
</tr>
<tr>
<td>R0100515</td>
<td>Agreement with DN 'DN' does not support realignment</td>
</tr>
<tr>
<td>R0100516</td>
<td>Resulting entry is not in same replication context as target entry</td>
</tr>
<tr>
<td>R0100517</td>
<td>Unable to parse URL for attribute 'name'</td>
</tr>
<tr>
<td>R0100518</td>
<td>User does not have authority to create/update a replication topology entry</td>
</tr>
<tr>
<td>R0100519</td>
<td>Cannot modify a replication topology entry</td>
</tr>
<tr>
<td>R0100520</td>
<td>No request data is found</td>
</tr>
<tr>
<td>R0100521</td>
<td>Decoding error occurs when processing extended operation</td>
</tr>
<tr>
<td>R0100522</td>
<td>The syntax of the replication context DN is not valid</td>
</tr>
<tr>
<td>Error Code</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>R010753</td>
<td>The backend for entry 'DN' does not replicate</td>
</tr>
<tr>
<td>R010754</td>
<td>The entry 'DN' does not exist or is not a replication context</td>
</tr>
<tr>
<td>R010755</td>
<td>User does not have the authority to perform this extended operation</td>
</tr>
<tr>
<td>R010756</td>
<td>Error encountered when normalizing the DN</td>
</tr>
<tr>
<td>R010757</td>
<td>Unexpected error occurs while processing the extended operation</td>
</tr>
<tr>
<td>R010758</td>
<td>The action specified on the extended operation input is not valid</td>
</tr>
<tr>
<td>R010759</td>
<td>The timeout value specified on the extended operation input cannot be negative</td>
</tr>
<tr>
<td>R010760</td>
<td>The requested option specified for the error log extended operation is not valid</td>
</tr>
<tr>
<td>R010761</td>
<td>No replication agreement DN is specified</td>
</tr>
<tr>
<td>R010762</td>
<td>The syntax of the replication agreement DN is not valid</td>
</tr>
<tr>
<td>R010763</td>
<td>The entry 'DN' does not exist or is not under a replication context</td>
</tr>
<tr>
<td>R010764</td>
<td>The entry 'DN' does not exist, is not an agreement, or is not serviced by this server</td>
</tr>
<tr>
<td>R010765</td>
<td>The failure ID 'identifier' exists but is not logged for 'DN'</td>
</tr>
<tr>
<td>R010766</td>
<td>Data for the requested failure ID 'identifier' cannot be retrieved or formatted</td>
</tr>
<tr>
<td>R010767</td>
<td>The scope specified on the extended operation input is not valid</td>
</tr>
<tr>
<td>R010768</td>
<td>No replication agreement found for the context entry 'DN'</td>
</tr>
<tr>
<td>R010769</td>
<td>Change ID is not specified or is not valid with skip single option</td>
</tr>
<tr>
<td>R010770</td>
<td>Error encountered while retrieving the list of changes</td>
</tr>
<tr>
<td>R010771</td>
<td>Requested change ID 'identifier' does not match next change ID 'identifier'</td>
</tr>
<tr>
<td>R010772</td>
<td>Error encountered when server is updating the replication status</td>
</tr>
<tr>
<td>R010773</td>
<td>There are no pending changes to skip</td>
</tr>
<tr>
<td>R010774</td>
<td>Context DN is required but not specified</td>
</tr>
<tr>
<td>R010775</td>
<td>Replication agreement entry 'DN' was deleted</td>
</tr>
<tr>
<td>R010776</td>
<td>Failed to contact target 'host:port' using replication agreement 'DN'</td>
</tr>
<tr>
<td>R010777</td>
<td>Replication extended operation timed out</td>
</tr>
<tr>
<td>R010778</td>
<td>( number ) servers synchronized successfully out of ( number ) attempts</td>
</tr>
<tr>
<td>R010779</td>
<td>Replication topology extended operation failed</td>
</tr>
<tr>
<td>R010780</td>
<td>Server 'host:port' skipped because there is no connection to it</td>
</tr>
<tr>
<td>R010781</td>
<td>Server 'host:port' skipped because it is on hold</td>
</tr>
<tr>
<td>R010782</td>
<td>Replication context is already quiesced/unquiesced</td>
</tr>
<tr>
<td>R010783</td>
<td>Failure ID 'identifier' was not successfully removed</td>
</tr>
</tbody>
</table>
R010784  Failure ID 'identifier' successfully removed

R010785  The replication extended operation will not continue since the target server is a master server

R010786  Failed to quiesce supplier

R010787  Failed to build 'type' list

R010788  Consumer server down or not accepting updates from supplier. Retry for failure ID 'identifier' failed

R010789  Target server 'host:port' skipped by gateway

R010790  Number changes were skipped

R010791  Error occurs while parsing data for change ID 'identifier' for replica 'host'

R010792  Error occurs while applying filter to replication operation

R010793  Failed to retrieve data for failure ID 'identifier'

R010794  Unable to remove failure ID 'identifier' from the replication error log

R010795  The target server 'host:port' does not support replication topology entries

R010796  Failed to add the context DN as a suffix to the config file of the target server 'host:port'

R010797  The target server 'host:port' does not have a needed suffix and it does not support config file update

R010798  Unable to quiesce the target server

R010799  Unable to update the topology entries on the target server

R010800  Number failures removed successfully, number remain

R010801  Number failures retried successfully, number remain

R010802  Failure ID 'identifier' was not successfully retried

R010803  Failure ID 'identifier' successfully retried

R010804  The target server 'host:port' does not support the extended operation

R010805  Extended operation failed since target server is not a master server for the context 'DN'
Chapter 29. Performance tuning

Overview
Several server configuration options and facilities significantly affect the performance of the server. In addition, specific LDAP server backends operate in conjunction with other products that may require tuning to accommodate the LDAP server. For example, the TDBM and DB2-based GDBM backends use DB2, that provides a large set of tuning options. The SDBM backend provides access to the RACF database, that has its own product specific tuning options. This topic describes some of the things to consider when configuring your server for optimal performance.

General LDAP server performance considerations

Threads
The commThreads configuration option specifies the number of communication threads that handle requests from clients to the LDAP server. However, the primary role of each of these threads is to serve as a worker thread for processing client requests to the directory.

Each communication thread is shared among client connections and is used to process requests as they occur. Therefore, this option does not need to be set nearly as large as the expected number of concurrently connected clients.

Each communication thread requires some resources of its own, including low storage, a connection to DB2 (when TDBM or DB2-based GDBM is configured), and other system resources associated with threads. Therefore, you may want to avoid making this option larger than is needed.

It is recommended that commThreads be set to approximately two times the number of processors that are running in your LPAR. However, this is a general rule depending upon the activity that your LDAP server experiences.

If most requests are search requests retrieved from storage in TDBM caches or DB2 buffer pools, then additional commThreads might not provide much benefit. However, if most requests to the directory require I/O wait time, then additional commThreads might allow more client requests to run concurrently.

Debug settings
Activating the LDAP server debug trace facility impacts performance. If optimal performance is desired, debug should only be activated when it is necessary to capture diagnostic information.

Storage in the LDAP address space
The LDAP server generally requires a minimum of 32 megabytes to run in 31-bit mode, and 96 megabytes in 64-bit mode even with a minimal directory. This storage is required for maintaining server-wide information and for processing client requests.

Note: These are estimates only, and the need for storage can increase depending on the size of any LDBM directories configured, and the size of the caches.

LDAP server cache tuning
The LDAP server implements many caches to help reduce processing time and to avoid access to the database. These caches are beneficial when most accesses to the directory are read operations. Tuning these caches involves monitoring their effectiveness and adjusting their size to increase the percent hit rate.
Increasing cache sizes may increase the amount of storage required by the server. Some caches are invalidated by update activities. If this is a frequent occurrence, increasing the cache size may be of little or no benefit. If the cache hit rate is never any higher than zero for a particular cache, the cache can be disabled by setting its size to 0. However, even caches with seemingly low cache hit rates might provide some benefit, therefore, you should generally avoid disabling them unless close monitoring is done to ensure they are not beneficial.

Most caches in the LDAP server are enabled by default, and the default sizes generally provide some benefit to most installations. However, many installations might benefit from additional tuning. The following approach can be used to evaluate the cache sizes:

- Monitor the cache performance during typical workloads: You can use either the \texttt{cn=monitor} search or the operator console \texttt{MODIFY} command to retrieve current cache statistics. These are described later in this topic.

  \textbf{Note:} The monitor search must be used with a scope of subtree or one-level to retrieve the cache statistics, since the caches are backend specific.

- Examine the cache hit rate, the current number of entries, and the maximum allowed entries (configured size). Also, note the number of cache refreshes and the average size of the cache at refresh.

- If the cache hit rate is well below 100\% and the cache is frequently fully populated, consider increasing the cache size. Since this is a configuration option, you must change the server configuration file and restart the server to affect the change.

The following caches are implemented in the LDAP server:

\textbf{ACL source cache}

This cache holds information regarding ACL definitions within the database. Retrieval of information from this cache avoids database read operations when resolving access permissions. This cache is implemented in the TDBM and DB2-based GDBM backends.

\textbf{DN cache}

This cache holds information related to the mapping of distinguished names between their raw form and their canonical form. Retrieval of information from this cache reduces processing required to locate entries in the database. This is a server-wide cache, and is implemented in the internal schema backend. To alter its setting from the default, adjust the \texttt{dnCacheSize} configuration option in the global section of the LDAP server configuration file.

\textbf{DN to eid cache}

This cache holds information related to the mapping of distinguished names in their canonical form and their entry identifier within the database. Retrieval of information from this cache avoids database read operations when locating entries within the database. This cache is implemented in the TDBM and DB2-based GDBM backends.

\textbf{entry cache}

This cache holds information contained within individual entries in the database. Retrieval of information from this cache avoids database read operations when processing entries within the database. This cache is implemented in the TDBM and DB2-based GDBM backends.

\textbf{entry owner cache}

This cache holds information regarding ACL definitions within the database. Retrieval of information from this cache avoids database read operations when resolving access permissions. This cache is implemented in the TDBM and DB2-based GDBM backends.

\textbf{filter cache}

This cache holds information related to the mapping of search request inputs and the result set. This cache is implemented in the TDBM, LDBM, CDBM, and GDBM backends. For TDBM and DB2-based GDBM, retrieval of information from this cache avoids database read operations when
processing search requests. For LDBM, CDBM, and file-based GDBM, this cache helps reduce
processing time for searches with complex filtering. Note that the GDBM filter cache is disabled,
by default.

Operations monitor

If the operations monitor is enabled, the LDAP server monitors search statistics for the types of search
patterns that are configured and stores search statistics for each search pattern. The operations monitor
supports two types of search patterns, searchStats and searchIPStats. A searchStats pattern consists of
the search parameters (search base, scope, filter, and attributes to be returned) and status (success or
failure). The searchStats pattern is useful for evaluating the performance of search patterns. A
searchIPStats pattern consists of the same elements as searchStats pattern does, but also includes the
client IP address. The searchIPStats pattern is useful in determining if there are any specific clients
spamming the LDAP server. The operationsMonitor configuration option determines which types of
search patterns are monitored. See “Monitoring performance with cn=monitor” on page 497 for more
information about the operations monitor.

A new search pattern is added to the operations monitor whenever the search pattern of an incoming
search does not match one of the existing operations monitor search patterns. When the number of search
patterns exceeds the value of the operationsMonitorSize configuration option (the cachesize attribute in
the cn=operations,cn=monitor entry), the least recently used search patterns are trimmed. The total
number of trimmed search patterns is stored in the numtrimmed attribute of the

The LDAP server uses the WLM health service to indicate a health value to WLM. The WLM health value
is calculated by the number of failures during the past 5000 operations as long as one minute has passed
since the value was last calculated. If the percentage of failures changes by 25% or more, the z/OS LDAP
server increases or decreases the WLM health value. An LDAP server operation is considered a failure
when it has one of the following return codes:

- LDAP_OPERATIONS_ERROR (1)
- LDAP_TIMELIMIT_EXCEEDED (3)
- LDAP_ADMIN_LIMIT_EXCEEDED (11)
- LDAP_BUSY (51)
- LDAP_UNAVAILABLE (52)
- LDAP_UNWILLING_TO_PERFORM (53)
- LDAP_OTHER (80)

The wlmExcept configuration option can be used to specify the client’s IP address, the bound user’s
distinguished name (DN), or both to route those requests to any configured WLM transaction name (TN).
The wlmExcept configuration option can be specified multiple times within the LDAP server configuration
file to allow multiple client IP addresses or bound users’ DNs to be associated with the same or different
WLM transaction name. The order that the `wlmExcept` configuration options are specified in the LDAP server configuration file determines the order the LDAP server uses to match incoming client requests and route them to the WLM transaction name. See page 108 for more information about the `wlmExcept` configuration option.

The `WLMEXCEPT` operator modify command can be used to change the routing of incoming client requests to new or different WLM transaction names while the server is running. If the operations monitor is configured, the `cn=operations,cn=monitor` entry has `searchStats` and `searchIPStats` attribute values with an ID parameter that indicates the operations monitor ID (OPID). See Monitoring performance with cn=monitor for details about the `searchStats` and `searchIPStats` attribute value format. The `WLMEXCEPT` operator modify command uses the OPID to associate a search pattern to a WLM transaction name. If the transaction name specified on the `WLMEXCEPT` operator modify command does not exist in WLM, a new WLM enclave is created; otherwise an existing enclave is used. Each time the `WLMEXCEPT` operator modify command is issued, the new mappings are added before any of the configured `wlmExcept` configuration options or previously issued `WLMEXCEPT` operator modify commands. See LDAP server operator commands for more information about the `WLMEXCEPT` operator modify command.

The `WLMEXCEPT` operator modify commands last for the life of the LDAP server, however, the `RESET WLMEXCEPT` can be issued to remove all previously issued `WLMEXCEPT` operator modify commands and default to using the initial LDAP server configuration. If the operations monitor ID (OPID) is specified on the `RESET WLMEXCEPT` operator modify command, then just that specific WLM routing for that search is removed. See LDAP server operator commands for more information about the `RESET WLMEXCEPT` operator modify command.

If the operations monitor is enabled, the `searchStats` and `searchIPStats` attributes in the `cn=operations,cn=monitor` entry can be used to identify spamming client applications or certain search requests that should have a higher priority within the LDAP server. This type of information is very valuable when configuring LDAP to use WLM transaction names and assigning service or report classes to those transaction names. For a spamming client application, a WLM transaction name with a low priority service or report class ought to be used. For important search requests, a WLM transaction name with a high priority service or report class ought to be used.

**Configuring LDAP with WLM examples**

Assume that WLM has been configured to contain the following transaction names and service classes for LDAP.

Assume that WLM has been configured to contain the following transaction names and service classes for LDAP.
Example 1:

After analyzing the searchStats and searchIPStats attributes returned on a cn=operations,cn=monitor search, it has been determined there is a spamming LDAP client application on IP address 1.2.3.4 that has been affecting performance of the LDAP server. Also, requests from bound user cn=importantguy,o=ibm should have a higher priority within the LDAP server.

The LDAP administrator can add the following wlmExcept configuration options to route these requests to the appropriate WLM transaction name in Figure 61:

```
wlmExcept EXCEPT1 1.2.3.4
wlmExcept EXCEPT2 cn=importantguy,o=ibm
```

After the LDAP server is restarted, LDAP client requests originating from IP address 1.2.3.4 are routed to WLM transaction name EXCEPT1 with a service class of SPAMREQ. The server routes requests from bound user cn=importantguy,o=ibm to WLM transaction name EXCEPT2 with a service class of CRITREQ. If there are requests from any other client IP addresses or bound users, the server routes them to the GENERAL WLM transaction name that has a service class of NORMREQ. See Z/OS MVS Planning: Workload Management for more information about configuring WLM.

If the transaction name specified on the wlmExcept configuration option or on the WLMEXCEPT modify command does not exist in WLM (such as EXCEPT3 does not exist in Figure 61), any client requests associated with that transaction name would use the default service class HIGHREQ.

Example 2:

After analyzing the searchStats and searchIPStats attribute values returned on a cn=operations,cn=monitor search, it has been determined that the search identified by a specific searchIPStats value should be mapped to WLM transaction name EXCEPT2 because it is an important LDAP search.

```
cn: cn=monitor,cn=operations
searchIPStats: ldap://fe00::f4f7:0:7442:750f/OU=_v,O=_v??sub?(|(&(sn=_v)(cn=_v*))
(description=*_v*))?success,numOps=42,avg=246,rate=5,maxRate=37,maxRateTime=20080313132626.545031Z,createTime=20080313132615.953823Z,
```

Chapter 29. Performance tuning 491
The following WLMEXCEPT operator modify command is used to route the search pattern in the searchIPStats attribute to the EXCEPT2 WLM transaction name:

```
F LDAPSRV,WLMEXCEPT EXCEPT2,2741
```

**Note:** The operations monitor ID (OPID) value is 2741 for the search on the searchIPStats attribute value.

If it is determined later that the client search requests identified by OPID 2741 no longer need to be routed to WLM transaction name EXCEPT2, the following RESET WLMEXCEPT operator modify command is used to remove just the specific WLM routing for that search:

```
F LDAPSRV,RESET WLMEXCEPT,2741
```

See [LDAP server operator commands](#) for more information about the RESET WLMEXCEPT operator modify command.

---

**LDBM performance considerations**

The LDAP server LDBM backend uses the z/OS UNIX System Services file system for its persistent storage of the directory entry data. When the LDAP server is executing, the entire directory contents are held in its address space, including index structures for quick access.

This provides extremely fast access to the directory data. LDAP operations that read directory data involve no DASD I/O during the operation. LDAP operations that update the directory generally perform DASD I/O only to write the changed information to the LDBM checkpoint file. The index updates only occur within the LDAP server address space, and are not stored on DASD. Compared to TDBM, LDBM operations generally run much faster and use much less processor resources.

However, LDBM has inherent scalability limitations. The following resources are affected by the size of the directory, and are generally proportional to the LDBM directory size:

- The storage required within the LDAP server address space
- The LDAP server initialization time, both elapsed time and processor time
- The time required to commit the directory
- The DASD space required for the directory, including space for commit processing.

**Storage in the LDAP address space for LDBM data**

Since the entire LDBM directory is kept in storage in the LDAP address space, you need to plan accordingly. The amount of storage required can be estimated from the size of the LDIF data used to load the directory. For 31-bit mode, the storage needed to contain the data is about 7 to 10 times the size of the LDIF file. If you are running in 64-bit mode, storage requirements increase to as much as 10 to 15 times the size of the LDIF file.

These are estimates only. Furthermore, these estimates pertain only to the storage required to hold the LDBM directory representation. You must plan for additional storage for running the server as mentioned in "Storage in the LDAP address space" on page 487.

**Note:** When running in multi-server mode, each of the LDAP servers sharing the LDBM directory in the sysplex (known as sysplex replicas for this backend) also keeps the entire LDBM directory in its address space, therefore, requires about the same amount of storage as the LDAP server that owns the shared LDBM directory in the sysplex (known as the sysplex master for this backend).

For systems that are constrained on storage, or for very large directories, you may need to use TDBM instead of LDBM.
**LDAP server initialization time with LDBM**

Whenever the LDAP server is restarted, it reads the entire LDBM directory into storage and builds the necessary index structures for efficient search processing. This can take several minutes depending on the speed of the processor, the speed of the DASD that holds the data, and the competition for resources because of other workloads. Generally, the initialization elapsed time and the consumed processor time during initialization are proportional to the size of the directory.

When running in multi-server mode, each of the sysplex replicas receives a copy of the directory from the sysplex master using sysplex services. The amount of processor time consumed by a sysplex replica during initialization is about the same as that used by the sysplex master when it reads the directory into storage. However, the elapsed time required to initialize the sysplex replica tends to be longer than that required to initialize the sysplex master server. Initialization of a sysplex replica also consumes processor time on the sysplex master as it sends the data to the sysplex replica. The processor time consumed on the sysplex master is about one-third the processor time consumed on the sysplex replica during sysplex replica initialization.

**Database commit processing**

The LDBM directory contents are kept on DASD in the database files and the checkpoint file. There is one checkpoint file for the backend, and a separate database file for each suffix defined in the backend. The database files contain the overall contents of each entry in the database at the last database commit point. The checkpoint file contains individual entry updates that occurred since the last database commit point, recorded as sequential changes beyond the contents of the database file.

To avoid unbounded growth of the checkpoint file, the database is periodically committed. Commit processing writes new copies of the database and checkpoint files such that the new database files contain the up-to-date contents of each entry in the directory, and the checkpoint file contains no individual file update information. Database commits occur at the following times:

- When the number of checkpoint entries exceeds the value of the `commitCheckpointEntries` option in the LDAP server configuration file.
- When the time of day reaches the `commitCheckpointTOD` option in the LDAP server configuration option.
- When the LDAP server COMMIT operator command is invoked.
- When the LDAP server is shut down normally.
- When the LDAP server is restarted and uncommitted updates exist in the checkpoint file after an abnormal termination of the LDAP server.

Commit processing requires both processor and DASD resources, and the resources needed increase as the size of the directory increases. For large directories, commit processing may take a minute or more depending on competition for resources.

When commit processing occurs, a new copy of each directory file is created in its entirety before deleting the old copy and before deleting the previous checkpoint file. Therefore, you should plan enough DASD space to accommodate two copies of each directory file plus the maximum size of your checkpoint file. The amount of DASD space needed for the checkpoint file is highly dependent on the nature of the updates performed, and is best determined by experimentation.

During commit processing, no update requests are processed. Therefore, you should consider avoiding unplanned commits caused by the configuration option `commitCheckpointEntries`. Instead, consider using `commitCheckpointTOD`, automated methods of using the LDAP server COMMIT operator command, or planned shutdowns of the LDAP server to control when commit processing occurs.
**DASD space for LDBM data**
The amount of space needed to store an LDBM backend in a z/OS UNIX System Services file system is approximately four to six times the size of the expected input LDIF data. Generally, the space required to hold the LDBM backend data is two to three times the size of the expected input LDIF data. However, during the LDBM commit process each of the LDBM database files is copied, therefore, resulting in occasionally needing twice the amount of file system space.

**Sample LDBM benchmark data**
The following data was gathered from benchmarks using an LDBM database on z/OS UNIX System Services using ESS800 model 2105 DASD, and running on a z9® model 2094 processor with no competing applications running.

<table>
<thead>
<tr>
<th>Table 75. LDBM benchmark data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database size</td>
</tr>
<tr>
<td>LDIF file size</td>
</tr>
<tr>
<td>LDAP server storage</td>
</tr>
<tr>
<td>Sysplex master initialization elapsed time</td>
</tr>
<tr>
<td>Sysplex master initialization processor time</td>
</tr>
<tr>
<td>Sysplex replica initialization elapsed time</td>
</tr>
<tr>
<td>Sysplex replica initialization processor time</td>
</tr>
<tr>
<td>Sysplex master processor time consumed by replica initialization</td>
</tr>
<tr>
<td>LDAP server database commit elapsed time</td>
</tr>
<tr>
<td>LDAP server database commit processor time</td>
</tr>
</tbody>
</table>

**CDBM performance considerations**
If the CDBM backend is only used to store configuration entries, then no tuning is necessary. However, if the CDBM backend is used to store user defined entries, see LDBM performance considerations for CDBM tuning information.

**TDBM performance considerations**
The z/OS LDAP server TDBM backend uses IBM Database 2™ (DB2), a powerful and scalable database product, for its data storage facility. In the most optimal LDAP environments, directory data is fairly static and the access for TDBM cached data is repetitive. In other environments, where directory data is updated frequently and the access for non-cached data is random, the power and scale of DB2 is used to enhance performance.

The following is included in this section:
- DB2 tuning to improve database access
- TDBM tuning that affects DB2 usage

DB2 tuning is important to ensure that TDBM requests that access the database in DB2 operate efficiently, and that response times do not increase as the database grows in size. Many general DB2 tuning guidelines are applicable to TDBM databases.

Also, there are choices in the initial setup of the TDBM database and in the TDBM backend section of the LDAP server configuration file that influence performance within DB2.
TDBM caches provide a significant benefit to performance, allowing the server to bypass read operations to the database. Optimizing the cache size is important to ensure a high percentage hit rate, without requiring excessive storage. See [LDAP server cache tuning](#) for more information about TDBM caches.

**DB2 tuning**

The following tasks relating to DB2 tuning are crucial to maintaining good performance. These tasks are typically performed by Database Administrators on most production DB2 data:

- Periodically reorganizing the TDBM database by using the DB2 **REORG** utility
- Periodically maintaining the database statistics by using the DB2 **RUNSTATS** utility
- Allocating DB2 buffer pools large enough to minimize I/O to the TDBM database

The TDBM table spaces and indexes should be reorganized periodically using the DB2 **REORG** utility. This helps to improve database access performance and to reclaim fragmented space.

In addition, the **RUNSTATS** utility should be run periodically to update the DB2 catalog with current statistics for the TDBM database, table spaces, tables, indexes, and partitions. This information is necessary for DB2 to select efficient access paths to the TDBM database. This information is also useful to the Database Administrator for determining when the database should be reorganized. The recommended parameters to specify when using the **RUNSTATS** utility are shown below:

```sql
//SYSIN DD *
RUNSTATS TABLESPACE GLDDB.SEARCHTS REPORT YES
  TABLE (ALL)
  INDEX (GLDSRV.DIR_SEARCHX1 KEYCARD
    FREQVAL NUMCOLS 1 COUNT 100
    FREQVAL NUMCOLS 2 COUNT 100,
    GLDSRV.DIR_SEARCHX2 KEYCARD
    FREQVAL NUMCOLS 1 COUNT 100
    FREQVAL NUMCOLS 2 COUNT 100)
RUNSTATS TABLESPACE GLDDB.ENTRYTS REPORT YES
  TABLE (ALL)
  INDEX (GLDSRV.DIR_ENTRYX1 KEYCARD
    FREQVAL NUMCOLS 1 COUNT 100)
RUNSTATS TABLESPACE GLDDB.DESCTS REPORT YES
  TABLE ALL INDEX ALL KEYCARD
RUNSTATS TABLESPACE GLDDB.LENTRYTS REPORT YES
  TABLE ALL INDEX ALL KEYCARD
RUNSTATS TABLESPACE GLDDB.LATTRTS REPORT YES
  TABLE ALL INDEX ALL KEYCARD
RUNSTATS TABLESPACE GLDDB.MISCTS REPORT YES
  TABLE ALL INDEX ALL KEYCARD
RUNSTATS TABLESPACE GLDDB.REPTS REPORT YES
  TABLE ALL INDEX ALL KEYCARD
/*
```

**Notes:**

1. In the example above, **GLDDB** is the TDBM database name and **GLDSRV** is the user ID used to create the TDBM tables and indexes. **GLDSRV** is the same value used in the LDAP server configuration file for **dbuserid**.

2. In the example above:
   - **GLDSRV.DIR_SEARCHX1** is the index on columns **ATTR_ID, VALUE, EID** of table **GLDSRV.DIR_SEARCH**.
   - **GLDSRV.DIR_SEARCHX2** is the index on columns **EID, ATTR_ID** of table **GLDSRV.DIR_SEARCH**.
   - **GLDSRV.DIR_ENTRYX1** is the index on columns **PEID, EID** of table **GLDSRV.DIR_ENTRY**.

3. In the example above, the values specified for **COUNT** are intended as guidelines and may need to be updated depending on the data in the LDAP TDBM database. These numbers represent a minimum frequency. If there is a large amount of data in a TDBM database and potentially many frequent values, these numbers may need to be increased until the optimum frequency is found.
4. When the LDAP server is started, it examines the statistics recorded by the `RUNSTATS` utility in the DB2 catalog. Informational messages are issued to the server output file detailing the column statistics. In addition, messages are issued if any statistics appear to be insufficient. Refer to your LDAP database definitions in your SPUFI files to correlate the table space and index names in the `RUNSTATS` input with the table and column names that appear in these messages. If you run the DB2 `RUNSTATS` utility after the LDAP server has completed initialization, you can use the LDAP server `REFRESH DB2RUNSTATS` operator modify command to reexamine the statistics.

Many installations populate the z/OS LDAP directory with a large amount of initial data and then gradually grow the directory over time with routine updates and additions. In such cases, it is highly recommended that the `REORG` and `RUNSTATS` utilities be run immediately after the directory is populated with this initial data prior to roll out to production. If the initial population of data is done using an application (as opposed to the `ldif2ds` load utility provided with the z/OS LDAP server), it may be necessary to run `REORG` and `RUNSTATS` one or more times during the process of initially populating the directory. This may be needed to ensure DB2 uses efficient access paths based on the statistics gathered from the database, once it contains a representative amount of information. Without this information, poor access paths may be chosen that cause increasing response times as the size of the database increases, and gradual slowing of the process of populating the directory.

DB2 buffer pool allocations should also be examined to ensure they are sufficient for the LDAP TDBM database. It is often useful to isolate specific TDBM table spaces and indexes to their own buffer pools. In particular, separating the indexes from the table spaces may help ensure that the index buffers remain in the buffer pools. This technique may also help evaluate overall behavior of the LDAP database regarding its buffer pool usage when specific tables and indexes correlate to specific buffer pools. Products such as the DB2 Performance Monitor for z/OS are especially useful for monitoring buffer pool activity.

See [Partitioning DB2 tables for TDBM](#) for more information about increasing TDBM performance for DB2 partitioning if you have a large directory and do a lot of update operations on the directory.

### TDBM database tuning

Several choices may ultimately affect the performance of TDBM when accessing its data within DB2:

- The LOCKSIZE chosen on the TDBM tablespaces can be important if you perform many database updates. The default LOCKSIZE of ANY is generally preferred, and is typically sufficient if you perform mostly query activity and low volumes of updates to the database. This generally results in PAGE locking, that causes locking of rows for directory entries other than the one being updated. However, if you have high volumes of update activity, you may experience DB2 deadlocks in the TDBM database with PAGE locking. If this occurs, you may want to set LOCKSIZE ROW on the TDBM tablespaces that contains the `DIR_ENTRY` and the `DIR_SEARCH` table.

- The size of the `DN_TRUNC` column of the `DIR_ENTRY` table specified at database creation time. The `DN_TRUNC` column is used to index data in the `DIR_ENTRY` table to speed up retrieval of directory entries by way of their distinguished name (DN). This column holds the leading portion of each DN, and should be defined long enough to make most values unique. Some applications generate directory entries where the leading portion of the DN is identical. For example, Tivoli Access Manager (TAM) generates entries under each user entry in the namespace where the DN starts with `cn=secPolicyData,secAuthority=Default,”`. To provide uniqueness, it is recommended that installations using TAM with the z/OS LDAP server, define the `DN_TRUNC` column to be 64 bytes in length.

  You should define this column at its proper length during initial set up of the directory. Changing the size requires the `DIR_ENTRY` table to be redefined, and the directory must be unloaded and reloaded to implement the change.

- The size of the `VALUE` column of the `DIR_SEARCH` table specified at database creation time. The `VALUE` column is used to index data in the `DIR_SEARCH` table to speed up retrieval of directory entries for search requests using the search filter values. This column holds the leading portion of textual attribute values, and should be defined long enough to accommodate most values specified in...
search filters. However, this column should not be made significantly larger than required, since this may cause the DIR_SEARCH table and its index to substantially increase in size.
You should define this column at its proper length during initial set up of the directory. Changing the size requires the DIR_SEARCH table to be redefined, and the directory must be unloaded and reloaded to implement the change.

- The attrOverflowSize value specified in the TDBM section of the LDAP server configuration file.
  This configuration option specifies the threshold size of attribute values that are stored separately from the DIR_ENTRY data and are instead stored in the DIR_LONGATTR overflow table.
  This option can avoid unnecessarily reading this overflow data for searches that do not request the attribute. For example, if your directory entries contain JPEG data, but many searches ask for specific attributes and omit the large JPEG attribute from those requested, this option can help avoid reading unnecessary data from the database.
  This option value should be specified large enough so that data that is typically retrieved with the entry remains in the DIR_ENTRY data. Note that entries with overflow data are not eligible for the entry cache, so making this option value too small can impact search performance.

Monitoring performance with cn=monitor

You can retrieve statistics from the server by issuing a search request with a search base of cn=monitor and a filter of (objectclass=*). These are the only values accepted for search base and filter on the monitor search. However, any of the possible scope values are accepted.

The z/OS LDAP server presents monitor data in multiple entries:

- Server-wide statistics are contained in an entry whose distinguished name is cn=monitor.
- Each configured backend has statistics contained in its own entry named cn=backendXXXX,cn=monitor, where XXXX is the backend name specified on the database configuration option in the server configuration file. If no backend name is specified on the database configuration option, the LDAP server generates a name. The naming contexts pertaining to the specific backend are also included in the entry to identify which server backend is being reported.
- Several entries contain statistics for backends that are created by the LDAP server:
  - cn=backendMonitor,cn=monitor - Statistics for the backend handling cn=monitor searches
  - cn=backendSchema,cn=monitor - Statistics for the backend managing the schema
  - cn=backendRootDSE,cn=monitor - Statistics for the backend handling root DSE searches
- If the operations monitor is on (the operationsMonitorSize configuration option is not set to zero), the cn=operations,cn=monitor entry contains statistics on search patterns.

For a scope of:

base Only the cn=monitor entry is returned containing server-wide statistics
one (one-level search) All backend-specific entries are returned and the operations monitor entry is returned (if configured)
sub (subtree search) All entries are returned, including the operations monitor entry (if configured).

The statistics reported on the cn=monitor subtree search can also be displayed by using the LDAP server DISPLAY operator modify command. Operations monitor statistics cannot be displayed by using the DISPLAY operator command. The command is:
f dssrv,display monitor

See the description of the DISPLAY MONITOR output in Displaying performance information and server settings for details.
Statistics generally reflect data gathered since the LDAP server was started. However, many of the
counters can be reset by using the LDAP server RESET operator modify command. The command is:

```
f dssrv,reset monitor
```

In this case, the values reflect data gathered since the last reset.

The monitor search returns the following attributes:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>currentconnections</td>
<td>Current number of client connections</td>
</tr>
<tr>
<td>currenttime</td>
<td>Current date and time on the server</td>
</tr>
<tr>
<td>livethreads</td>
<td>Configured number of communication threads (commThreads)</td>
</tr>
<tr>
<td>maxconnections</td>
<td>Configured maximum number of connections (maxConnections)</td>
</tr>
<tr>
<td>maxreachedconnections</td>
<td>High water mark for concurrent client connections</td>
</tr>
<tr>
<td>resets</td>
<td>Number of times statistics were reset</td>
</tr>
<tr>
<td>resettetime</td>
<td>Date and time statistics were last reset</td>
</tr>
<tr>
<td>starttime</td>
<td>Date and time the server was started</td>
</tr>
<tr>
<td>sysmaxconnections</td>
<td>System defined maximum number of connections</td>
</tr>
<tr>
<td>totalconnections</td>
<td>Number of client connections made to the server</td>
</tr>
<tr>
<td>version</td>
<td>Version of the LDAP server</td>
</tr>
</tbody>
</table>

The statistics reported for the maxconnections, sysmaxconnections, totalconnections,
currentconnections, and maxreachedconnections attribute values only contain information for network
connections. PC connection statistics are not included in these attribute values.

The sysmaxconnections value may be lower than the maxconnections value because of system limits.
If the value for the maxConnections configuration option is not valid, the maxconnections attribute value
on cn=monitor search reflects the system maximum connection limit. For information about how the
maximum number of client connections is set in the LDAP server, see the maxConnections configuration
option at 90.

When statistics are reset, resettetime is set to the value of currenttime, resets is incremented, and
maxreachedconnections is set to the value of currentconnections. None of the other server statistics
listed above are affected by a reset.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>abandonsrequested</td>
<td>Number of abandon operations requested</td>
</tr>
<tr>
<td>abandonscompleted</td>
<td>Number of abandon operations completed</td>
</tr>
<tr>
<td>addsrequested</td>
<td>Number of add operations requested</td>
</tr>
<tr>
<td>addscompleted</td>
<td>Number of add operations completed</td>
</tr>
<tr>
<td>bindsrequested</td>
<td>Number of bind operations requested</td>
</tr>
<tr>
<td>bindscompleted</td>
<td>Number of bind operations completed</td>
</tr>
<tr>
<td>bytessent</td>
<td>Number of bytes of data sent</td>
</tr>
<tr>
<td>comparesrequested</td>
<td>Number of compare operations requested</td>
</tr>
<tr>
<td>comparescompleted</td>
<td>Number of compare operations completed</td>
</tr>
<tr>
<td>deletesrequested</td>
<td>Number of delete operations requested</td>
</tr>
<tr>
<td>deletestcomplited</td>
<td>Number of delete operations completed</td>
</tr>
<tr>
<td>entriessent</td>
<td>Number of search entries sent</td>
</tr>
<tr>
<td>extopsrequested</td>
<td>Number of extended operations requested</td>
</tr>
<tr>
<td>extopscompleted</td>
<td>Number of extended operations completed</td>
</tr>
<tr>
<td>modifiesrequested</td>
<td>Number of modify operations requested</td>
</tr>
<tr>
<td>modifiescompleted</td>
<td>Number of modify operations completed</td>
</tr>
<tr>
<td>modifydnsrequested</td>
<td>Number of modifyDn operations requested</td>
</tr>
<tr>
<td>modifydnsocompleted</td>
<td>Number of modifyDn operations completed</td>
</tr>
<tr>
<td>opscompleted</td>
<td>Number of operations completed</td>
</tr>
</tbody>
</table>
Table 77. Server and backend specific statistics (continued)

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>opsinitiated</td>
<td>Number of operations initiated</td>
</tr>
<tr>
<td>searchreferencessent</td>
<td>Number of search references sent</td>
</tr>
<tr>
<td>searchesrequested</td>
<td>Number of search operations requested</td>
</tr>
<tr>
<td>searchescompleted</td>
<td>Number of search operations completed</td>
</tr>
<tr>
<td>unbindsrequested</td>
<td>Number of unbind operations requested</td>
</tr>
<tr>
<td>unbindscompleted</td>
<td>Number of unbind operations completed</td>
</tr>
<tr>
<td>unknownopsrequested</td>
<td>Number of unrecognized operations requested</td>
</tr>
<tr>
<td>unknownopscompleted</td>
<td>Number of unrecognized operations completed</td>
</tr>
</tbody>
</table>

When statistics are reset, all of the server and backend specific statistics listed above are set to zero.

Table 78. Backend specific statistics

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>acl_source_cache_size</td>
<td>Configured maximum size (in entries) of the ACL Source cache</td>
</tr>
<tr>
<td>acl_source_cache_current</td>
<td>Current size (in entries) of the ACL Source cache</td>
</tr>
<tr>
<td>acl_source_cache_hit</td>
<td>Number of lookups that have hit the ACL Source cache</td>
</tr>
<tr>
<td>acl_source_cache_miss</td>
<td>Number of lookups that have missed the ACL Source cache</td>
</tr>
<tr>
<td>acl_source_cache_percent_hit</td>
<td>Percent of lookups that have hit the ACL Source cache</td>
</tr>
<tr>
<td>acl_source_cache_refresh</td>
<td>Number of times the ACL Source cache was invalidated</td>
</tr>
<tr>
<td>acl_source_cache_refresh_avgsize</td>
<td>Average number of entries in the ACL Source cache at invalidation</td>
</tr>
<tr>
<td>dn_cache_size</td>
<td>Configured maximum size (in entries) of the DN cache</td>
</tr>
<tr>
<td>dn_cache_current</td>
<td>Current size (in entries) of the DN cache</td>
</tr>
<tr>
<td>dn_cache_hit</td>
<td>Number of lookups that have hit the DN cache</td>
</tr>
<tr>
<td>dn_cache_miss</td>
<td>Number of lookups that have missed the DN cache</td>
</tr>
<tr>
<td>dn_cache_percent_hit</td>
<td>Percent of lookups that have hit the DN cache</td>
</tr>
<tr>
<td>dn_cache_refresh</td>
<td>Number of times the DN cache was invalidated</td>
</tr>
<tr>
<td>dn_cache_refresh_avgsize</td>
<td>Average number of entries in the DN cache at invalidation</td>
</tr>
<tr>
<td>dn_to_eid_cache_size</td>
<td>Configured maximum size (in entries) of the DN to Entry ID cache</td>
</tr>
<tr>
<td>dn_to_eid_cache_current</td>
<td>Current size (in entries) of the DN to Entry ID cache</td>
</tr>
<tr>
<td>dn_to_eid_cache_hit</td>
<td>Number of lookups that have hit the DN to Entry ID cache</td>
</tr>
<tr>
<td>dn_to_eid_cache_miss</td>
<td>Number of lookups that have missed the DN to Entry ID cache</td>
</tr>
<tr>
<td>dn_to_eid_cache_percent_hit</td>
<td>Percent of lookups that have hit the DN to Entry ID cache</td>
</tr>
<tr>
<td>dn_to_eid_cache_refresh</td>
<td>Number of times the DN to Entry ID cache was invalidated</td>
</tr>
<tr>
<td>dn_to_eid_cache_refresh_avgsize</td>
<td>Average number of entries in the DN to Entry ID cache at invalidation</td>
</tr>
<tr>
<td>entry_cache_size</td>
<td>Configured maximum size (in entries) of the Entry cache</td>
</tr>
<tr>
<td>entry_cache_current</td>
<td>Current size (in entries) of the Entry cache</td>
</tr>
<tr>
<td>entry_cache_hit</td>
<td>Number of lookups that have hit the Entry cache</td>
</tr>
<tr>
<td>entry_cache_miss</td>
<td>Number of lookups that have missed the Entry cache</td>
</tr>
<tr>
<td>entry_cache_percent_hit</td>
<td>Percent of lookups that have hit the Entry cache</td>
</tr>
<tr>
<td>entry_cache_refresh</td>
<td>Number of times the Entry cache was invalidated</td>
</tr>
<tr>
<td>entry_cache_refresh_avgsize</td>
<td>Average number of entries in the Entry cache at invalidation</td>
</tr>
<tr>
<td>entry_owner_cache_size</td>
<td>Configured maximum size (in entries) of the Entry Owner cache</td>
</tr>
<tr>
<td>entry_owner_cache_current</td>
<td>Current size (in entries) of the Entry Owner cache</td>
</tr>
<tr>
<td>entry_owner_cache_hit</td>
<td>Number of lookups that have hit the Entry Owner cache</td>
</tr>
<tr>
<td>entry_owner_cache_miss</td>
<td>Number of lookups that have missed the Entry Owner cache</td>
</tr>
<tr>
<td>entry_owner_cache_percent_hit</td>
<td>Percent of lookups that have hit the Entry Owner cache</td>
</tr>
<tr>
<td>entry_owner_cache_refresh</td>
<td>Number of times the Entry Owner cache was invalidated</td>
</tr>
<tr>
<td>entry_owner_cache_refresh_avgsize</td>
<td>Average number of entries in the Entry Owner cache at invalidation</td>
</tr>
</tbody>
</table>
Table 78. Backend specific statistics (continued)

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>filter_cache_size</td>
<td>Configured maximum size (in entries) of the Filter cache</td>
</tr>
<tr>
<td>filter_cache_current</td>
<td>Current size (in entries) of the Filter cache</td>
</tr>
<tr>
<td>filter_cache_hit</td>
<td>Number of lookups that have hit the Filter cache</td>
</tr>
<tr>
<td>filter_cache_miss</td>
<td>Percent of lookups that have hit the Filter cache</td>
</tr>
<tr>
<td>filter_cache_percent_hit</td>
<td>Percent of lookups that have hit the Filter cache</td>
</tr>
<tr>
<td>filter_cache_refresh</td>
<td>Number of times the Filter cache was invalidated</td>
</tr>
<tr>
<td>filter_cache_refresh_avgsize</td>
<td>Average number of entries in the Filter cache at invalidation</td>
</tr>
<tr>
<td>filter_cache_bypass_limit</td>
<td>Configured Filter cache bypass limit (filterCacheBypassLimit)</td>
</tr>
<tr>
<td>namingcontexts</td>
<td>Suffixes managed by this backend</td>
</tr>
</tbody>
</table>

Note that not all cache statistics shown above appears for each backend. A backend reports statistics for those caches that it supports. The schema backend reports dn_cache statistics. The LDBM and CDBM backends report filter_cache statistics. A TDBM backend reports statistics for all caches except the dn_cache. A DB2-based GDBM backend reports statistics for all caches except the dn_cache, while a file-based GDBM backend only reports filter_cache statistics.

When statistics are reset, the cache_hit, cache_miss, cache_percent_hit, cache_refresh, and cache_refresh_avgsize for each cache are reset to zero. Resetting the statistics has no effect on the cache_size for each cache, nor on the filter_cache_bypass_limit, since these are configured values. Resetting the statistics also has no effect on the cache_current for each cache, since the contents of the caches are not altered by a reset of statistics. Some caches may get invalidated and refreshed because of directory update operations. When this occurs, cache_refresh is incremented and cache_current is set to zero to reflect the refreshed (empty) cache. The cache_hit, cache_miss, and values cache_percent_hit are accumulated across cache invalidation and refresh until a RESET MONITOR command is issued or the server ends.

Table 79. Operations monitor statistics

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cachesize</td>
<td>Configured maximum number of search patterns in the operations monitor</td>
</tr>
<tr>
<td>currenttimestamp</td>
<td>Current date and time in Zulu time stamp format</td>
</tr>
<tr>
<td>entries</td>
<td>Total number of search patterns in the operations monitor entry</td>
</tr>
<tr>
<td>numtrimmed</td>
<td>Number of search patterns trimmed from the operations monitor</td>
</tr>
<tr>
<td>resets</td>
<td>Number of times the operations monitor statistics were reset</td>
</tr>
<tr>
<td>resettimestamp</td>
<td>Date and time in Zulu time stamp format of last reset or server start up if the reset command was never issued</td>
</tr>
<tr>
<td>searchStats</td>
<td>Search statistics for search patterns based on the search parameters</td>
</tr>
<tr>
<td>searchIPStats</td>
<td>Search statistics for search patterns consisting of the same elements as the searchStats pattern, but also including the client IP address</td>
</tr>
</tbody>
</table>

When statistics are reset, resetTimestamp is set to currentTimestamp, resets is incremented by one, entries is set to zero, numtrimmed is set to zero, and all search patterns are deleted.

The Zulu time stamp format used in the currenttimestamp and resettimestamp attribute values is: yyyyMMddhhmiss.uuuuuuZ

Where,

yyyy is year, mm is month, dd is day, hh is hour, ii is minutes, ss is seconds, uuuuuu is microseconds, Z is a character constant meaning that this time is based on Zulu time, also known as GMT.
The **searchIPStats** and **searchStats** attribute values contain search rates and other search activity that are being monitored. Depending upon the LDAP server configuration, there can be **searchIPStats** and **searchStats** attribute values returned in the **cn=operations,cn=monitor** entry for each search executed against the LDAP server. The **searchStats** attribute values contain the total of all data collected for all searches matching this search pattern no matter the client's IP address.

The format of the **searchIPStats** and **searchStats** attribute values is:

```
rate=rate,maxRate=maxRate,maxRateTimeStamp=maxRateTimeStamp,
createTimeStamp=createTimeStamp,ID=opid
```

The following describes the LDAP search pattern parts:

- **attributes**
  List of attributes to be returned.

- **avg**
  Average elapsed time for each occurrence of search pattern in microseconds.

- **baseDN**
  Distinguished name of the base of the search, with \_v substituted for attribute values.

- **clientIP**
  Client IP address (omitted for **searchStats** search patterns).

- **createTimeStamp**
  Date and time this search pattern was first added, in Zulu time stamp format.

- **filter-string**
  Search filter with substitutions for literal attribute values. Excluding the \* character, all strings in values are substituted with \_v. For example: (cn=*bob*bah*) would be (cn=_v_v_v_*). There is no substitution on **objectclass** equality values when the **objectclass** is defined in the schema.

- **maxRate**
  The highest rate on this entry.

- **maxRateTimeStamp**
  Date and time **maxRate** was last set, in Zulu time stamp format.

- **numOps**
  Total number of times this search pattern has occurred.

- **opid**
  A unique integer value that distinguishes each operations monitor search pattern.

- **rate**
  Number of search operations processed in the previous one minute interval. Starting with server startup or the last reset command, rate is recalculated for each search pattern every 60 seconds.

- **scope**
  **base** for base object searches, **one** for one-level searches, and **sub** for subtree searches.

- **status**
  **success** for any search operation that results in return code LDAP_SUCCESS, LDAP_PARTIAL_RESULTS, or LDAP_REFERRAL. Any other return codes result in status being set to **failure**.

See Table 79 on page 500 for the time stamp format.

In addition to the above syntax, the following character escaping is performed:
- **comma** = %2C
- **percent** = %25
- **question mark** = %3F
- **space** = %20

**Note:** The comma, percent, and question mark characters are not escaped when they are used as metacharacters in the search pattern.
For information about monitoring performance with the LDAP server DISPLAY MONITOR operator command, see Displaying performance information and server settings.

**Note:** DISPLAY MONITOR output does not display *cn=operations,cn=monitor* data.

**Monitor search examples**

Following is an example of a monitor search using scope=base. This returns only statistics related to the entire server:

```
ldapsearch -h ldaphost -p ldapport -b cn=monitor -s base objectclass=* 
```

```
version=z/OS Version 1 Release 11 IBM Tivoli Directory Server  
livethreads=10  
maxconnections=24982  
sysmaxconnections=25000  
totalconnections=20709  
currentconnections=1  
maxreachedconnections=15  
opsinitiated=62126  
opscompleted=62125  
abandonsrequested=0  
abandonscompleted=0  
addsrequested=2318  
addscompleted=2318  
bindsrequested=20709  
bindscompleted=20709  
comparestrequested=0  
comparestcompleted=0  
deletesrequested=2228  
deletescompleted=2228  
extopsrequested=0  
extopscompleted=0  
modifiesrequested=11501  
modifiescompleted=11501  
modifydnrequested=440  
modifydncompleted=440  
searchesrequested=4222  
searchescompleted=4221  
unbindstrequested=20708  
unbindstcompleted=20708  
unknownopsrequested=0  
unknownopscompleted=0  
entriessent=4221  
bytessent=1564656734  
searchreferencessent=0  
currenttime=Thu Sep 25 16:33:00.187846 2008  
resets=0
```

Following is an example of output of a monitor search with scope=one for a server configured with TDBM and LDBM backends. This returns backend specific statistics and operations monitor statistics. The cache statistics shown would only be included for TDBM, LDBM, GDBM, CDBM, and schema backends, since the other backend types do not implement caches. Operations monitor statistics are included for all backends.

```
Note that not all operational statistics for each backend are shown in the example below. They have been omitted from the example only, and appear in full for a *cn=monitor* search.
```

```
ldapsearch -L -h ldaphost -p ldapport -b cn=monitor -s one objectclass=* 
```

```
502  z/OS V1R11.0 IBM Tivoli Directory Server Administration and Use for z/OS
dn: cn=backendMyTDBM,cn=monitor
namingcontexts: C=CA
namingcontexts: C=TDBM
namingcontexts: CN=MOVER
namingcontexts: CN=MOVING
opsinitiated: 3013
opscompleted: 3013
abandonsrequested: 0
abandonscompleted: 0
addsrrequested: 380
addsccompleted: 380
bindsrequested: 0
bindscompleted: 0
compareresrequested: 0
comparerescompleted: 0
deletesrequested: 365
deletescompleted: 365
extopsrequested: 0
extopscompleted: 0
modifiesrequested: 1645
modifiescompleted: 1645
modifydnsrequested: 63
modifydnscompleted: 63
searchesrequested: 560
searchescompleted: 560
unbindsrequested: 0
unbindscompleted: 0
unknownopsrequested: 0
unknownopscompleted: 0
entriessent: 560
bytessent: 105692
searchreferencessent: 0
acl_source_cache_size: 100
acl_source_cache_current: 1
acl_source_cache_hit: 3012
acl_source_cache_miss: 1
acl_source_cache_percent_hit: 99.97%
acl_source_cache_refresh: 0
acl_source_cache_refresh_avgsize: 0
dn_to_eid_cache_size: 1000
dn_to_eid_cache_current: 555
dn_to_eid_cache_hit: 195263
dn_to_eid_cache_miss: 4035
dn_to_eid_cache_percent_hit: 97.98%
dn_to_eid_cache_refresh: 0
dn_to_eid_cache_refresh_avgsize: 0
entry_cache_size: 5000
entry_cache_current: 562
entry_cache_hit: 381420
entry_cache_miss: 1259
entry_cache_percent_hit: 99.67%
entry_cache_refresh: 0
entry_cache_refresh_avgsize: 0
entry_owner_cache_size: 100
entry_owner_cache_current: 1
entry_owner_cache_hit: 3012
entry_owner_cache_miss: 1
entry_owner_cache_percent_hit: 99.97%
entry_owner_cache_refresh: 0
entry_owner_cache_refresh_avgsize: 0
filter_cache_size: 5000
filter_cache_current: 0
filter_cache_hit: 0
filter_cache_miss: 0
filter_cache_percent_hit: 0.00%
filter_cache_refresh: 2446
filter_cache_refresh_avgsize: 0
filter_cache_bypass_limit: 100

dn: cn=backendLDBM-002,cn=monitor
namingcontexts: C=AU
namingcontexts: C=LDBM
searchreferencessent: 0
filter_cache_size: 5000
filter_cache_current: 0
filter_cache_hit: 0
filter_cache_miss: 0
filter_cache_percent_hit: 0.00%
filter_cache_refresh: 16487
filter_cache_refresh_avgsize: 0
filter_cache_bypass_limit: 100

dn: cn=backendMonitor,cn=monitor
namingcontexts: CN=MONITOR

searchreferencessent: 0
dn_cache_size: 1000
dn_cache_current: 1000
dn_cache_hit: 123743
dn_cache_miss: 22017
dn_cache_percent_hit: 84.90%
dn_cache_refresh: 0
dn_cache_refresh_avgsize: 0

dn: cn=backendRootDSE,cn=monitor
searchStats: ldap://9.12.47.208/OU=_v,O=_v,C=_v?telephoneNumber,postalAddress,mail,uid?one?(objectclass/inetOrgPerson)?failure,numOps=51,avg=230,rate=32,maxRate=32,
maxRateTimeStamps=20080313132741.415477Z,createTimeStamps=20080313132628.361618Z,ID=2737
searchIPStats: ldap://127.0.0.1/RACFGROUPID=_v+RACFUSERID=_v,PROFILETYPE=_v,CN=_v?racfconnectowner,racfconnectgroupauthority,racfconnectgroupuacc?base?(objectclass=*)?success,numOps=4,avg=240,rate=0,maxRate=4,maxRateTimeStamps=20080313132628.361618Z,createTimeStamps=20080313132628.878552Z,ID=2740
searchStats: ldap://9.12.47.208/OU=_v,O=_v,C=_v??sub?(|(&(sn=_v)(cn=_v*))?(description=*_v*)|)success,numOps=42,avg=246,rate=5,maxRate=37,maxRateTimeStamps=20080313132628.878552Z,createTimeStamps=20080313132628.361618Z,ID=2738
searchStats: ldap://9.12.47.208/OU=_v,O=_v,C=_v?telephoneNumber,postalAddress,mail,uid?one?(objectclass/inetOrgPerson)?failure,numOps=51,avg=230,rate=32,maxRate=32,
maxRateTimeStamps=20080313132741.415477Z,createTimeStamps=20080313132628.361618Z,ID=2740
searchIPStats: ldap://fe00::f4f7:0:0:7442:750f/OU=_v,O=_v??sub?(|(&(sn=_v)(cn=_v*))?(description=*_v*)|)success,numOps=42,avg=246,rate=5,maxRate=37,maxRateTimeStamps=20080313132628.878552Z,createTimeStamps=20080313132628.361618Z,ID=2741
searchIPStats: ldap://fe00::f4f7:0:0:7442:750f/OU=_v,O=_v??sub?(|(&(sn=_v)(cn=_v*))?(description=*_v*)|)success,numOps=42,avg=246,rate=5,maxRate=37,maxRateTimeStamps=20080313132628.878552Z,createTimeStamps=20080313132628.361618Z,ID=2741
Large access groups considerations

Users with large access groups in z/OS LDAP may experience performance problems and increased storage usage in the LDAP server as access groups grow in size. Tivoli Access Manager (TAM) users are susceptible to this.

TAM users often create access groups in LDAP containing many members with every user in the registry defined in one large access group. The performance impacts may worsen as the registry grows, with any of the following symptoms:

- Increased response time in the application
- Increased processor utilization in the LDAP server
- Increased storage requirements in the LDAP server
- Increased resource consumption in DB2 (logging, I/O, processor usage, buffer pool demands)

Some scenarios that require substantial amounts of processing and storage within the z/OS LDAP server, are:

- A search operation that returns all the members of a large access group. This includes either a search that returns the many values with the member or uniqueMember attribute, or a search that returns the many values in the ibm-allMembers operational attribute.
- A search operation that requests all the members of a large access group, but the members are not returned because ACL read permissions prevent the requester from seeing the data.
- Update requests that touch a large access group entry when persistentSearch on is configured for a TDBM or LDBM backend that contains the large entry.

These scenarios are also susceptible to the affects of LE HEAPPOOL usage as described below.

The addressability limits of the z/OS LDAP server may become a factor when there are hundreds of thousands or millions of members in a single access group.

In this case, consider the following corrective actions:

- Increase the LDAP server’s region size, if possible.
- Limit the number of members placed within a single access group and partition the users into separate access groups. The number of members for each access group that can be managed successfully depends on many factors, such as the size of the member values, the amount of region defined for the z/OS LDAP server, and the level of concurrent activity within the server.
- If possible, avoid configuring persistentSearch on for a TDBM or LDBM backend that contains large entries. Some applications that exploit persistent search may only do so with the changelog, and only need persistentSearch on configured for the GDBM backend.

LE heap pools considerations

By default, the z/OS LDAP server uses LE heap pools to improve performance. This facility reduces the processor consumption and allows better parallelism of concurrent requests within the z/OS LDAP server. However, overall storage consumption is typically larger with the use of LE heap pools as compared to running without the facility enabled. Also, once storage is allocated to a given LE heap pool, it remains
allocated to that heap pool and can only be used for future storage requests that are eligible (based on size) for the given heap pool. For example, when the z/OS LDAP server must process a large access group entry in storage, the following may occur:

- While the request is processing, the z/OS LDAP server may use all available storage in its address space, causing a failure of the request, a failure of other concurrent requests, or a failure and abnormal termination of the server.
- Because of the sudden, large demand for storage to process the large group, most or all of the storage available to the z/OS LDAP server may be allocated and reserved to specific heap pools. Although the z/OS LDAP server may appear to be available and able to process a variety of requests, many subsequent requests may fail because of insufficient storage, particularly those for entries with large or numerous attributes. In the absence of any failures, this large increase in storage use by the z/OS LDAP server may be detectable by system resource monitoring products, such as the Resource Measurement Facility (RMF™).

If these problems occur, consider either tuning the heap pool sizes or disabling the heap pools for the z/OS LDAP server.

Tuning the heap pool sizes optimizes storage usage for the data within the LDAP server. See [z/OS Language Environment Programming Guide](https://www.ibm.com/support/knowledgecenter/STXKQY_11.1.0/com.ibm.zos.zosle.doc/guides/guides/zz7014988z17b.html) for details on how to tune the heap pool settings. Note that the procedure for tuning heap pool settings requires a controlled environment with representative workloads. In this case, the workload should include the scenarios described earlier that cause the large demands for storage. Note that it is recommended that the storage reports needed for the tuning procedure be gathered in a non-production environment because tracking the storage statistics significantly impacts performance.

Disabling heap pools reduces the total heap storage requirements of the LDAP server, at the cost of increased processing.

Overriding the heap pool settings for the LDAP server can be done by specifying the LE run-time option ‘HEAPPOOLS’ when running in AMODE 31, or ‘HEAPPOOLS64’ in AMODE 64. These options can be specified in the PARM field on the EXEC statement or within a dataset specified on a CEEOPTS DD statement in the LDAP server JCL. For more details on setting this parameter, see [z/OS Language Environment Programming Reference](https://www.ibm.com/support/knowledgecenter/STXKQY_11.1.0/com.ibm.zos.zosle.doc/guides/ref/guides/zz7014989z17b.html) and [z/OS Language Environment Programming Guide](https://www.ibm.com/support/knowledgecenter/STXKQY_11.1.0/com.ibm.zos.zosle.doc/guides/guides/zz7014988z17b.html).

---

**GDBM (Changelog) performance considerations**

The GDBM database is used only for the changelog function. By its very nature, this function tends to have a high intensity of update activity compared to read activity. Since update activity is generally more costly than read activity, this function should only be enabled when its use is actually needed.

GDBM can be configured to store its entries in DB2 (like TDBM) or in files (like LDBM). Most of the performance considerations for GDBM are identical to those of TDBM or LDBM, depending on which configuration you choose. For example, if you choose a DB2-based GDBM, DB2 tuning and cache tuning are important. If you choose a file-based GDBM, storage, DASD space, initialization elapsed time, and database commit settings are important, especially if you allow the changelog to contain a large number of entries. However, the following GDBM-specific differences should be noted as compared to TDBM or LDBM:

- The distinguished names (DNs) of entries and the searchable attributes within entries in GDBM tend to be well bounded in size and content. Therefore, when you configure GDBM to be DB2-based, the default sizes for the `DN_TRUNC` column in the `DIR_ENTRY` table and the `VALUE` column in the `DIR_SEARCH` table do not require adjustment.
- Since most GDBM requests are update operations, the search filter cache is disabled by default. You may enable the cache, if desired, but if this is done, it is recommended that the cache is monitored to ensure it is providing a benefit. Note that the entry cache is not implemented in file-based backends.
When the `changeLogMaxAge` or `changeLogMaxEntries` option is specified in the GDBM section of the LDAP server configuration file, the change log is periodically trimmed, based on the limits set in the configuration file. For more information about these configuration options, see Configuration file options.

Since GDBM generally experiences high volumes of update activity, if you are using DB2-based GDBM, consider setting `LOCKSIZE ROW` on the table spaces that contain the `DIR_ENTRY` and the `DIR_SEARCH` table to help avoid DB2 deadlocks.

### SDBM performance considerations

The z/OS LDAP server SDBM backend allows access to the RACF database. Most tuning that affects performance in this area is within the RACF product. Refer to [z/OS MVS Initialization and Tuning Guide](#) for more information about tuning RACF.

Also, see [SDBM operational behavior](#) for details regarding different types of LDAP requests supported, and the RACF operations issued by these requests. This information can also be helpful when assessing RACF tuning considerations.

When writing applications that only require authentication to the SDBM backend by using LDAP bind requests, performance can be improved by specifying the `authenticateOnly` control on the bind request within the application. See `authenticateOnly` for more information.
Chapter 30. LDAP server messages

This part contains the messages returned by the LDAP server. The messages are in alphanumeric order.

**LDAP server and ldif2ds messages (1000)**

<table>
<thead>
<tr>
<th>Code</th>
<th>Message Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLD1001I</td>
<td>LDAP server version version.release, Service level level, Build date date, Time time.</td>
</tr>
</tbody>
</table>

**Explanation:** The LDAP server with version, release, service level, build date, and build time indicated in the message is running.

In the message text:
- **version**
  - Server version
- **release**
  - Server release
- **level**
  - Server service level
- **date**
  - Server build date
- **time**
  - Server build time

**Example:** None.

**System action:** The program continues.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** None.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

| GLD1002I   | LDAP runtime version version.release, Service level level, Build date date, Time time. |

**Explanation:** The LDAP runtime with version, release, service level, build date, and build time indicated in the message is running.

In the message text:
- **version**
  - Runtime version
- **release**
  - Runtime release
- **level**
  - Runtime service level
- **date**
  - Runtime build date
- **time**
  - Runtime build time

**Example:** None.

**System action:** The program continues.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** None.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.
GLD1004I LDAP server is ready for requests.
Explanation: The LDAP server has started and is ready for requests.
Example: None.
System action: The program continues.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1005I LDAP server start command processed.
Explanation: The LDAP server has processed the START command.
Example: None.
System action: The program continues.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1007I LDAP server is stopping.
Explanation: The LDAP server is stopping.
Example: None.
System action: The program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1008E Unable to allocate storage.
Explanation: The LDAP server or utility is unable to allocate the necessary storage to continue processing the request.
Example: None.
System action: The program ends.
Operator response: Increase the storage available for use by the LDAP server or utility. Then restart the program. If the problem persists, contact the service representative.
User response: None.
System programmer response: None.
Administrator response: If running the 31-bit LDAP server (GLDSRV31), consider using the 64-bit LDAP server (GLDSRV64) to utilize the additional storage available in a 64-bit address space. Then restart the program.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
GLD1009A LDAP server is not APF-authorized.

Explanation: The LDAP server is not running with APF authorization. The PDS which contains the LDAP server, SYS1.SIEALNKE, and the PDSs containing all the DLLs that the LDAP server loads must be APF-authorized to allow the LDAP server to make the necessary program control threading calls.

Example: None.

System action: The program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Add SYS1.SIEALNKE to the list of APF-authorized datasets. If using a JOBLIB or STEPLIB for the LDAP server started task, verify that all datasets in the concatenation are also APF-authorized. Then restart the program.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1010A Unable to make address space non-swappable: Error error_code.

Explanation: The LDAP server is unable to make its address space non-swappable. Refer to the description of SYSEVENT in z/OS MVS Programming: Authorized Assembler Services Reference SET-WTC for more information on the error. The LDAP server must be non-swappable in order to support system-level program calls. This capability is required when the LDAP server supports RACF change logging or Policy Directory extended operations.

In the message text:

error_code

Error code from SYSEVENT

Example: None.

System action: The program ends.

Operator response: Contact the LDAP Administrator or see the Administrator response.

User response: None.

System programmer response: None.

Administrator response: Use the information in the message to correct the error. Then restart the program.

If the problem persists, contact the service representative.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1011E Unable to register for restart: Error error_code, Reason 0xreason_code.

Explanation: The LDAP server is unable to register with ARM (Automatic Restart Management). Refer to the description of IXCARM in z/OS MVS Programming: Sysplex Services Reference for more information on the error.

In the message text:

error_code

Error code from IXCARM

reason_code

Reason code from IXCARM

Example: None.

System action: The LDAP server continues, but will not be automatically restarted if it fails unexpectedly.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the information in the message to correct the error. Restart the program if ARM support is needed. If the problem persists, contact the service representative.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1012I LDAP server restart registration complete on system system_name.

Explanation: The LDAP server has successfully registered with ARM (Automatic Restart Management) on the system indicated in the message. The LDAP server will be automatically restarted if it fails unexpectedly. It will not be restarted if it detects an error and stops.

In the message text:
GLD1013I LDAP server restarting on system system_name.

Explanation: The LDAP server on the system indicated in the message is being restarted following an unexpected failure. The RESTART_ATTEMPTS value in the ARM policy determines the number of restarts which will be attempted.

In the message text:

system_name
Local system name

Example: None.

System action: The LDAP server continues.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1015I LDAP server restart deregistration complete on system system_name.

Explanation: The LDAP server has successfully deregistered with ARM (Automatic Restart Management) on the system indicated in the message. The LDAP server will no longer be automatically restarted if it fails unexpectedly.

In the message text:

system_name
Local system name

Example: None.

System action: The LDAP server is in the process of stopping and continues with shutdown.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1014E Unable to deregister for restart: Error error_code, Reason 0xreason_code.

Explanation: The LDAP server is unable to deregister with ARM (Automatic Restart Management) during server shutdown. Refer to the description of IXCARM in z/OS MVS Programming: Sysplex Services Reference for more information on the error.

In the message text:

error_code
Error code from IXCARM

reason_code
Reason code from IXCARM

Example: None.

System action: The LDAP server is in the process of stopping and continues with shutdown.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1016E Unable to create mutex:
   error_code/reason_code - error_text

Explanation: The LDAP server or utility is unable to create a mutex. Refer to the description of pthread_mutex_init() in z/OS XL C/C++ Run-Time Library Reference for more information on the error.

In the message text:
   error_code
      Error code from pthread_mutex_init()
   reason_code
      Reason code from pthread_mutex_init()
   error_text
      Error text corresponding to the error code

Example: None.
System action: The program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Use the information in the message to correct the error. Then restart the program. If the problem persists, contact the service representative.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1018A Unable to initialize the directory schema.

Explanation: The LDAP server or utility is unable to initialize the directory schema. A previous message indicates the reason for the failure.

Example: None.
System action: The program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Use the information in the earlier message to correct the error. Then restart the program. If the problem persists, contact the service representative.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1019A Unable to open from_code_page to to_code_page converter:
   error_code/reason_code - error_text

Explanation: The LDAP server or utility is unable to open a code page converter to convert character strings. Refer to the description of iconv_open() in z/OS XL C/C++ Run-Time Library Reference for more information on the error.

In the message text:
   error_code
      Error code from pthread_mutex_init()
   reason_code
      Reason code from pthread_mutex_init()
   error_text
      Error text corresponding to the error code
from_code_page  
   Code page to be converted from

to_code_page  
   Code page to be converted to

error_code  
   Error code from iconv_open()

reason_code  
   Reason code from iconv_open()

text  
   Error text corresponding to the error code

Example:  None.
System action:  The program ends.
Operator response:  None.
User response:  None.
System programmer response:  None.
Administrator response:  Use the information in the message to correct the error. Then restart the program. If the problem persists, contact the service representative.

Problem determination:  Not applicable.
Source:  LDAP
Module:  None.
Routing code:  None.
Descriptor code:  None.
Automation:  Not applicable.

GLD1020E  Unrecognized LDAP server command.

Explanation:  An unrecognized LDAP server operator modify command is detected. The valid LDAP server commands are AUDIT, BACKEND, COMMIT, DEBUG, DISPLAY, LOG, MAINTMODE, REFRESH, RESET, SNAP, and WLMEXCEPT. The SNAP command is available only with the 31-bit LDAP server.

Example:  None.
System action:  The LDAP server ignores the entered command and continues. A new LDAP server operator modify command may be entered.
Operator response:  Issue a valid LDAP server operator modify command.
User response:  None.
System programmer response:  None.
Administrator response:  See Operator response or contact Operator.

Problem determination:  Not applicable.
Source:  LDAP
Module:  None.
Routing code:  None.
Descriptor code:  None.
Automation:  Not applicable.

GLD1021E  Incorrect LDAP server command option specified.

Explanation:  An incorrect command option was found within an LDAP server operator modify command.

Example:  None.
System action:  The LDAP server ignores the entered command and continues. A new LDAP server operator modify command may be entered.
Operator response:  Issue a valid LDAP server operator modify command.
User response:  None.
System programmer response:  None.
Administrator response:  See Operator response or contact Operator.

Problem determination:  Not applicable.
Source:  LDAP
Module:  None.
Routing code:  None.
Descriptor code:  None.
Automation:  Not applicable.

GLD1022I  Debug option processed: debug_level.

Explanation:  The debug level for the LDAP server has been reset using the value indicated in the message.

In the message text:

d debug_level
   Debug level

Example:  None.
System action:  The LDAP server continues. Debug messages corresponding to the updated debug level are now created.
Operator response:  None.
User response:  None.
System programmer response:  None.
Administrator response:  None.

Problem determination:  Not applicable.
Source:  LDAP
Module:  None.
Routing code:  None.
Descriptor code:  None.
GLD1023I  Processing configuration file **filename**.

**Explanation:** The LDAP server or utility is processing the configuration file indicated in the message.

In the message text:

**filename**  
LDAP server configuration file name

**Example:** None.

**System action:** The program continues.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** None.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

GLD1024I  Configuration file **filename** processed.

**Explanation:** The LDAP server or utility has successfully processed the configuration file indicated in the message.

In the message text:

**filename**  
LDAP server configuration file name

**Example:** None.

**System action:** The program continues.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** None.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

GLD1025A  Unable to process command options.

**Explanation:** The LDAP server is unable to process the command-line options. A previous message indicates the reason for the failure.

**Example:** None.

**System action:** The program ends.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Use the information in the earlier message to correct the error. Then restart the program.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

GLD1026E  Incorrect LDAP debug option specified: **debug_options**.

**Explanation:** The value specified for the `-d` parameter on the LDAP server or utility command line is not valid.

In the message text:

**debug_options**  
Debug options

**Example:** None.

**System action:** The program ends.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Refer to the description of the `-d` parameter on the LDAP server or utility command line for more information on the available debug options and how they are specified. Specify valid debug options for the `-d` command parameter. Then restart the program.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.
GLD1027E  parameter is an unrecognized command parameter.

Explanation: The command-line parameter indicated in the message is not supported by the LDAP server or utility.

In the message text:

parameter
  Unrecognized command parameter

Example: None.
System action: The program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Determine the correct command-line parameter to use. Then restart the program.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1028E  No value specified for the option parameter.

Explanation: The command-line parameter indicated in the message cannot be specified without a value when starting the LDAP server or utility. The parameter must have a value.

In the message text:

option
  Command parameter with missing value

Example: None.
System action: The program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Specify a valid value for the command-line parameter. Then restart the program.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1029E  port is not a valid TCP/IP port number.

Explanation: The TCP/IP port number specified for an LDAP server command-line parameter or in the LDAP server configuration file is not valid. The port number must be between 1 and 65535.

In the message text:

port
  TCP/IP port number

Example: None.
System action: The program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Provide a valid TCP/IP port number. Then restart the program.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1030E  Unable to parse LDAP URL url: error_text.

Explanation: The LDAP URL specified for an LDAP server command-line parameter or in the LDAP server configuration file is not valid.

In the message text:

url
  LDAP URL
error_text
  Error message text

Example: None.
System action: The program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Use the information in the message to correct the error. Then restart the program.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
GLD1031A Unable to process the server configuration file.

Explanation: The LDAP server or utility is unable to process the LDAP server configuration file. A previous message indicates the reason for the failure.

Example: None.

System action: The program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.

Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1032E Unable to open configuration file filename: error_code/reason_code - error_text

Explanation: The LDAP server or utility is unable to open the LDAP server configuration file. The file can be a filesystem file or a dataset. Refer to the description of fopen() in z/OS XL C/C++ Run-Time Library Reference for more information on the error.

In the message text:
filename
LDAP server configuration file name
error_code
Error code from fopen()
reason_code
Reason code from fopen()
error_text
Error text corresponding to the error code

Example: None.

System action: The program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.

Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1033E Unable to read configuration file filename: error_code/reason_code - error_text

Explanation: The LDAP server or utility is unable to read the LDAP server configuration file. The file can be a filesystem file or a dataset. Refer to the description of fgets() in z/OS XL C/C++ Run-Time Library Reference for more information on the error.

In the message text:
filename
LDAP server configuration file name
error_code
Error code from fgets()
reason_code
Reason code from fgets()
error_text
Error text corresponding to the error code

Example: None.

System action: The program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.

Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1034E Configuration line is too long.

Explanation: The LDAP server or utility cannot process the LDAP server configuration file because a line is too long. The maximum length of a line in the LDAP server configuration file is 1024 characters. This includes any continuation lines.

Example: None.

System action: The program ends.
GLD1035E  *option* is an unrecognized configuration option.

**Explanation:** The LDAP server or utility cannot process the LDAP server configuration file because it contains an option that is not supported.

In the message text:

```
option
   LDAP server configuration option
```

**Example:** None.

**System action:** The program ends.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Correct the LDAP server configuration file. Then restart the program.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

GLD1036W  *option* is an obsolete configuration option.

**Explanation:** The LDAP server or utility found an option that is no longer used in the LDAP server configuration file.

In the message text:

```
option
   LDAP server configuration option
```

**Example:** None.

**System action:** The program ignores the configuration option and continues.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Correct the LDAP server configuration file. Then restart the program.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

GLD1037E Either no value or not enough values have been specified for the *option* configuration option.

**Explanation:** The LDAP server or utility found an option with either no value or not enough values in the LDAP server configuration file. Every configuration option must have an appropriate number of values specified for it.

In the message text:

```
option
   LDAP server configuration option
```

**Example:** None.

**System action:** The program ends.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Remove the obsolete option from the LDAP server configuration file.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

GLD1038E  *value* for configuration option *option* is not valid.

**Explanation:** The LDAP server or utility found an option in the LDAP server configuration file that has a value that is not supported for that option.

In the message text:

```
value
   LDAP server configuration option
```

**Example:** None.
value
   LDAP server configuration option value

option
   LDAP server configuration option

Example: None.
System action: The program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Correct the LDAP server configuration file. If the option value looks correct, check that the option on the next line after this option line starts in column 1. A blank in column 1 of the next line indicates that it is a continuation line. The next line is then appended to the preceding option line and thus can result in a value that is not supported for the option. Then restart the program.

Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1040E Unable to normalize name: error_text.
Explanation: The LDAP server or utility is unable to normalize a distinguished name (DN). This error can occur if any part of the DN does not contain an attribute type and value or if the attribute type is not defined in the directory schema or does not have an equality matching rule. The message displays either the DN or information about where the DN is specified. If a DN is displayed, the DN can be part of the value of an LDAP server configuration option, an LDAP utility command-line option, or an attribute value. Otherwise, the message displays a name indicating where the DN was specified, for example, the name of an LDAP server configuration option or of an attribute in an entry.

In the message text:
name
   DN or source of DN

error_text
   Error message text

Example: None.
System action:
- If the error occurs while running an LDAP utility, the program ends.
- If the error occurs during LDAP server processing of the configuration file, the program ends.
- If the error occurs during initialization of an LDAP server backend, then the backend does not start. If the srvStartUpError option in the LDAP server configuration file is set to ignore, the LDAP server continues to run with those backends that successfully start. If the srvStartUpError option is set to terminate (this is the default if the configuration option is not specified), the program ends.

Chapter 30. LDAP server messages 521
• If the error occurs while processing an LDAP server operation, the operation may fail.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Use the information in the message to correct the problem. This may involve changing the value of an option in the LDAP server configuration file, an LDAP utility command-line option, or an attribute value in an entry. Restart the program if it didn't start or if a backend that didn't initialize is needed. If the error occurs during an LDAP operation, retry the operation.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD1041E Configuration option option is not allowed in the section section.**

**Explanation:** The LDAP server or utility found an option in a section of the LDAP server configuration file that is not appropriate for that section. Global options must be specified before the first `database` option, while backend-specific options must be specified following the `database` option for that backend.

In the message text:

```
option
  LDAP server configuration option
section
  LDAP server configuration section name
```

**Example:** None.

**System action:** The program ends.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administration response:** Correct the LDAP server configuration file. Then restart the program.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD1042E Backend name name is already defined.**

**Explanation:** The LDAP server or utility found a backend name on a `database` option in the LDAP server configuration file that is the same as the name for a previous backend. If a backend name is specified, the name must be unique. There are also several reserved backend names that cannot be used: `RootDSE`, `Schema`, and `Monitor`. Backend names are not case sensitive.

In the message text:

```
name
  Backend name
```

**Example:** None.

**System action:** The program ends.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Correct the backend name in the LDAP server configuration file. Then restart the program.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD1043E Configuration file filename causes a recursion loop.**

**Explanation:** The LDAP server or utility found an LDAP server configuration file that is included again while it is still being processed. This is a result of nested `include` options for the same configuration file.

In the message text:

```
filename
  LDAP server configuration file name
```

**Example:** None.

**System action:** The program ends.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Remove the nested `include` options from the LDAP server configuration file. Then restart the program.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.
GLD1045E  The MAC address must be 12 hexadecimal digits.

Explanation: The LDAP server or utility found that the value for the serverEtherAddr option in the LDAP server configuration file is not valid. The MAC address must consist of 12 hexadecimal digits.

Example: None.

System action: The program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Correct the value for the serverEtherAddr option in the LDAP server configuration file. Then restart the program.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1046E  Configuration option option cannot be converted to IBM-1047.

Explanation: The LDAP server or utility cannot convert the value of an option in the LDAP server configuration file. The value needs to be converted to the IBM-1047 code page but contains characters that cannot be represented in that code page.

In the message text:

option
  LDAP server configuration option

Example: None.

System action: The program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the databaseDirectory option to specify a unique file directory for each LDBM and file-based GDBM backend in the LDAP server configuration file. Then restart the program.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1047E  Directory path is in use by another database instance.

Explanation: The LDAP server or utility found multiple LDBM or file-based GDBM backends using the same directory for database files. Each instance of the LDBM or file-based GDBM backend requires a unique directory for its database files. The file directory is specified by the databaseDirectory option in the backend section of the LDAP server configuration file.

In the message text:

path
  Database directory path

Example: None.

System action: The program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Specify a string consisting of valid characters in the IBM-1047 character set for the option in the LDAP server configuration file. Then restart the program.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1048E  Unable to add schema definition: error_text.

Explanation: The LDAP server is unable to add a new definition to the directory schema.

In the message text:

error_text
  Error message text

Example: None.

System action: The program ends.

Operator response: None.
GLD1050E Unable to create thread: error_code/reason_code - error_text

Explanation: The LDAP server is unable to create a thread. Refer to the description of pthread_create() in z/OS XL C/C++ Run-Time Library Reference for more information on the error.

In the message text:

- error_code
  Error code from pthread_create()

- reason_code
  Reason code from pthread_create()

- error_text
  Error text corresponding to the error code

Example: None.

System action: The program continues. The request fails.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the information in the message to correct the error. Then retry the request. If the problem persists, contact the service representative.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1051A Unable to start the console task.

Explanation: The LDAP server is unable to start the console task. A previous message indicates the reason for the failure.

Example: None.

System action: The program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the information in the earlier message to correct the problem. Then restart the program. If the problem persists, contact the service representative.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1052A Unable to start the network task.

Explanation: The LDAP server is unable to start the interfaces used by the LDAP server. A previous message indicates the reason for the failure.

Example: None.

System action: The program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the information in the earlier message to correct the problem. Then restart the program. If the problem persists, contact the service representative.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1053E Unable to wait on condition variable: error_code/reason_code - error_text

Explanation: The LDAP server is unable to wait on a condition variable. Refer to the description of pthread_cond_wait() in z/OS XL C/C++ Run-Time Library Reference for more information.

Example: None.

System action: The program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the information in the earlier message to correct the problem. Then restart the program. If the problem persists, contact the service representative.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.
Library Reference for more information on the error.

In the message text:

**error_code**
Return code from `pthread_cond_wait()`

**reason_code**
Reason code from `pthread_cond_wait()`

**error_text**
Error text corresponding to the error code

**Example:** None.

**System action:** The program continues. The request fails.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Use the information in the message to correct the error. Then retry the request. If the problem persists, contact the service representative.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

GLD1055E Unable to create a network socket:

```
error_code/reason_code - error_text
```

**Explanation:** The LDAP server is unable to create a network socket. Refer to the description of the `socket()` function in [z/OS XL C/C++ Run-Time Library Reference](https://www.ibm.com/docs/en/zos-run-time-library-reference) for more information on the error.

In the message text:

**error_code**
Error code from `socket()`

**reason_code**
Reason code from `socket()`

**error_text**
Error text corresponding to the error code

**Example:** None.

**System action:** The program continues. The request fails.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Use the information in the message to correct the error. Then retry the request. If the problem persists, contact the service representative.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

GLD1054E Unable to signal a condition variable:

```
error_code/reason_code - error_text
```

**Explanation:** The LDAP server is unable to signal a condition variable. Refer to the description of the `pthread_cond_signal()` function in [z/OS XL C/C++ Run-Time Library Reference](https://www.ibm.com/docs/en/zos-run-time-library-reference) for more information on the error.

In the message text:

**error_code**
Error code from `pthread_cond_signal()`

**reason_code**
Reason code from `pthread_cond_signal()`

**error_text**
Error text corresponding to the error code

**Example:** None.

**System action:** The program continues. The request fails.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Use the information in the message to correct the error. Then retry the request. If the problem persists, contact the service representative.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

GLD1056E Unable to obtain the network configuration:

```
error_code/reason_code - error_text
```

**Explanation:** The LDAP server is unable to obtain the network configuration. Refer to the description of the `ioctl()` function in [z/OS XL C/C++ Run-Time Library Reference](https://www.ibm.com/docs/en/zos-run-time-library-reference) for more information on the error.

In the message text:

**error_code**
Error code from `ioctl()`

**reason_code**
Reason code from `ioctl()`

**error_text**
Error text corresponding to the error code

**Example:** None.

**System action:** The program continues. The request fails.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Use the information in the message to correct the error. Then retry the request. If the problem persists, contact the service representative.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.
error_code
    Error code from ioctl()

reason_code
    Reason code from ioctl()

error_text
    Error text corresponding to the error code

Example: None.
System action: The program continues. The request fails.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Use the information in the message to correct the error. Then retry the request. If the problem persists, contact the service representative.

Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1057E Unable to obtain the status of the interface: error_code/reason_code - error_text

Explanation: The LDAP server is unable to obtain the status of the indicated network interface. Refer to the description of the SIOCIFLAGS option for ioctl() in z/OS XL C/C++ Run-Time Library Reference for more information on the error.

In the message text:

name
    Network interface name

error_code
    Error code from ioctl()
reason_code
    Reason code from ioctl()
error_text
    Error text corresponding to the error code

Example: None.
System action: The program continues. The request fails.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Use the information in the message to correct the error. Then retry the request. If the problem persists, contact the service representative.

Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1058E Unable to obtain the IPv6 home interfaces: error_code/reason_code - error_text

Explanation: The LDAP server is unable to obtain the list of IPv6 home interfaces. Refer to the description of the SIOCCHOMEIF6 option for ioctl() in z/OS XL C/C++ Run-Time Library Reference for more information on the error.

In the message text:

error_code
    Error code from ioctl()
reason_code
    Reason code from ioctl()
error_text
    Error text corresponding to the error code

Example: None.
System action: The program continues. The request fails.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Use the information in the message to correct the error. Then retry the request. If the problem persists, contact the service representative.

Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1059I Listening for requests on ip port port.

Explanation: The LDAP server is listening for non-secure requests on the indicated network interface.

In the message text:

ip
    IP address
**GLD1060I** No longer listening for requests on ip port port.

Explanation: The LDAP server is no longer listening for requests on the indicated network interface. This indicates that the network interface is no longer available.

In the message text:
- **ip** IP address
- **port** Port number

Example: None.

System action: The program continues.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

**GLD1063E** Unable to initialize the SSL environment: return_code - error_text.

Explanation: The LDAP server cannot initialize the SSL environment. Refer to the description of the gsk_environment_open() and gsk_environment_init() routines in Z/OS Cryptographic Services System SSL Programming for more information on the error.

In the message text:
- **return_code** Return code from SSL routine
- **error_text** Error text corresponding to the return code

Example: None.

System action: If the error occurs during backend initialization, the tcpTerminate option in the LDAP server configuration file determines what the server does. If the tcpTerminate option is set to recover (this is the default if the configuration option is not specified), LDAP server initialization continues. In this case, SSL support is not available until the error is corrected and the server is restarted. If the tcpTerminate option is set to terminate, the program ends. If the error occurs while processing the LDAP server REFRESH SSL operator modify command, the program continues, using the existing SSL environment.

Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Use the information in the message to correct the error. If SSL connections are not needed, remove the sslKeyRingFile option from the LDAP server configuration file. Restart the program if it ended or if SSL connections are needed.
Problem determination: Not applicable.
Source: LDAP
GLD1064E Unable to load the System SSL runtime: error_code/reason_code - error_text

Explanation: The LDAP server cannot load the System SSL runtime DLL. Refer to the description of dllload() in z/OS XL C/C++ Run-Time Library Reference for more information on the error.

In the message text:

error_code
   Error code from dllload()
reason_code
   Reason code from dllload()
error_text
   Error text corresponding to the error code

Example: None.

System action: LDAP server initialization continues if the tcpTerminate option in the LDAP server configuration file is set to recover (this is the default if the configuration option is not specified). In this case, SSL support is not available until the error is corrected and the server is restarted. If the tcpTerminate option is set to terminate, the program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the information in the message to correct the error. Verify that the SYS1.SIEALNKE dataset is available to the LDAP server job step. If SSL connections are not needed, remove the sslKeyRingFile option from the LDAP server configuration file. Restart the program if it ended or if SSL connections are needed.

Problem determination: Not applicable.

Source: LDAP

Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1066E Unable to bind to ip port: error_code/reason_code - error_text

Explanation: The LDAP server is unable to bind to the indicated network interface. Refer to the description of bind() in z/OS XL C/C++ Run-Time Library Reference for more information on the error.

In the message text:

ip
   IP address
port
   Port number
error_code
   Error code from bind()
reason_code
   Reason code from bind()
error_text
   Error text corresponding to the error code

Example: None.

System action: The program continues. The request fails.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the information in the message to correct the error. Ensure that no other application is using the indicated port and that the port is not reserved. Then retry the request. If the problem persists, contact the service representative.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1067E Unable to listen for requests:
   error_code/reason_code - error_text

Explanation: The LDAP server cannot listen for requests on a network interface. Refer to the description of listen() in [z/OS XL C/C++ Run-Time Library Reference] for more information on the error.

In the message text:

   error_code
      Error code from listen()

   reason_code
      Reason code from listen()

   error_text
      Error text corresponding to the error code

Example: None.

System action: The program continues. The request fails. This message will be issued at most once a minute for a limit of 60 times when this condition exists. Although this message may not be issued after being displayed 60 times on the console, the condition may still exist.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the information in the message to correct the error. Then retry the request. If the problem persists, contact the service representative.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1068E Unable to accept connection:
   error_code/reason_code - error_text

Explanation: The LDAP server cannot accept a connection on a network interface. Refer to the description of accept() in [z/OS XL C/C++ Run-Time Library Reference] for more information on the error.

In the message text:

   error_code
      Error code from accept()

   reason_code
      Reason code from accept()

   error_text
      Error text corresponding to the error code

Example: None.

System action: The program continues. The request fails. This message will be issued at most once a minute for a limit of 60 times when this condition exists. Although this message may not be issued after being displayed 60 times on the console, the condition may still exist.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the information in the message to correct the error. Then retry the request. If the problem persists, contact the service representative.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1069E Unable to receive data:
   error_code/reason_code - error_text

Explanation: The LDAP server cannot receive data on a network interface. Refer to the description of recv() in [z/OS XL C/C++ Run-Time Library Reference] for more information on the error.

In the message text:

   error_code
      Error code from recv()

   reason_code
      Reason code from recv()
error_text
    Error text corresponding to the error code
Example: None.
System action: The program continues. The request fails.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Use the information in the message to correct the error. Then retry the request. If the problem persists, contact the service representative.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1070E  suffix is a reserved database suffix.

Explanation: The LDAP server or utility found a suffix option in the LDAP server configuration file which specifies a value that is reserved for use by the LDAP server. The LDAP server reserves "", "cn=*,", and "cn=monitor" as suffixes for internal backends. It restricts usage of "cn=change*" when the GDBM backend is configured. The LDAP server also reserves "cn=*,", "cn=Authenticated", and "cn=This", because it uses these distinguished names to represent special-purpose access groups.

In the message text:

  suffix
    Suffix option value
Example: None.
System action: The program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Correct the suffix option in the LDAP server configuration file. Then restart the program.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1071E  suffix is a duplicate database suffix.

Explanation: The LDAP server or utility found a suffix option value in the LDAP server configuration file which is a duplicate of another suffix option value. Each suffix value must be unique and must not be subordinate to another suffix value. For example, "o=IBM, c=US" and "c=US" cannot both be assigned as suffixes since the first value is a subordinate of the second value.

In the message text:

  suffix
    Suffix option value
Example: None.
System action: The program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Correct the suffix option in the LDAP server configuration file. Then restart the program.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1072E  Unable to send data:

Explanation: The LDAP server cannot send data to a client application on a network interface. Refer to the description of send() in z/OS XL C/C++ Run-Time Library Reference for more information on the error.

In the message text:

  error_code
    Error code from send()
reason_code
    Reason code from send()
error_text
    Error text corresponding to the error code
Example: None.
System action: The program continues. The request may fail. This message could be accompanied by a failure of the client application due to lost response data. Client symptoms might include timeouts, long waits, or connection failures.
Operator response: None.
**GLD1074W** Maximum client connections changed from old_value to new_value.

**Explanation:** The value for the **maxConnections** configuration option in the LDAP server configuration file is too large compared to the maximum number of file descriptors allowed for the LDAP server process. The **maxConnections** configuration option determines the maximum number of concurrent client connections. Each client connection requires a socket descriptor and each socket descriptor counts against the maximum number of files for a process. The LDAP server requires 4 file descriptors plus 2 file descriptors for each backend, plus a minimum of 30 file descriptors for network connections. To avoid running out of file descriptors, a limit is placed on the maximum number of concurrent client connections based on the current file limit.

In the message text:

- **old_value**
  - Old maximum client connections value
- **new_value**
  - New maximum client connections value

**Example:** None.

**System action:** The program continues, using the updated value for the maximum number of concurrent client connections in order to honor the current file limit for the LDAP server process.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** If the **maxConnections** configuration option must be old_value, increase the values of the **MAXFILEPROC** statement and of **MAXSOCKETS** on the **NETWORK** statement in the **BPXPRMxx** member. It may also be necessary to increase the **FILEPROCMAX** value in the RACF OMVS segment of the user ID running the LDAP server so that the old_value can be supported. Then restart the LDAP server. See the documentation on the **maxConnections** configuration option for more information.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

**GLD1075W** Client connection threshold reached, currently using current_value of maximum_value.

**Explanation:** The number of concurrent client connections has reached 90% of the maximum number of connections allowed on the LDAP server.

In the message text:

- **current_value**
  - Current number of client connections
- **maximum_value**
  - Maximum number of client connections

**Example:** None.

**System action:** The program continues, but is in danger of reaching the maximum number of concurrent client connections allowed.

**Operator response:** Contact the LDAP Administrator or see Administrator response.

**User response:** None.

**System programmer response:** None.

**Administrator response:** A common reason that client connections are consumed in the LDAP server is because client applications are not unbinding from the LDAP server when they are finished making requests. Ensure that client applications disconnect when they are finished making requests to the LDAP server. If this is not the problem, increase the number of connections allowed on the LDAP server.

- If the **maxConnections** configuration option is set in the LDAP server configuration file, increase its value. Verify the increased value of **maxConnections** can be supported by obtaining the values of the **MAXFILEPROC** statement and of **MAXSOCKETS** on the **NETWORK** statement in the **BPXPRMxx** member. Also verify that the **FILEPROCMAX** value in the RACF OMVS segment of the user ID running the LDAP server is set to a sufficient value to support the increased value of **maxConnections**.
- If **maxConnections** is not set in the LDAP server configuration file, the number of connections is limited by the values of the **MAXFILEPROC** statement and of **MAXSOCKETS** on the **NETWORK** statement in **BPXPRMxx**, and also by the value of...
FILEPROCMAX in the RACF OMVS segment of the user ID running the LDAP server. Ensure these are set to a sufficient value.

If any of these values are updated, it is necessary to restart the LDAP server to put these changes into effect. See the documentation on the maxConnections configuration option for more information.

Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1076I Number of client connections now below threshold, currently using current_value of maximum_value.

Explanation: The number of concurrent client connections has dropped below 90% of the maximum number of connections allowed on the LDAP server. Warning messages may be issued once again if the number of concurrent client connections exceeds the warning threshold.

In the message text:
current_value  Current number of client connections
maximum_value  Maximum number of client connections

Example: None.
System action: The program continues. Additional client applications cannot connect to the LDAP server. This message will be issued at most once a minute for a limit of 60 times when this condition exists. Although this message may not be issued after being displayed 60 times on the console, the condition may still exist.

Operator response: Contact the LDAP Administrator or see Administrator response.
User response: None.
System programmer response: None.
Administrator response: A common reason that client connections are consumed in the LDAP server is because client applications are not unbinding from the LDAP server when they are finished making requests. Ensure that client applications disconnect when they are finished making requests to the LDAP server. If this is not the problem, increase the number of connections allowed on the LDAP server.

- If the maxConnections configuration option is set in the LDAP server configuration file, increase its value. Verify the increased value of maxConnections can be supported by obtaining the values of the MAXFILEPROC statement and of MAXSOCKETS on the NETWORK statement in the BPXPRMxx member. Also verify that the FILEPROCMAX setting in the RACF OMVS segment of the user ID running the LDAP server is set to a sufficient value to support the increased value of maxConnections.

- If maxConnections is not set in the LDAP server configuration file, the number of connections is limited
by the values of the MAXFILEPROC statement and of MAXSOCKETS on the NETWORK statement in BPXPRMxx, and also by the value of FILEPROCMax in the RACF OMVS segment of the user ID running the LDAP server. Ensure these are set to a sufficient value.

If any of these values are updated, it is necessary to restart the LDAP server to put these changes into effect. See the documentation on the maxConnections configuration option for more information.

Problem determination: Not applicable.

Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

---

GLD1078E Unable to get the value for the maximum number of files:

**Explanation:** The LDAP server is unable to determine the maximum number of files allowed for a process. Refer to the description of getrlimit() in the z/OS XL C/C++ Run-Time Library Reference for more information on the error.

In the message text:

- error_code
  - Error code from getrlimit()
- reason_code
  - Reason code from getrlimit()
- error_text
  - Error text corresponding to the error code

**Example:** None.

System action: The program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Increase the values of the MAXFILEPROC statement and of MAXSOCKETS on the NETWORK statement in the BPXPRMxx member. It may also be necessary to increase the FILEPROCMax value in the RACF OMVS segment of the user ID running the LDAP server. Then restart the LDAP server. See the documentation on the maxConnections configuration option for more information.

---

GLD1079E Maximum file limit of current_limit is too small, change it to at least new_limit.

**Explanation:** The maximum number of files that can be opened by the LDAP server process is too small. The LDAP server requires 4 file descriptors plus 2 file descriptors for each backend, plus a minimum of 30 file descriptors for network connections. The current maximum file limit displayed in the message is not large enough to support the minimum of 30 file descriptors required for network connections. The maximum file limit must be set to at least the indicated new limit for the LDAP server to start.

In the message text:

- current_limit
  - Current maximum file limit
- new_limit
  - Recommended maximum file limit

**Example:** None.

System action: The program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Use the information in the message to correct the error. Then restart the LDAP server. See the documentation on the maxConnections configuration option for more information.

---

GLD1080E Unable to load the Kerberos runtime:

**Explanation:** The LDAP server encountered an error in attempting to load the Kerberos runtime DLL. Refer to the description of krb5_dll_load() in the z/OS Integrated Security Services Network Authentication Service Programming: SC24-5927 for more information on the return codes. Refer to z/OS UNIX System Services Messages and Codes, SA22-7807 for more information on the reason codes.

In the message text:

- return_code
  - Return code from krb5_dll_load()
**reason_code**
Reason code from `krb5_dll_load()`

**error_text**
Error text corresponding to the return code

**Example:** None.

**System action:** The LDAP server continues initialization if the `tcpTerminate` option in the LDAP server configuration file is set to `recover` (this is the default if the configuration option is not specified). In this case, Kerberos support is not available until the error is corrected and the server is restarted. If the `tcpTerminate` option is set to `terminate`, the program ends.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Use the information in the message to correct the error. Verify that the `SYS1.SIEALNKE` dataset is available to the LDAP server job step. If Kerberos authentication is not needed, set the `supportKrb5` option in the LDAP server configuration file to `off`. Restart the program if it ended or if Kerberos support is needed.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD1081A** No network interface is available for a 'listen' statement.

**Explanation:** There are no network interfaces available for a `listen` statement. This error can also occur if a `listen` option in the LDAP server configuration file or on the LDAP server command line specifies SSL connections but SSL support is not available.

**Example:** None.

**System action:** The LDAP server continues if the `tcpTerminate` option in the LDAP server configuration file is set to `recover` (this is the default if the configuration option is not specified) or if at least one network interface starts successfully. Otherwise, the program ends.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Either start the required network interface or remove the corresponding `listen` option from the LDAP server configuration file or command line. Restart the program if it ended.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD1082A** Network interface `ip` is not available.

**Explanation:** A required network interface is not available for use.

In the message text:

```
ip  IP address
```

**Example:** None.

**System action:** The LDAP server continues if the `tcpTerminate` option in the LDAP server configuration file is set to `recover` (this is the default if the configuration option is not specified) or if at least one network interface starts successfully. Otherwise, the program ends.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Either start the required network interface or remove the corresponding `listen` option from the LDAP server configuration file or command line. Restart the program if it ended.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD1083A** Host `host` cannot be resolved.

**Explanation:** The host name specified on a `listen` option in the LDAP server configuration file or on the LDAP server command line cannot be resolved.

In the message text:

```
host
  Host name
```

**Example:** None.

**System action:** The LDAP server continues if the `tcpTerminate` option in the LDAP server configuration file is set to `recover` (this is the default if the configuration option is not specified) or if at least one network interface starts successfully. Otherwise, the program ends.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Either start the required network interface or remove the corresponding `listen` option from the LDAP server configuration file or command line. Restart the program if it ended.

**Problem determination:** Not applicable.
network interface starts successfully. Otherwise, the program ends.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Verify that the DNS name server is available and that the host name is defined. Ensure that the host name is specified correctly or remove the corresponding `listen` option from the LDAP server configuration file or command line. Restart the program if it ended.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

GLD1084I Network interface status

**Explanation:** This message is displayed in response to the LDAP server `DISPLAY NETWORK` operator modify command. The remaining lines in this multi-line message display the status of each network interface. A network interface is **ACTIVE** if the LDAP server is listening for requests on that interface. A network interface is **INACTIVE** if the interface has been stopped and has not been restarted yet. No entry is displayed for network interfaces which were not started when the LDAP server was started. The LDAP server checks for network interface changes based on the value of the `LDAP_NETWORK_POLL` environment variable, which has a default value of 5 minutes.

**Example:** None.

**System action:** The program continues.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** None.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

GLD1085I No active network interfaces.

**Explanation:** This message is displayed in response to the LDAP server `DISPLAY NETWORK` operator modify command when there are no active network interfaces to display, and no network interfaces ever started successfully.

**Example:** None.

**System action:** The program continues.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** None.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

GLD1086I Maximum number of lines displayed.

**Explanation:** There is a limit of 254 lines of output from an LDAP server operator modify command. The maximum number of output lines has been reached for this command and the rest of the output is not displayed.

**Example:** None.

**System action:** The program continues.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** None.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

GLD1087E The type backend is already defined.

**Explanation:** The LDAP server or utility found multiple `database` options in the LDAP server configuration file for a GDBM, SDBM, or EXOP backend. Each of these backends can be defined at most once in the configuration file.
GLD1088E The EXOP backend requires Program Call services.

Explanation: The Policy Directory extended operations backend requires Program Call services. The LDAP server must have a listen option specifying ldap://:pc or ldaps://:pc in order to provide Program Call services. The listen option can be specified in the LDAP server configuration file or on the LDAP server command line when starting the server.

Example: None.

System action: The program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Either specify a listen option for Program Call services in the LDAP server configuration file or on the LDAP server command line, or remove the EXOP database configuration option. Then restart the program.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1090E The SDBM backend supports a single suffix.

Explanation: The LDAP server or utility found multiple suffix options in the SDBM section of the LDAP server configuration file. There can only be one SDBM backend section in the configuration file and it must contain exactly one suffix option.

Example: None.

System action: The program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Remove the extra suffix options from the SDBM section of the LDAP server configuration file. Then restart the program.

Problem determination: Not applicable.

Source: LDAP

Module: None.
Routing code:  None.
Descriptor code:  None.
Automation:  Not applicable.

GLD1091E Unable to open schema database file
filename: error_code/reason_code - error_text

Explanation:  The LDAP server or utility is unable to open the schema database file. Refer to the description of fopen() in z/OS XL C/C++ Run-Time Library Reference for more information on the error.
In the message text:
filename
  Schema database file name
error_code
  Error code from fopen()
reason_code
  Reason code from fopen()
error_text
  Error text corresponding to the error code

Example:  None.
System action:
  • If the error occurs during LDAP server initialization, the program ends.
  • If the error occurs during a schema modify operation, the schema modification is successful. If the LDAP server is part of a cross-system group in a sysplex, the other LDAP servers in the sysplex may not apply the schema change to their version of the schema. In this case, add and modify operations on those LDAP servers may fail if they involve the modified schema elements.
  • If the error occurs when processing a request for the schema from another LDAP server in the sysplex, the other LDAP server will end because it cannot obtain the schema.
Operator response:  None.
User response:  None.
System programmer response:  None.
Administrator response:  Use the information in the message to correct the error. Then restart the program.
Problem determination:  Not applicable.
Source:  LDAP
Module:  None.
Routing code:  None.
Descriptor code:  None.
Automation:  Not applicable.

GLD1093E Unable to write to schema database file
filename: error_code/reason_code - error_text

Explanation:  The LDAP server is unable to write the schema database file. Refer to the description of fwrite() in z/OS XL C/C++ Run-Time Library Reference for more information on the error.
In the message text:
filename
  Schema database file name
error_code
  Error code from fwrite()
reason_code
  Reason code from fwrite()
**GLD1094E Unable to create directory name**:  
_
error_code/reason_code - error_text_

**Explanation:** The LDAP server or utility is unable to create the indicated directory for the schema database file or for the checkpoint file for an LDBM or file-based GDBM backend. Refer to the description of `mkdir()` in `z/OS XL C/C++ Run-Time Library Reference` for more information on the error.

In the message text:

- `name`:
  - Directory name
- `error_code`:
  - Error code from `mkdir()`
- `reason_code`:
  - Reason code from `mkdir()`
- `error_text`:
  - Error text corresponding to the error code

**Example:** None.

**System action:**
- If the error occurs during schema initialization, the program ends.
- If the error occurs during LDBM or GDBM initialization, then the LDBM or GDBM backend does not start. If the `srvStartUpError` option in the LDAP server configuration file is set to `ignore`, the LDAP server continues to run with those backends that successfully start. If the `srvStartUpError` option is set to `terminate` (this is the default if the configuration option is not specified), the program ends. The utility ends regardless of the option value.
- If the error occurs during a modify operation of the schema, the modify operation fails and the server continues to run with its current schema.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Use the information in the message to correct the error. Also verify that the LDAP server has write access to the directory containing the file and that there are no filesystem errors. Restart the program if it didn’t start or retry the schema modify operation.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD1095E Schema database file filename is not valid.**

**Explanation:** The LDAP server or utility is not able to load the schema from the schema database file. Either the record format is not as expected or the schema is not complete. If the file name indicated in the message is **XCF**,** then the schema was sent to the LDAP server from another LDAP server in the sysplex.

In the message text:

- `filename`:
  - Schema database file name

**Example:** None.

**System action:**
- The program ends unless the internal schema is still usable.

**Operator response:** None.

**User response:** None.
System programmer response: None.

Administrator response: Verify that the schema database file is not modified by any application other than the LDAP server. Then restart the program. If the problem persists, contact the service representative.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1096E Unable to decode schema database record.

Explanation: The LDAP server or utility detected an error while trying to load the schema. The schema can be loaded from either the schema database file or from the sysplex group owner through XCF. This error indicates that the LDAP server could not decode one of the schema database records.

Example: None.

System action: The program ends unless the internal schema is still usable.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Verify that the schema database file is not modified by any application other than the LDAP server. Then restart the program. If the problem persists, contact the service representative.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1097E Unable to encode schema database record.

Explanation: The LDAP server detected an error while trying to save the schema to the schema database file. It could not encode one of the database records.

Example: None.

System action: The program ends unless the internal schema is still usable.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Verify that the schema database file is not modified by any application other than the LDAP server. Then restart the program. If the problem persists, contact the service representative.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1097E Unable to rename oldfile to newfile: error_code/reason_code - error_text

Explanation: The LDAP server is unable to rename a file. Refer to the description of rename() in z/OS XL C/C++ Run-Time Library Reference for more information on the error.

In the message text:

oldfile
    Old file name

newfile
    New file name

error_code
    Error code from rename()

reason_code
    Reason code from rename()

error_text
    Error text corresponding to the error code

Example: None.

System action:

- If the error occurs during schema initialization, the server ends.
- If the error occurs during LDBM or GDBM initialization, then the LDBM or GDBM backend does not start. If the srvStartUpError option in the LDAP server configuration file is set to ignore, the LDAP server continues to run with those backends that successfully start. If the srvStartUpError option is set to terminate (this is the default if the configuration option is not specified), the program ends.
- If the error occurs during a modify operation of the schema, the modify operation fails and the LDAP server continues to run with its current schema.
- If the error occurs during an LDBM or GDBM operation, then, if the fileTerminate option in the LDAP server configuration file is set to recover (this is the default if the configuration option is not specified), the server continues to run but the LDBM...
or GDBM backend is placed in read-only state. If the fileTerminate option is set to terminate, the program ends.

**Operator response:** Use the information in the message to assist the LDAP administrator to correct the error. If requested, issue the LDAP server BACKEND operator modify command to set the LDBM or GDBM backend to read-write state.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Use the information in the message to correct the error. Also verify that the LDAP server has write access to the directory. Then restart the LDAP server if it didn't start. If an LDBM or GDBM backend was placed in read-only state, it can be reset to read-write state by restarting the LDAP server or by using the LDAP server BACKEND operator modify command.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD1099E** Schema unique identifier number is already assigned.

**Explanation:** Each attribute and object class in the LDAP server schema is identified by a unique internal identifier. While adding an attribute or object class to the schema, the LDAP server has detected that the attribute or object class identifier is already in use. This should not occur.

In the message text:

number

Identifier number

**Example:** None.

**System action:** If the error occurs during LDAP server initialization, the server ends. If it occurs during a modify operation of the schema, the modify operation fails and the server continues to run with its current schema.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Restart the LDAP server. If the problem persists, contact the service representative.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD1100A** LDAP server shutdown initiated because directory schema cannot be restored.

**Explanation:** The LDAP server is stopping because an attempt to load the directory schema has failed and the schema cannot be used. A previous message identifies the reason for the failure.

**Example:** None.

**System action:** The server ends.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Use the information in the earlier message to correct the problem. Then restart the program. If the problem persists, contact the service representative.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD1101A** Unable to load the database backends.

**Explanation:** The LDAP server is unable to load the database backends. A previous message indicates the reason for the failure.

**Example:** None.

**System action:** The server ends.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Use the information in the earlier message to correct the problem. Then restart the program. If the problem persists, contact the service representative.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.
GLD1102E  No type load module specified for 64-bit addressing mode.

**Explanation:** The LDAP server or utility is running in 64-bit addressing mode but one of the `database` options in the LDAP server configuration file does not specify a load module for 64-bit addressing mode. As a result, the backend is not loaded.

In the message text:

```plaintext
type
Backend type
```

**Example:** None.

**System action:** The program ends.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Either specify a load module for 64-bit addressing mode on the `database` option or remove the backend section from the LDAP server configuration file. The backend section includes the `database` option and all the options following it until the next `database` option. Restart the program.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

GLD1103E  No backend load module specified for 31-bit addressing mode.

**Explanation:** The LDAP server or the utility is running in 31-bit addressing mode but one of the `database` options in the LDAP server configuration file does not specify a load module for 31-bit addressing mode. As a result, the backend is not loaded.

In the message text:

```plaintext
backend
Backend type
```

**Example:** None.

**System action:** The program ends.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Either specify a load module for 31-bit addressing mode on the `database` option or remove the backend section from the LDAP server configuration file. The backend section includes the `database` option and all the options following it until the next `database` option. Restart the program.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

GLD1104E  Unable to load DLL module: `error_code`/`reason_code` - `error_text`

**Explanation:** The LDAP server or utility is unable to load the indicated DLL. Refer to the description of `dllload()` in [z/OS XL C/C++ Run-Time Library Reference](https://www.ibm.com/support/pages/ibm-z-os-xl-c-c-run-time-library-reference) for more information on the error.

In the message text:

```plaintext
module
  DLL module
error_code
  Error code from `dllload()`
reason_code
  Reason code from `dllload()`
error_text
  Error text corresponding to the error code
```

**Example:** None.

**System action:** The backend does not start. If the `srvStartUpError` option in the LDAP server configuration file is set to `ignore`, the LDAP server continues to run with those backends that successfully start. If the `srvStartUpError` option is set to `terminate` (this is the default if the configuration option is not specified), the program ends. The utility ends regardless of the option value.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Use the information in the message to correct the error. Ensure that the DLL is installed and can be accessed by the LDAP server or utility. If the DLL module name is specified on a `database` option in the LDAP server configuration file, ensure that it is entered correctly there. Restart the program if it didn’t start or if the backend is needed.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.
GLD1105E Unable to query entry point name in DLL module: error_code/reason_code - error_text

Explanation: The LDAP server or the utility is unable to locate a required entry point in the indicated DLL. Refer to the description of dllqueryfn() in z/OS XL C/C++ Run-Time Library Reference for more information on the error.

In the message text:

name  Entry point name
module  DLL module
error_code  Error code from dllqueryfn()
reason_code  Reason code from dllqueryfn()
error_text  Error text corresponding to the error code

Example: None.

System action: The backend does not start. If the srvStartUpError option in the LDAP server configuration file is set to ignore, the LDAP server continues to run with those backends and plug-ins that successfully start. If the srvStartUpError option is set to terminate (this is the default if the configuration option is not specified), the program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the information in the earlier message to correct the error. For a TDBM or DB2-based GDBM backend, ensure that DB2 is available. Restart the program if it didn’t start or if the backend is needed.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1106E type backend initialization failed for backend named name.

Explanation: The indicated backend failed to initialize. A previous message indicates the reason for the failure.

In the message text:

type  Backend type
name  Backend name

Example: None.

System action: The backend does not start. If the srvStartUpError option in the LDAP server configuration file is set to ignore, the LDAP server continues to run with those backends and plug-ins that successfully start. If the srvStartUpError option is set to terminate (this is the default if the configuration option is not specified), the program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the information in the earlier message to correct the error. For a TDBM or DB2-based GDBM backend, ensure that DB2 is available. Restart the program if it didn’t start or if the backend is needed.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1107I SNAP dump completed.

Explanation: The LDAP server has completed writing the dump requested by the LDAP server SNAP operator modify command. The dump is written to the dataset specified by the CEEDUMP DD statement in the start procedure for the LDAP server.

Example: None.

System action: The LDAP server continues.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP
GLD1108I  Server statistics reset.

Explanation: The statistics monitored by the LDAP server have been reset. This message is displayed in response to the LDAP server `RESET THREADS` command.

Example: None.

System action: The LDAP server continues.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1109I  Server activity statistics

Explanation: This message is displayed in response to the LDAP server `DISPLAY THREADS` operator modify command. The remaining lines in this multi-line message display the activity statistics. The `RESET THREADS` operator modify command can be used to reset the activity statistics.

Example: None.

System action: The LDAP server continues.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1110E  An administrator DN must be specified using the adminDN configuration option.

Explanation: The LDAP server or utility found that the adminDN option is missing in the LDAP server configuration file. This option defines the LDAP administrator and is required.

Example: None.

System action: The program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Add the adminDN option to the global section of the LDAP server configuration file. Then restart the program.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1111E  Unable to normalize schema owner: text.

Explanation: The LDAP server or utility is unable to normalize the distinguished name of the schema entry owner. This may occur due to a series of conflicting modifications to the schema entryowner attribute and one or more attributetypes definitions within the schema.

In the message text:

```
text
```

Example: None.

System action: The program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the information in the message to correct the error. Then restart the program. It may be necessary to restore the schema from a backup. If the problem persists, contact the service representative.

Problem determination: Not applicable.

Source: LDAP

Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1112E Unable to process schema ACL:

Explanation: The LDAP server or utility is unable to process the access control list for the schema entry. This may occur due to a series of conflicting modifications to the schema aclentry attribute and one or more attributetypes definitions within the schema.

In the message text:

error_text

Example: None.
System action: The program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Use the information in the message to correct the error. Then restart the program. It may be necessary to restore the schema from a backup. If the problem persists, contact the service representative.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1113E The keylabel record in the encryption keys dataset is incorrect.

Explanation: The LDAP server or utility found a record in the encryption keys dataset that is not valid. Each record in the encryption keys dataset consists of a key label followed by one or more key components. Each key component consists of 16 hexadecimal characters. Blank lines and lines beginning with '#' or an '*' are commentary records and are ignored.

The following is an example of a properly formatted key in the encryption keys dataset.

```
label1 1010101010101010 1010101010101010
```

In the message text:

keylabel

Encryption key label name

Example: None.
System action: The utility ends. The LDAP server continues, but encryption and decryption of values using a key in the key dataset may fail. In particular, this may result in bind failures if the userPassword attribute value cannot be decrypted.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Correct any records in the encryption keys dataset that are longer than 255 bytes. Then restart the program.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1114E A record in the encryption keys dataset is longer than 255 bytes.

Explanation: The LDAP server or utility found a record in the encryption keys dataset that is too long. The maximum length of a record in the encryption keys dataset is 255 bytes.

Example: None.
System action: The utility ends. The LDAP server continues, but encryption and decryption of values using a key in the key dataset may fail. In particular, this may result in bind failures if the userPassword attribute value cannot be decrypted.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Correct any records in the encryption keys dataset that are longer than 255 bytes. Then restart the program.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1115E Label 'keylabel' is not available:

Explanation: The LDAP server or utility encountered an error attempting to encrypt a value using the key label indicated in the message and the AES or DES
algorithm. The label and algorithm to use are specified on the **pwEncryption** or **secretEncryption** options in the LDAP server configuration file.

In the message text:

- **keylabel**
  - Encryption key label name

- **error_text**
  - Error message text

**Example:** None.

**System action:** The program ends.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** If using an encryption keys dataset to store AES or DES keys, ensure that **keylabel** matches the label on one of the records in the dataset and that the keys are valid. A DES key that is specified in the encryption keys dataset consists of 8, 16 or 24 bytes with odd parity while an AES key consists of 32 bytes. If AES or DES keys are stored in an ICSF CKDS dataset, ensure that ICSF is running prior to starting the LDAP server or utility. Correct the LDAP server configuration file or the encryption keys dataset. Then restart the program.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD1116E Unable to initialize an SSL connection**

with **IP_address**: **return_code** - **Error_text**.

**Explanation:** The LDAP server encountered an error while initializing an SSL connection with the client connecting from the IP address indicated in the message. Refer to the descriptions of these routines in *z/OS Cryptographic Services System SSL Programming* for more information on the error.

In the message text:

- **IP_address**
  - Client IP address

- **return_code**
  - Return code from **gsk_secure_socket_open()**

- **Error_text**
  - Error text corresponding to the return code

**Example:** None.

**System action:** The LDAP server continues. The client request fails.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Use the information in the message to correct the error. Then retry the client operation. If the problem persists, contact the service representative.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD1117E Unable to read SSL data from**

**IP_address**: **return_code** - **Error_text**.

**Explanation:** The LDAP server encountered an error while attempting to read data from an SSL connection with the client connecting from the IP address indicated in the message. Refer to the description of **gsk_secure_socket_read()** in *z/OS Cryptographic Services System SSL Programming* for more information on the error.

In the message text:

- **IP_address**
  - Client IP address

- **return_code**
  - Return code from **gsk_secure_socket_read()**

- **Error_text**
  - Error text corresponding to the return code

**Example:** None.

**System action:** The LDAP server continues. The client request fails.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Use the information in the message to correct the error. Then retry the client operation. If the problem persists, contact the service representative.

**Problem determination:** Not applicable.

**Source:** LDAP
GLD1118E Unable to send SSL data to IP_address:
  return_code - Error_text.

Explanation: The LDAP server encountered an error while attempting to send data over an SSL connection with the client connecting from the IP address indicated in the message. Refer to the description of gsk_secure_socket_write() in z/OS Cryptographic Services System SSL Programming for more information on the error.

In the message text:

IP_address
  Client IP address

return_code
  Return code from gsk_secure_socket_write()

Error_text
  Error text corresponding to the return code

Example: None.

System action: The LDAP server continues. The bind request fails.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the information in the message to correct the error. Then retry the client operation. If the problem persists, contact the service representative.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1119E Unable to get SSL certificate information: return_code - Error_text.

Explanation: The LDAP server encountered an error while attempting to obtain certificate information for a client performing an SASL EXTERNAL bind over SSL. Refer to the description of gsk_attribute_get_cert_info() in z/OS Cryptographic Services System SSL Programming for more information on the error.

In the message text:

return_code
  Return code from gsk_attribute_get_cert_info()

Error_text
  Error text corresponding to the return code

Example: None.

System action: The LDAP server continues. The bind request fails.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the information in the message to correct the error. Then retry the client operation. If the problem persists, contact the service representative.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1120E Kerberos initialization failed:
  0xreturn_code - error_text.

Explanation: The LDAP server encountered an error while attempting to initialize the Kerberos runtime environment. The failing routine can be krb5_init_context(), krb5_sname_to_principal(), or krb5_unparse_name(). Refer to the descriptions of these routines in z/OS Integrated Security Services Network Authentication Service Programming for more information on the error.

In the message text:

return_code
  Return code from Kerberos routine

error_text
  Error text corresponding to the return code

Example: None.

System action: The LDAP server continues initialization if the tcpTerminate option in the LDAP server configuration file is set to recover (this is the default if the configuration option is not specified). In this case, Kerberos support is not available until the error is corrected and the server is restarted. If the tcpTerminate option is set to terminate, the program ends.

Operator response: None.

User response: None.

System programmer response: None.
**Administrator response:** Use the information in the message to correct the error. If Kerberos authentication is not needed, set the `supportKrb5` option in the LDAP server configuration file to `off`. Restart the program if it ended or if Kerberos support is needed. If the problem persists, contact the service representative.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD1121E Unable to parse Kerberos principal 'principal_name': 0xreturn_code - error_text.**

**Explanation:** The LDAP server is unable to parse the Kerberos principal specified by the `serverKrbPrinc` option in the LDAP server configuration file. Refer to the description of `krb5_parse_name()` in [z/OS Integrated Security Services Network Authentication Service Programming](https://www.ibm.com) for more information on the error.

In the message text:

- `principal_name` Kerberos principal name
- `return_code` Return code from `krb5_parse_name()`
- `error_text` Error text corresponding to the return code

**Example:** None.

**System action:** The LDAP server continues initialization if the `tcpTerminate` option in the LDAP server configuration file is set to `recover` (this is the default if the configuration option is not specified). In this case, Kerberos support is not available until the error is corrected and the server is restarted. If the `tcpTerminate` option is set to `terminate`, the program ends.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Use the information in the message to correct the error. Verify that a valid Kerberos principal name is specified for the `serverKrbPrinc` option in the LDAP server configuration file. Correct the option, or, if Kerberos authentication is not needed, set the `supportKrb5` option in the LDAP server configuration file to `off`. Restart the program if it ended or if Kerberos support is needed.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1123E Unable to wrap a GSSAPI message:
   Major 0xmajor_error, Minor 0xminor_error - principal_name.

Explanation: The LDAP server encountered a problem while attempting to cryptographically sign and possibly encrypt (wrap) a GSSAPI message. Refer to the description of gss_wrap() in z/OS Integrated Security Services Network Authentication Service Programming for more information on the error.

In the message text:
   major_error
   Major error code from gss_wrap()
   minor_error
   Minor error code gss_wrap()
   principal_name
   Kerberos server principal name

Example: None.
System action: The LDAP server continues. The client request fails.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Use the information in the message to correct the error. Correct the GSSAPI environment on the system. Then retry the client operation. If the problem persists, contact the service representative.

Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1125W The option configuration option is specified more than once.

Explanation: The LDAP server or utility found an option in the LDAP server configuration file that is specified more than once in the global section or in a backend section. The option can only be specified once in a section of the configuration file.

In the message text:
   option
   LDAP server configuration option

Example: None.
System action: The program continues, using the value in the last occurrence of the option in the LDAP server configuration file.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Remove the extra occurrences of the option in the global or backend section of the LDAP server configuration file so that the option is only specified once in that section. Restart the program if the desired option value is not being used.

Problem determination: Not applicable.

Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1124E Unable to unwrap a GSSAPI message:
   Major 0xmajor_error, Minor 0xminor_error - principal_name.

Explanation: The LDAP server encountered a problem while attempting to unwrap a GSSAPI message sealed by the gss_wrap() routine and verify the embedded signature. Refer to the description of gss_unwrap() in z/OS Integrated Security Services Network Authentication Service Programming for more information on the error.

In the message text:
GLD1126I Server lock statistics
Explanation: This message is displayed in response to the LDAP server DISPLAY LOCKS operator modify command. The remaining lines in this multi-line message display the lock contention statistics. There are two types of contention: waiting for shared control of the lock and waiting for exclusive control of the lock. For each type of request, the number of times that contention was encountered and the average wait time until the lock was obtained is displayed. The LDAP server RESET LOCKS operator modify command can be used to reset the lock contention statistics.
Example: None.
System action: The LDAP server continues.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1128E ARM element name arm_name is already in use.
Explanation: The LDAP server is unable to register with the Automatic Restart Management (ARM) service because the element name indicated in the message is already in use. This error can occur if the LDAP server is started multiple times on the same system and unique ARM names are not specified by the armName option in the LDAP server configuration file.
In the message text:

arm_name ARM element name
Example: None.
System action: The LDAP server continues, but Automatic Restart Management (ARM) is not available to the LDAP server. The LDAP server is not automatically restarted if it fails unexpectedly.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Specify unique values for the armName option in the LDAP server configuration file if multiple instances of the LDAP server are being ran on the same system. Restart the program if ARM support is needed.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1129I Program Call communication is active.
Explanation: The Program Call support interface is now active on the LDAP server.
Example: None.
System action: The LDAP server continues.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.
GLD1130E Program Call initialization failed:

   Return code return_code, Reason code reason_code.

Explanation: The LDAP server is unable to initialize the Program Call support. The return code has the following values:

1. Job step is not APF-authorized.
2. Program Call support is being used by another LDAP server on the same system.
3. ESTAEX create failed. The reason code is the ESTAEX return code. Refer to the description of ESTAEX in z/OS MVS Programming: Authorized Assembler Services Reference EDT-IXG for more information on the error.
4. LXRES failed. The reason code is the LXRES return code. Refer to the description of LXRES in z/OS MVS Programming: Authorized Assembler Services Reference LLA-SDU for more information on the error.
5. ETCRE failed. The reason code is the ETCRE return code. Refer to the description of ETCRE in z/OS MVS Programming: Authorized Assembler Services Reference EDT-IXG for more information on the error.
6. ETCON failed. The reason code is the ETCON return code. Refer to the description of ETCON in z/OS MVS Programming: Authorized Assembler Services Reference EDT-IXG for more information on the error.
7. IEANTCR failed. The reason code is the IEANTCR return code. Refer to the description of IEANTCR in z/OS MVS Programming: Authorized Assembler Services Reference EDT-IXG for more information on the error.

In the message text:

   return_code
      Return code

   reason_code
      Reason code

Example: None.

System action: The Program Call interface is not available. If the srvStartUpError option in the LDAP server configuration file is set to ignore, the LDAP server continues to run with those backends that successfully start. If the srvStartUpError option is set to terminate (this is the default if the configuration option is not specified), the program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the information in the message to correct the error. Restart the program if it ended or if Program Call support is needed. Program Call support is used by RACF change logging and Policy Directory extended operations. If the problem persists, contact the service representative.

Problem determination: Not applicable.

Source: LDAP

Module: None.
Routing code: None.
Descriptor code: None.

GLD1131E Program Call support not activated because another server already provides Program Call support.

Explanation: Another LDAP server is already running with Program Call support activated. Only one LDAP server on each system can provide Program Call support.

Example: None.

System action: If the srvStartUpError option in the LDAP server configuration file is set to ignore, the LDAP server continues to run with those backends that successfully start. If the srvStartUpError option is set to terminate (this is the default if the configuration option is not specified), the program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: If Program Call support is needed on this LDAP server, stop the other LDAP server that is running with Program Call support and remove the listen option for Program Call support from the LDAP server configuration file for the other server. Then restart both LDAP servers. If Program Call support is not needed on this LDAP server, remove the listen option for Program Call support from the LDAP server configuration file for this LDAP server.

Problem determination: Not applicable.

Source: LDAP

Module: None.
Routing code: None.
Descriptor code: None.
GLD1132E  Program Call termination failed: Return code return_code, Reason code reason_code.

Explanation: The LDAP server is unable to stop the Program Call support. The return code has the following values:

101 ESTAEX cancel failed. The reason code is the ESTAEX return code. Refer to the description of ESTAEX in z/OS MVS Programming, Authorized Assembler Services Reference EDT-IXG for more information on the error.

102 IEANTDL failed. The reason code is the IEANTDL return code. Refer to the description of IEANTDL in z/OS MVS Programming, Authorized Assembler Services Reference EDT-IXG for more information on the error.

103 Unable to obtain control area lock. This indicates another task abnormally ended while holding the lock.

In the message text:

return_code
Return code

reason_code
Reason code

Example: None.

System action: The Program Call interface is not available. If the srvStartUpError option in the LDAP server configuration file is set to ignore, the LDAP server continues to run with those backends that successfully start. If the srvStartUpError option is set to terminate (this is the default if the configuration option is not specified), the program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the information in the earlier message to correct the error. Restart the program if it ended or if Program Call support is needed. Program Call support is used by RACF change logging and Policy Directory extended operations. If the problem persists, contact the service representative.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1135I Sysplex status

Explanation: This message is displayed in response to the LDAP server DISPLAY XCF operator modify command. The remaining lines in this multi-line message display the status of each LDAP server in the cross-system group.

Example: None.

System action: The LDAP server continues.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1133A Unable to start the Program Call support.

Explanation: The LDAP server is unable to initialize the Program Call support. A previous message identifies the reason for the failure.

Example: None.

System action: The LDAP server continues.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1136I Cross-system services are not available.

Explanation: This message is displayed in response to the LDAP server DISPLAY XCF operator modify command.
command when the LDAP server is not a member of a cross-system group.

Example: None.
System action: The LDAP server continues.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1137A Unable to initialize sysplex services.

Explanation: The LDAP server is unable to initialize the sysplex support. A previous message indicates the reason for the failure.

Example: None.
System action: The LDAP server ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1138E Cross-system group name value is not valid.

Explanation: The cross-system group name specified by the serverSysplexGroup option in the LDAP server configuration file is not valid. A cross-system group name is 1-8 characters and consists of letters (A-Z), numbers (0-9) and special characters (#, @, $). The special characters must be in the IBM-1047 code page.

In the message text:

value
  Cross-system group name

Example: None.
System action: The program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1139E Not authorized to join cross-system group group_name.

Explanation: The LDAP server is not authorized to join the cross-system group. The user ID associated with the LDAP server must have at least READ access to the GLD.XCF.GROUP.group_name resource in the FACILITY class. The group name is specified by the serverSysplexGroup option in the LDAP server configuration file.

In the message text:

group_name
  Cross-system group name

Example: None.
System action: The LDAP server ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Give the LDAP server user ID at least READ access to the cross-system resource. Then restart the program.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.
GLD1140E Cross-system group group_name is not defined.

Explanation: The cross-system group is not defined to the external security manager. The cross-system group for the LDAP server must have a profile in the FACILITY class. The resource name is GLD.XCF.GROUP.group_name, where the group name is specified by the serverSysplexGroup option in the LDAP server configuration file. For example, if the cross-system group name is LDAP6, then the resource name is GLD.XCF.GROUP.LDAP6.

In the message text:

In the message text:

Example: None.

System action: The LDAP server ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Create the profile to define the cross-system group to the external security manager. Give the LDAP server user ID at least READ access to the resource. Then restart the program.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1141E XCF initialization failed: Return code return_code, Reason code reason_code.

Explanation: The LDAP server is unable to initialize the cross-system coupling facility (XCF) support. The return code has the following values:

1 **IXCJOIN** failed. The reason code contains the IXCJOIN return code in the upper 16 bits and the IXCJOIN reason code in the lower 16 bits. Refer to the description of IXCJOIN in z/OS MVS Programming: Sysplex Services Reference for more information on the error.

2 **IXCQUERY** failed. The reason code contains the IXCQUERY return code in the upper 16 bits and the IXCQUERY reason code in the lower 16 bits. Refer to the description of IXCQUERY in z/OS MVS Programming: Sysplex Services Reference for more information on the error.

3 **IXCSETUS** failed. The reason code contains the IXCSETUS return code in the upper 16 bits and the IXCSETUS reason code in the lower 16 bits. Refer to the description of IXCSETUS in z/OS MVS Programming: Sysplex Services Reference for more information on the error.

In the message text:

In the message text:

Example: None.

System action: The LDAP server ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the information in the message to correct the error. Then restart the program. If the problem persists, contact the service representative.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1142E System system_name is already active in cross-system group group_name.

Explanation: Another LDAP server on the same system is already a member of the cross-system group. Only one LDAP server on each system in the sysplex can be a member of a particular cross-system group. The cross-system group name is specified by the serverSysplexGroup option in the LDAP server configuration file.

In the message text:

In the message text:

Example: None.

System action: The LDAP server ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Either stop the other LDAP...
server or specify a different cross-system group in the `serverSysplexGroup` option in the LDAP server configuration file for this LDAP server. Then restart the program.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD1143E** XCF termination failed: Return code `return_code`, Reason code `x'reason_code'`.

**Explanation:** The LDAP server is unable to end the cross-system coupling facility (XCF) support. The return code has the following values:

1. `IXCLEAVE` failed. The reason code contains the `IXCLEAVE` return code in the upper 16 bits and the `IXCLEAVE` reason code in the lower 16 bits. Refer to the description of `IXCLEAVE` in [z/OS MVS Programming: Sysplex Services](z/OS MVS Programming: Sysplex Services) for more information on the error.

In the message text:

```
return_code
  Return code
reason_code
  Reason code
```

**Example:** None.

**System action:** The LDAP server ends.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** None.

**Problem determination:** Not applicable.

---

**GLD1145I** SSL support is not active.

**Explanation:** The LDAP server `REFRESH SSL` operator modify command cannot be processed because SSL support is not enabled in the LDAP server. Either SSL support is not configured or the LDAP server is unable to initialize the SSL environment.

**Example:** None.

**System action:** The LDAP server continues.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** None.

**Problem determination:** Not applicable.

---

**GLD1146I** System `system_name` has joined LDAP cross-system group `group_name`.

**Explanation:** The LDAP server running on the indicated system has joined the LDAP cross-system group. Sysplex services are now active for that server. This message is displayed by each active LDAP server when a new LDAP server joins the cross-system group.
In the message text:

system_name
  System name

group_name
  Cross-system group name

Example: None.

System action: The LDAP server continues.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1147I System system_name has left LDAP cross-system group group_name.

Explanation: The LDAP server running on the indicated system has left the LDAP cross-system group. Sysplex services are no longer active for that server. This message is displayed by each active LDAP server when an LDAP server leaves the cross-system group.

In the message text:

system_name
  System name

group_name
  Cross-system group name

Example: None.

System action: The LDAP server continues.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1148E Unable to set cross-system group owner: Return code return_code, Reason code x’reason_code’.

Explanation: The LDAP server is unable to set the owner for the cross-system group. The return code has the following values:

1 IXCSETUS failed. The reason code contains the IXCSETUS return code in the upper 16 bits and the IXCSETUS reason code in the lower 16 bits. Refer to the description of IXCSETUS in z/OS MVS Programming: Sysplex Services Reference for more information on the error.

In the message text:

return_code
  Return code
reason_code
  Reason code

Example: None.

System action: The LDAP server continues. Update operations to the LDAP server probably fail. Search operations may succeed.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Stop the LDAP server. Use the information in the message to correct the error. Then restart the LDAP server. If the problem persists, contact the service representative.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1149I System system_name is leaving the sysplex.

Explanation: The LDAP server on the indicated system is leaving the sysplex and LDAP cross-system services are no longer available to this server.

In the message text:

system_name
  System name

Example: None.

System action: The LDAP server issuing this message continues. If the server leaving the sysplex is the sysplex owner, another server will become the sysplex owner. During that transition period, update
operations to shared directories may fail.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** None.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD1150E Unable to send cross-system message:**

Return code `return_code`, Reason code `reason_code`

**Explanation:** The LDAP server is unable to send a message to another member of the LDAP cross-system group. The return code has the following values:

2. XCF services are not available.
3. No response received.
4. Insufficient storage available on source system.
5. Insufficient storage available on target system.
6. Target member not defined.
7. Target member not active.
8. IXCMSGO failed. The reason code contains the IXCMSGO return code in the upper 16 bits and the IXCMSGO reason code in the lower 16 bits. Refer to the description of IXCMSGO in [z/OS MVS Programming: Sysplex Services Reference](https://www.ibm.com/support/docview.wss?uid=swg21385871) for more information on the error.
9. IXCMSGI failed on the target system. The reason code contains the IXCMSGI return code in the upper 16 bits and the IXCMSGI reason code in the lower 16 bits. Refer to the description of IXCMSGI in [z/OS MVS Programming: Sysplex Services Reference](https://www.ibm.com/support/docview.wss?uid=swg21385871) for more information on the error.
10. IXCMSGI failed on the source system. The reason code contains the IXCMSGI return code in the upper 16 bits and the IXCMSGI reason code in the lower 16 bits. Refer to the description of IXCMSGI in [z/OS MVS Programming: Sysplex Services Reference](https://www.ibm.com/support/docview.wss?uid=swg21385871) for more information on the error.
11. Message canceled or timed out.
12. Unknown notification response.

In the message text:

```
return_code
   Return code
reason_code
   Reason code
```

**Example:** None.

**System action:** The LDAP server may continue or it may end, depending on which function attempted to send a message.

---

**GLD1151E Unable to reply to cross-system message:**

Return code `return_code`, Reason code `reason_code`

**Explanation:** The LDAP server is unable to reply to a message received from another member of the LDAP cross-system group. The return code has the following values:

2. XCF services are not available.
4. Insufficient storage available on source system.
8. IXCMSGO failed. The reason code contains the IXCMSGO return code in the upper 16 bits and the IXCMSGO reason code in the lower 16 bits. Refer to the description of IXCMSGO in [z/OS MVS Programming: Sysplex Services Reference](https://www.ibm.com/support/docview.wss?uid=swg21385871) for more information on the error.
11. Message canceled.
12. Unknown notification response.

In the message text:

```
return_code
   Return code
reason_code
   Reason code
```

**Example:** None.

**System action:** The LDAP server will continues.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Use the information in the message to correct the error. Stop the server and then restart the program. If the problem persists, contact the service representative.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

**GLD1152E** Time limit exceeded while loading schema from group owner.

Explanation: The LDAP server waits a maximum of 30 seconds after requesting a copy of the schema from the LDAP cross-system owner. The owning LDAP server is not responding to cross-system requests.

Example: None.

System action: If the error occurs during LDAP server initialization, the server ends.

- If the error occurs when requesting the new schema after it is modified by the schema owner, the server continues with its current (unmodified) schema. Add and modify operations that involve the modified schema elements may fail on this server.
- If the error occurs while sending a schema modify request to the schema owner, the server continues but the schema modify request fails.

Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Determine the owning system by issuing the LDAP server `DISPLAY XCF` operator modify command for the LDAP server reporting the error. The command output indicates which LDAP server is the group owner. Then issue the `DISPLAY XCF` operator modify command for the owning LDAP server and verify that this server is really the group owner. Restart the owning LDAP server if there is no response to the `DISPLAY XCF` operator modify command. Restart this LDAP server if it ended or if the updated schema is needed. Retry the schema modify operation if it failed. If the problem persists, contact the service representative.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

**GLD1153E** The schema owner cannot be contacted.

Explanation: The LDAP server is unable to contact the schema owner to obtain a copy of the current schema.

Example: None.

System action:
- If the error occurs during LDAP server initialization, the server ends.
- If the error occurs when requesting the new schema after it is modified by the schema owner, the server continues with its current (unmodified) schema. Add and modify operations that involve the modified schema elements may fail on this server.
- If the error occurs while sending a schema modify request to the schema owner, the server continues but the schema modify request fails.

Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Determine the owning system by issuing the LDAP server `DISPLAY XCF` operator modify command for the LDAP server reporting the error. The command output indicates which LDAP server is the group owner. Then issue the `DISPLAY XCF` operator modify command for the owning LDAP server and verify that this server is really the group owner. Restart the owning LDAP server if there is no response to the `DISPLAY XCF` operator modify command. Restart this LDAP server if it ended or if the updated schema is needed. Retry the schema modify operation if it failed. If the problem persists, contact the service representative.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

**GLD1154E** A database name is required in multi-server mode.

Explanation: A backend name must be specified on the `database` option for an LDBM, GDBM or TDBM backend in the LDAP server configuration file when multi-server mode is enabled for the backend. Multi-server mode is enabled by specifying the `serverSysplexGroup` option in the global section and the `multiserver` option in the backend section of the LDAP server configuration file.

Example: None.
System action: The program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Specify a backend name on the database option in the LDAP server configuration file. Then restart the program.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1155E The database name may not exceed 8 characters in multi-server mode.

Explanation: The backend name for an LDBM, GDBM or TDBM backend has a maximum length of 8 characters when multi-server mode is enabled for the backend. Multi-server mode is enabled by specifying the serverSysplexGroup option in the global section and the multiserver on option in the backend section of the LDAP server configuration file.
Example: None.
System action: The program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1156E A fully-qualified path must be specified for the database directory.

Explanation: The directory specified by the databaseDirectory option in the LDBM or GDBM backend section of the LDAP server configuration file must be a fully-qualified path. That is, the path must start with a ‘/’.
Example: None.
System action: The program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Specify a valid backend name on the database option in the LDAP server configuration file. Then restart the program.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1157E Multi-server mode requires cross-system services.

Explanation: The multiserver option in a backend section of the LDAP server configuration file cannot be set on unless cross-system services are configured. Cross-system services are configured by specifying the name of the LDAP cross-system group on the serverSysplexGroup option in the global section of the configuration file.
Example: None.
System action: The program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Either add the serverSysplexGroup option to the global section of the configuration file or set the multiserver option off in the backend section (or remove it from the backend section). Then restart the program.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1158E Multi-server change log support is required.

Explanation: The multiserver option in the GDBM section of the LDAP server configuration file must be set on because there is another backend section that has multiserver set on. When GDBM is configured, all GDBM, TDBM, LDBM, and CDBM backends must have
the same setting for the multiserver option.

Example: None.

System action: The program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Either set the multiserver option in the GDBM section of the LDAP server configuration file on or set all the multiserver options off. Then restart the program.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1159E Multi-server support required for all LDBM, TDBM, CDBM, and GDBM backends.

Explanation: If the GDBM backend is configured, the multiserver option in the LDBM, TDBM, or CDBM section of the LDAP server configuration file must be set on because the GDBM section has multiserver set on. If the CDBM backend is configured, the multiserver option in the LDBM, TDBM, or GDBM section of the LDAP server configuration file must be set on because the CDBM section has multiserver set on. When a GDBM or CDBM backend is configured, all GDBM, TDBM, LDBM, and CDBM backends must have the same setting for the multiserver option.

Example: None.

System action: The program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1160E Unable to initialize the LDAP client SSL support: Error return_code, Reason reason_code.

Explanation: The LDAP server is unable to initialize the LDAP client SSL support. Refer to the description of ldap_ssl_client_init() in [IBM Tivoli Directory Server Client Programming for z/OS](https://www.ibm.com) for more information on the error.

In the message text:

return_code

Return code from ldap_ssl_client_init()

reason_code

Reason code from ldap_ssl_client_init()

Example: None.

System action: LDAP server initialization continues if the tcpTerminate option in the LDAP server configuration file is set to recover (this is the default if the configuration option is not specified). In this case, SSL support is not available until the error is corrected and the server is restarted. If the tcpTerminate option is set to terminate, the program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the information in the message to correct the error. If SSL connections are not needed, remove the sslKeyRingFile option from the LDAP server configuration file. Restart the program if it ended or if SSL connections are needed.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1161E The option1 configuration option requires the option2 configuration option.

Explanation: Certain LDAP server configuration options and values are dependant on other configurations options being specified to provide complete configuration information. Note that the value of some configuration options may be specified as a command-line parameter when starting the LDAP server (in this case, the command-line parameter overrides the value in the configuration file).

In the message text:

option1

LDAP server configuration option

option2

LDAP server configuration option
**option1**
 LDAP server configuration option

**Example:** None.
**System action:** The program ends.
**Operator response:** None.
**User response:** None.

**System programmer response:** None.
**Administrator response:** If **option1** is needed, then either add **option2** to the configuration file or change the value of **option2** to support **option1**. If **option1** is not needed, then either remove **option1** or change its value to one that does not require **option2**. Then restart the program.

**Problem determination:** Not applicable.
**Source:** LDAP
**Module:** None.
**Routing code:** None.
**Descriptor code:** None.
**Automation:** Not applicable.

GLD1162E  Configuration options **option1** and **option2** are mutually exclusive.

**Explanation:** The two options indicated in the message cannot both be specified in the same LDAP server configuration file.

In the message text:

**option1**
 Configuration option one

**option2**
 Configuration option two

**Example:** None.
**System action:** The program ends.
**Operator response:** None.
**User response:** None.

**System programmer response:** None.

---

GLD1163I  Replication status

**Explanation:** This message is displayed in response to the LDAP server **DISPLAY REPLICAS** operator modify command. The remaining lines in this multi-line message display the status of each peer or replica server.

**Example:** None.
**System action:** The LDAP server continues.
**Operator response:** None.
**User response:** None.

**System programmer response:** None.
**Administrator response:** None.
**Problem determination:** Not applicable.
**Source:** LDAP
**Module:** None.
**Routing code:** None.
**Descriptor code:** None.
**Automation:** Not applicable.

---

GLD1164I  No replication status.

**Explanation:** This message is displayed in response to the LDAP server **DISPLAY REPLICAS** operator modify command when there are no peer or replicas servers. It can also be displayed if the LDAP server that received the operator modify command is in a sysplex but is not the sysplex owner. Only the LDAP server that is the cross-system group owner has information about peer and replica servers.

**Example:** None.
**System action:** The LDAP server continues.
**Operator response:** None.
**User response:** None.

**System programmer response:** None.
**Administrator response:** None.
**Problem determination:** Not applicable.
**Source:** LDAP
**Module:** None.
**Routing code:** None.
**Descriptor code:** None.
**Automation:** Not applicable.
GLD1165I The LDAP server is in maintenance mode.

Explanation: The LDAP server has entered maintenance mode either because the LDAP server MAINTMODE ON operator modify command is issued or because the -m command-line parameter is specified when the LDAP server is started.

Example: None.

System action: The LDAP server changes to maintenance mode. Update requests are accepted only from users who are bound with the distinguished name specified on the adminDN, masterServerDN, or peerServerDN options in the LDAP server configuration file.

Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1166I LDAP server maintenance mode has ended.

Explanation: The LDAP server is no longer in maintenance mode, due to usage of the LDAP server MAINTMODE OFF operator modify command.

Example: None.

System action: The LDAP server changes to regular mode. Update requests are now accepted from all users.

Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1167I The LDAP server is already in maintenance mode.

Explanation: The LDAP server MAINTMODE ON operator modify command is issued when the LDAP server is already in maintenance mode.

Example: None.

System action: The LDAP server continues in maintenance mode.

Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1168I The LDAP server is not in maintenance mode.

Explanation: The LDAP server MAINTMODE OFF operator modify command is issued when the LDAP server is not in maintenance mode.

Example: None.

System action: The LDAP server continues in regular mode.

Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1169E The option configuration option must be the same for all TDBM and DB2-based GDBM backends.

Explanation: The indicated option must be specified, with the same value, in all TDBM and DB2-based GDBM backends in the LDAP server configuration file.
In the message text:

option
  LDAP server configuration option

Example: None.
System action: The program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Either specify the same option and value in all TDBM and DB2-based GDBM backend sections or remove the option from all TDBM and DB2-based GDBM backend sections in the LDAP server configuration file. Then restart the program.

Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1171E Native return code return_code, SQL state state, SQL message: error_text

Explanation: The LDAP server encountered an error while performing a DB2 database operation. Refer to DB2 Messages and Codes for more information on DB2 errors.

In the message text:

return_code
  Native return code
state
  SQL state
error_text
  SQL message text

Example: None.
System action: The backend does not start. If the srvStartUpError option in the LDAP server configuration file is set to ignore, the LDAP server continues to run with those backends that successfully start. If the srvStartUpError option is set to terminate (this is the default if the configuration option is not specified), the program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Use the information in this message and the additional messages to correct the error. Restart the program if it ended or if the backend is needed. If the problem is unable to be resolved, contact the DB2 database administrator.

Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1172E Error code error_code received for ODBC function name.

Explanation: The LDAP server encountered an error for an ODBC (Open Database Connectivity) function. This message may be followed by additional messages providing further information on the error. Refer to DB2 ODBC Guide and Reference for more information on ODBC errors.

In the message text:

error_code
  Error code
name
  ODBC function name

Example: None.
System action: The backend does not start. If the srvStartUpError option in the LDAP server configuration file is set to ignore, the LDAP server continues to run with those backends that successfully start. If the srvStartUpError option is set to terminate (this is the default if the configuration option is not specified), the program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Use the information in this message and the additional messages to correct the error. Restart the program if it ended or if the backend is needed. If the problem is unable to be resolved, contact the DB2 database administrator.

Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.
GLD1173E  DB2 based backends are not supported for 64-bit addressing mode.

**Explanation:** A DB2-based backend cannot be configured in an LDAP server that will run in 64-bit mode. DB2 does not support 64-bit mode processing.

**Example:** None.

**System action:** The backend does not start. If the `srvStartUpError` option in the LDAP server configuration file is set to `ignore`, the LDAP server continues to run with those backends that successfully start. If the `srvStartUpError` option is set to `terminate` (this is the default if the configuration option is not specified), the program ends. The utility ends regardless of the option value.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Either remove the DB2-based backend from the LDAP server configuration file or run the LDAP server or utility in 31-bit mode.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

GLD1175E  Unable to establish DB2 monitor connection: type return code return_code, reason code reason_code.

**Explanation:** The LDAP server is unable to establish a connection with the DB2 server. The return code and reason code are from the `CONNECT` function if the type is DSNALI or from the `IDENTIFY` function if the type is DSNRLI. Refer to [DB2 Application Programming and SQL Guide](#) for more information on the error.

In the message text:

- **type**
  - DB2 attachment facility type
- **return_code**
  - Return code
- **reason_code**
  - Reason code

**Example:** None.

**System action:** If the error occurs in the LDAP server during initialization, then the DB2-based backends do not start. If the `srvStartUpError` option in the LDAP server configuration file is set to `ignore`, the LDAP server continues to run with those backends that successfully start. If the `srvStartUpError` option is set to `terminate` (this is the default if the configuration option is not specified), the program ends. If the error occurs after server initialization, the DB2-based backends are disabled and all requests to those backends are rejected.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Use the information in the message to correct the error. Verify the values in the DSNNOINI configuration file used by the LDAP server and ensure that DB2 is running. Restart the program if it ended or if the DB2 backends are needed. If the problem is unable to be resolved, contact the DB2 database administrator.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.
GLD1176E Unable to open the encryption keys dataset: error_code/reason_code - error_text

Explanation: The LDAP server or utility is unable to open the encryption keys dataset. Refer to the description of fopen() in z/OS XL C/C++ Run-Time Library Reference for more information on the error. The keys dataset is specified by the -k parameter on the utility command line. For the LDAP server, and for the utility when the -k parameter is not specified, the keys dataset can be specified in the LDAPKEYS DD statement in the JCL used to start the server or utility.

In the message text:
- error_code
  Error code from fopen()
- reason_code
  Reason code from fopen()
- error_text
  Error text corresponding to the error code

Example: None.

System action: The utility ends. The LDAP server continues, but encryption and decryption of values using a key in the key dataset may fail. In particular, this may result in bind failures if the userPassword attribute value cannot be decrypted.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the information in the message to correct the error. Verify that the encryption keys dataset exists and can be accessed by the LDAP server or utility. Then restart the program.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1177E Unable to read the encryption keys dataset: error_code/reason_code - error_text

Explanation: The LDAP server or utility is unable to read the encryption keys dataset. Refer to the description of fgets() in z/OS XL C/C++ Run-Time Library Reference for more information on the error. The keys dataset is specified by the -k parameter on the utility command line. For the LDAP server, and for the utility when the -k parameter is not specified, the keys dataset can be specified in the LDAPKEYS DD statement in the JCL used to start the server or utility.

In the message text:
- error_code
  Error code from fgets()
- reason_code
  Reason code from fgets()
- error_text
  Error text corresponding to the error code

Example: None.

System action: The utility ends. The LDAP server continues, but encryption and decryption of values using a key in the key dataset may fail. In particular, this may result in bind failures if the userPassword attribute value cannot be decrypted.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1178E The schema owner busy, retrying.

Explanation: The LDAP server which is the owner of the schema in the LDAP cross-system group in the sysplex is currently busy and cannot send the schema.

Example: None.

System action: The LDAP server continues and retries the request.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: If the problem persists, restart the LDAP server that owns the schema. If the problem still persists, contact the service representative.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.
GLD1179E The *option* configuration option value must be different for each *type* backend.

Explanation: The indicated option in the LDAP server configuration file must have a unique value for each backend section in which it is included.

In the message text:

*option*  
Configuration option name

*type*  
Backend type

Example: The *dbuserid* value must be unique for each DB2-based backend (TDBM and DB2-based GDBM). Different backends cannot share a DB2 database.

System action: The program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Correct the LDAP server configuration file so that the option value is unique throughout the LDAP server configuration file. Then restart the program.

Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

---

GLD1180I Activity log option processed: *option*.

Explanation: The indicated activity log option specified on an LDAP server LOG operator modify command has successfully been processed.

In the message text:

*option*  
Activity log option

Example: None.

System action: The LDAP server continues, with activity logging using the indicated log option.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.

Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

---

GLD1181E Incorrect LDAP server activity log option specified: *option*.

Explanation: An activity log option that is not valid is specified on an LDAP server LOG operator modify command.

In the message text:

*option*  
Incorrect activity log option

Example: None.

System action: The LDAP server continues, with no change to activity logging.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Retry the LOG operator modify command using a valid activity log option.

Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

---

GLD1182A Unable to initialize activity logging.

Explanation: The LDAP server cannot initialize the activity logging facility. A previous message indicates the reason for the failure.

Example: None.

System action: The LDAP server ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Use the information in the earlier message to correct the error. Then restart the program. If the problem persists, contact the service representative.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1184E Unable to start activity logging.
Explanation: The LDAP server cannot open the activity log file. The name of the file is specified by the logfile option in the LDAP server configuration file. The default logfile name is /etc/ldap/gldlog.output if the configuration option is not specified.
Example: None.
System action: The LDAP server continues, but activity logging is not done.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Verify that the LDAP server has write access to the logfile and to its directory if the file does not exist. Restart the program if activity logging is needed.

Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1186E Incorrect LDAP server audit option specified: option.
Explanation: An audit option that is not valid is specified on an LDAP server AUDIT operator modify command.
In the message text:
option Incorrect audit option
Example: None.
System action: The LDAP server continues, with no change to auditing.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Retry the AUDIT operator modify command using a valid audit option.

Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1185I LDAP server audit option processed: option.
Explanation: The indicated audit option specified on an LDAP server AUDIT operator modify command has successfully been processed.
In the message text:
option
Audit option
Example: None.
System action: The LDAP server continues, with auditing using the indicated option.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.

GLD1187I LDAP server SMF auditing ON.
Explanation: LDAP server auditing is activated, either via the audit option in the LDAP server configuration file or the LDAP server AUDIT operator modify command.
Example: None.
System action: The LDAP server continues, with auditing activated.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: The LDAP server DISPLAY AUDIT operator modify command can be used to review the current LDAP server auditing settings. Use the LDAP server AUDIT operator modify command to make any necessary updates.
Problem determination:  Not applicable.
Source:     LDAP
Module:    None.
Routing code:  None.
Descriptor code:  None.
Automation:  Not applicable.

GLD1188I  LDAP server SMF auditing OFF.
Explanation:  LDAP server auditing has been deactivated, either via the audit option in the LDAP server configuration file or the LDAP server AUDIT operator modify command.
Example:  None.
System action:  The LDAP server continues, without auditing activated.
Operator response:  None.
User response:  None.
System programmer response:  None.
Administrator response:  If auditing is needed, use the LDAP server AUDIT operator modify command to turn LDAP server auditing on.

Problem determination:  Not applicable.
Source:     LDAP
Module:    None.
Routing code:  None.
Descriptor code:  None.
Automation:  Not applicable.

GLD1189I  LDAP server audit settings updated with values values.
Explanation:  The LDAP server is updated with the indicated audit settings.  The audit settings are specified either in the audit option in the LDAP server configuration file or on the LDAP server AUDIT operator modify command.
Example:  None.
System action:  The LDAP server continues, with auditing using the indicated values.
Operator response:  None.
User response:  None.
System programmer response:  None.
Administrator response:  If auditing is needed, use the LDAP server AUDIT operator modify command to turn LDAP server auditing on.

Problem determination:  Not applicable.
Source:     LDAP
Module:    None.
Routing code:  None.
Descriptor code:  None.
Automation:  Not applicable.

GLD1190I  Audit status
Explanation:  This message displays the current LDAP server audit settings as a result of issuing the LDAP server DISPLAY AUDIT operator modify command.
Example:  None.
System action:  The LDAP server continues.
Operator response:  None.
User response:  None.
System programmer response:  None.
Administrator response:  The LDAP server DISPLAY AUDIT operator modify command can be used to review the current LDAP server auditing settings. Use the LDAP server AUDIT operator modify command to make any necessary updates.

Problem determination:  Not applicable.
Source:     LDAP
Module:    None.
Routing code:  None.
Descriptor code:  None.
Automation:  Not applicable.

GLD1191I  LDAP server auditing is only available on V1R7 and above.
Explanation:  LDAP server auditing is unavailable on this level of z/OS.
Example:  None.
System action:  The LDAP server continues. No audit records are created.
Operator response:  None.
User response:  None.
System programmer response:  None.
Administrator response:  Remove the audit option from the LDAP server configuration file and do not issue the LDAP server AUDIT or DISPLAY AUDIT operator modify commands.

Problem determination:  Not applicable.
Source:     LDAP
Module:    None.
Routing code:  None.
Descriptor code:  None.
Automation:  Not applicable.

GLD1193E  XCF send for name timed out, retrying.
Explanation:  XCF was unable complete the send of a request from the indicated backend to the owner of the resource in the LDAP server cross-system group.
In the message text:

name  Backend name

Example:  None.
System action:  The LDAP server continues and retries the request.
Operator response:  None.
User response:  None.
System programmer response:  None.
Administrator response:  If the problem persists, restart this LDAP server and the LDAP server that owns the resource. If the problem still persists, contact the service representative.
Problem determination:  Not applicable.
Source:  LDAP
Module:  None.
Routing code:  None.
Descriptor code:  None.
Automation:  Not applicable.

GLD1195A  Unable to start ctrace: Error return_code, Reason reason_code.
Explanation:  The LDAP server cannot identify its component tracing facility to the z/OS CTRACE subsystem. Refer to the description of CTRACE DEFINE in z/OS MVS Programming: Authorized Assembler Services Reference ALE-DYN for more information on the error.
In the message text:

return_code  Return code from CTRACE DEFINE
reason_code  Reason code from CTRACE DEFINE

Example:  None.
System action:  The program continues, but component trace is not used.
Operator response:  None.
User response:  None.
System programmer response:  None.
Administrator response:  Use the information in the message to correct the error. Restart the program if component tracing is needed. If the problem persists, contact the service representative.
Problem determination:  Not applicable.
Source:  LDAP
Module:  None.
Routing code:  None.
Descriptor code:  None.
Automation:  Not applicable.

GLD1194I  Component Trace has been successfully started for GLDSRVR.
Explanation:  Component TRACE (CTRACE) has successfully started on the LDAP server in CTRACE component name GLDSRVR. The LDAP server writes CTRACE records to a subnode under the GLDSRVR component name.
Example:  None.
System action:  The LDAP server continues.
Operator response:  None.
User response:  None.
System programmer response:  None.
Administrator response:  None.
Problem determination:  Not applicable.
Source:  LDAP
Module:  None.
Routing code:  None.
Descriptor code:  None.
Automation:  Not applicable.

GLD1196I  Active trace points only written to memory now for GLDSRVR.
Explanation:  The LDAP debug output generated by the LDAP server will only be written to memory.
Example:  None.
System action:  The LDAP server continues.
Operator response:  None.
User response:  None.
System programmer response:  None.
Administrator response:  None.
Problem determination:  Not applicable.
Source:  LDAP
GLD1197I Active trace points now written to memory and trace file for GLDSRVR.

Explanation: The LDAP debug output generated by the LDAP server will be written to both memory and the output trace file. The LDAP_DEBUG_FILENAME environment variable specifies the output trace filename. If the LDAP_DEBUG_FILENAME environment variable is not specified, then LDAP debug output goes to stdout.

Example: None.

System action: The LDAP server continues.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1198E Unable to open logfile filename: error_code/reason_code - error_text

Explanation: The LDAP server is unable to open a log file. Refer to the description of fopen() in z/OS XL C/C++ Run-Time Library Reference for more information on the error. For activity logging, the name of the file is specified by the logfile option in the LDAP server configuration file. The default logfile name is /etc/ldap/gldlog.output if the configuration option is not specified. For replication error logging, the name of the file is specified by the ibm-slapdLog attribute in the replica entry.

In the message text:

filename
  Logfile file name
error_code
  Error code from fopen()
reason_code
  Reason code from fopen()
error_text
  Error text corresponding to the error code

Example: None.

System action: The LDAP server continues, but logging using this file is not done. For replication, replication to the replica server using this log may stall.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Verify that the LDAP server has write access to the log file and to its directory if the file does not exist. Restart the program if logging is needed.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1199I The backend named name has been set to read-only mode.

Explanation: The indicated backend is successfully set to read-only mode after the LDAP server BACKEND operator modify command is issued.

In the message text:

name
  Backend name

Example: None.

System action: The program continues.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1200I The backend named name has been set to read/write mode.

Explanation: The indicated backend is successfully set to read/write mode after the LDAP server BACKEND operator modify command is issued.

In the message text:

name
  Backend name

Example: None.

System action: The program continues.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.
GLD1201I The backend named name is already set to read-only mode.

Explanation: The indicated backend is already running in read-only mode. The LDAP server BACKEND operator modify command is ignored.

In the message text:

name
   Backend name

Example: None.
System action: The program continues.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1202I The backend named name is already set to read/write mode.

Explanation: The indicated backend is already running in read/write mode. The LDAP server BACKEND operator modify command is ignored.

In the message text:

name
   Backend name

Example: None.
System action: The program continues.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1203E Incorrect LDAP server backend option specified: options.

Explanation: An option that is not valid is specified on the LDAP server BACKEND operator modify command.

In the message text:

options
   Options specified on LDAP server BACKEND operator modify command

Example: None.
System action: The program continues.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Retry the BACKEND operator modify command with valid options.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1204I Schedule commit for all backends completed.

Explanation: A database commit resulting from an LDAP server COMMIT operator modify command has successfully completed on all backends.

Example: None.
System action: The program continues.
Operator response: None.
**User response:** None.
**System programmer response:** None.
**Administrator response:** None.
**Problem determination:** Not applicable.
**Source:** LDAP
**Module:** None.
**Routing code:** None.
**Descriptor code:** None.
**Automation:** Not applicable.

---

**GLD1205I** Schedule commit for all backends failed.

**Explanation:** A database commit resulting from an LDAP server COMMIT or BACKEND operator modify command has failed. A previous message may indicate the reason for the failure.

**Example:** None.

**System action:** The program continues, but the operator modify command fails. One or more databases are not been committed.

**Operator response:** None.
**User response:** None.
**System programmer response:** None.
**Administrator response:** None.
**Problem determination:** Not applicable.
**Source:** LDAP
**Module:** None.
**Routing code:** None.
**Descriptor code:** None.
**Automation:** Not applicable.

---

**GLD1206I** Only sysplex owner can perform command.

**Explanation:** A database commit resulting from an LDAP server COMMIT or BACKEND operator modify command has failed because the LDAP server is not the owner of the backend database in the sysplex. Only the LDAP server that is the database owner in the LDAP cross-system group can process this command.

**Example:** None.

**System action:** The program continues, but the operator modify command fails. One or more databases are not been committed.

**Operator response:** None.
**User response:** None.
**System programmer response:** None.
**Administrator response:** None.
**Problem determination:** Not applicable.
**Source:** LDAP
**Module:** None.
**Routing code:** None.
**Descriptor code:** None.
**Automation:** Not applicable.

---

**GLD1207E** Type backend specified for a non-type database.

**Explanation:** The LDAP server or utility found that the backend type specified on a database option in the LDAP server configuration file does support the backend DLL specified on the configuration option.

In the message text:

*type*  
LDAP server backend type

**Example:** None.

**System action:** The backend does not start. If the srvStartUpError option in the LDAP server configuration file is set to ignore, the LDAP server continues to run with those backends that successfully start. If the srvStartUpError option is set to terminate (this is the default if the configuration option is not specified), the program ends. The utility ends regardless of the option value.

**Operator response:** None.
**User response:** None.
**System programmer response:** None.
**Administrator response:** Correct the database option in the LDAP server configuration file so that the specified DLL and backend type match. Restart the program if it ended or if the backend is needed.
**Problem determination:** Not applicable.
**Source:** LDAP
**Module:** None.
**Routing code:** None.
**Descriptor code:** None.
**Automation:** Not applicable.
GLD1208E  Configuration for type backend named name failed.

Explanation:   The LDAP server backend indicated in the message cannot be started due to a configuration error. A previous message indicates the reason for the failure.

In the message text:

   type  Backend type
   name  Backend name

Example:   None.

System action:   The program ends.

Operator response:   None.

User response:   None.

System programmer response:   None.

Administrator response:   Use the information in the earlier message to correct the error in the LDAP server configuration file. Then restart the program.

Problem determination:   Not applicable.

Source:   LDAP

Module:   None.

Routing code:   None.

Descriptor code:   None.

Automation:   Not applicable.

GLD1209E  Unable to obtain file lock with fcntl() on filename: error_code reason_code - error_text

Explanation:   The LDAP server or utility is not able to obtain a read or write lock on the file indicated in the message. Refer to the description of fcntl() in z/OS XL C/C++ Run-Time Library Reference for more information on the error.

In the message text:

   filename  File name
   error_code  Error code from fcntl()
   reason_code  Reason code from fcntl()
   error_text  Error text corresponding to the error code

Example:   None.

System action:   If a read or write file lock error is encountered during LDAP server initialization of an LDBM or GDBM (file-based) backend, the backend does not start. If the srvStartUpError option in the LDAP server configuration file is set to ignore, the LDAP server continues to run with those backends that successfully start. If the srvStartUpError option is set to terminate (this is the default if the configuration option is not specified), the program ends. If a read or write file lock error is encountered while executing the ds2ldif utility to unload an LDBM backend, an unloadRequest extended operation is attempted to unload the desired backend data.

Operator response:   None.

User response:   None.

System programmer response:   None.

Administrator response:   None.

Problem determination:   Not applicable.

Source:   LDAP

Module:   None.

Routing code:   None.

Descriptor code:   None.

Automation:   Not applicable.

GLD1210E  Terminating LDAP server because tcpTerminate option is set to 'terminate'.

Explanation:   The LDAP server has detected that there are no active network interfaces or has found an error while initializing SSL or Kerberos. The LDAP server is ending because the tcpTerminate option in the LDAP server configuration file is set to terminate.

Example:   None.

System action:   The program ends.

Operator response:   None.

User response:   None.

System programmer response:   None.

Administrator response:   None.

Problem determination:   Not applicable.

Source:   LDAP

Module:   None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1211I Listening for requests on ip secure port port.

Explanation: The LDAP server is listening for secure requests on the indicated network interface.

In the message text:

ip  IP address
port  Port number

Example: None.

System action: The program continues.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1212E Unable to chmod file filename: error_code/reason_code - error_text

Explanation: The LDAP server or utility is unable to change the permission bits on the indicated file. The failing routine can be fchmod() or chmod(). Refer to the description of these routines in z/OS XL C/C++ Run-Time Library Reference for more information on the error.

In the message text:

filename  LDBM database, schema filename, or replication error log
error_code  Error code from routine
reason_code  Reason code from routine
error_text  Error text corresponding to the error code

Example: None.

System action:
• If the error occurs during backend initialization, the backend does not start. If the srvStartUpError option in the LDAP server configuration file is set to ignore, the LDAP server continues to run with those backends that successfully start. If the srvStartUpError option is set to terminate (this is the default if the configuration option is not specified), the program ends.
• If the error occurs while executing the ds2dif utility to unload an LDBM backend, an unloadRequest extended operation is attempted to unload the desired backend data.
• If the error occurs during an LDBM or GDBM operation, then, if the fileTerminate option in the LDAP server configuration file is set to recover (this is the default if the configuration option is not specified), the server continues to run but the LDBM or GDBM backend is placed in read-only state. If the fileTerminate option is set to terminate, the program ends.
• For a replication error log file, the permission bits are changed when the file is first created. If an error occurs at this point, the replication error information is written to the error log file the next time a replication error occurs.
• If the error occurs while attempting to modify the global schema, the schema modification fails but the updates to the schema are saved in the schema.db.new file.

Operator response: None.
User response: None.
System programmer response: None.

Administrator response: Use the information in the message to correct the error. Ensure that the LDAP server has the appropriate authority to change permission bits on all of the LDBM, GDBM (file-based), schema, and replication error log files. Also ensure that the user ID that is running the ds2dif utility is a superuser or is the owner of the LDBM database files or in the same group as the LDAP server user ID. See the documentation on setting up the user ID that runs the LDAP server for more information on giving the LDAP server the authority to perform permission bit updates on the files. Restart the program if it ended or if the backend did not start and is needed. If an LDBM or GDBM backend was placed in read-only state, it can be reset to read-write state by restarting the LDAP server or by using the LDAP server BACKEND operator modify command. For a schema update, re-issue the schema modify request.

Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1213E Unable to chown file filename:
error_code/reason_code - error_text

Explanation: The LDAP server or utility is unable to change the owner or group of the indicated database file. The failing routine can be fchown() or chown(). Refer to the description of these routines in z/OS XL C/C++ Run-Time Library Reference for more information on the error.

In the message text:
filename
LDBM database or schema filename
error_code
Error code from routine
reason_code
Reason code routine
error_text
Error text corresponding to the error code

Example: None.

System action:
• If the error occurs during backend initialization, the backend does not start. If the srvStartUpError option in the LDAP server configuration file is set to ignore, the LDAP server continues to run with those backends that successfully start. If the srvStartUpError option is set to terminate (this is the default if the configuration option is not specified), the program ends.
• If the error occurs while executing the ds2ldif utility to unload an LDBM backend, an unloadRequest extended operation is attempted to unload the desired backend data.
• If the error occurs during an LDBM or GDBM operation, then, if the fileTerminate option in the LDAP server configuration file is set to recover (this is the default if the configuration option is not specified), the server continues to run but the LDBM or GDBM backend is placed in read-only state. If the fileTerminate option is set to terminate, the program ends.
• If the error occurs while attempting to modify the global schema, the schema modification fails but the updates to the schema are saved in the schema.db.new file.

System programmer response: None.

Administrator response: Use the information in the message to correct the error. Then restart the program.

Problem determination: Not applicable.

Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1214A Unable to create schema search entry:
return_code return_code - error_text

Explanation: The LDAP server is unable to create an internal search entry containing the contents of the LDAP server schema.

In the message text:
return_code
LDAP return code
error_text
Error text

Example: None.

System action: The program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the information in the message to correct the error. Then restart the program.

Problem determination: Not applicable.

Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1215A Unable to complete schema change due to failure in backend name: return code return_code - error_text

Explanation: During schema initialization or after

Example: None.
schema modification, each active backend is notified that the schema has changed so that the backend can do any needed processing to use the new schema. In particular, if a TDBM backend is running in multi-server mode and has a DB_VERSION less than 4, then it must update the schema entry within the TDBM database. An error occurred during backend processing for the new schema.

In the message text:

name  Backend name
return_code  LDAP return code
error_text  Error text

Example: None.

System action: The program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Use the information in the message to correct the error. Then restart the program.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1216E Unable to send request for backend name to sysplex group owner: return code return_code.

Explanation: An error occurred while sending an XCF request to the LDAP cross-system group owner in the sysplex. The return code has the following values:

1  An unavailable XCF service.
80  An XCF error.

In the message text:

name  Backend name
return_code  Return code

Example: None.

System action: If the error occurs during LDAP server initialization, the program ends. If the error occurs after initialization, the LDAP server continues if the LDAP server schema is still usable; otherwise the program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1217E Unable to receive schema.

Explanation: The LDAP server is unable to load schema from the LDAP cross-system group owner in the sysplex. The problem might be that the sysplex group owner stopped the schema load or that the LDAP server is unable to allocate sufficient storage.

Example: None.

System action: If the error occurs during LDAP server initialization, the program ends. If the error occurs after initialization, the LDAP server continues if the LDAP server schema is still usable; otherwise the program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Verify that there is enough storage available for use by the LDAP server. Then restart the program if it ended. If the problem persists, contact the service representative.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1218E operation operation failed for dn to host: port, rc=return_code setting change aside into filename.

Explanation: The LDAP server is unable to replicate a change to the indicated server. The change is removed...
from the replication queue and placed in the replication error log file.

In the message text:
operation
  Operation name
dn  Distinguished name of entry to replicate
host  Replica server host name
port  Replica server port number
return_code
  LDAP return code
filename
  Replication error log file name
Example:  None.
System action:  The LDAP server continues. Replication to the indicated replica server continues.
Operator response:  None.
User response:  None.
System programmer response:  None.
Administrator response:  Use the information in the replication error log to correct the replication errors and apply the changes to the replica. To enable future operations to be set aside, restart the LDAP server, delete and add the replication entry, or increase the value of the ibm-slapdReplMaxErrors attribute in the replica entry. Note that these actions do not delete anything from the replication error log; they just allow additional changes to be set aside into the error log.
Problem determination:  Not applicable.
Source:  LDAP
Module:  None.
Routing code:  None.
Descriptor code:  None.
Automation:  Not applicable.

GLD1219I  The amount of replication changes set aside has reached the maximum for error log filename

Explanation:  The number of changes set aside into the replication error log since the LDAP server was last started has reached the maximum allowed. The maximum number is set in the ibm-slapdReplMaxErrors attribute in the replica entry.
In the message text:
filename
  Replication error log file name
Example:  None.
System action:  The LDAP server continues. Replication to the indicated server continues, although it may stall. Future replication failures will not be set aside.
Operator response:  None.
User response:  None.
System programmer response:  None.
Administrator response:  Use the information in the replication error log to correct the error on the replica server, then wait for the replication operation to be retried. If necessary, resynchronize the replica server. See the documentation on replication for more information on how to recover from out-of-sync conditions.
Problem determination:  Not applicable.
Source:  LDAP
Module:  None.
Routing code:  None.
Descriptor code:  None.
Automation:  Not applicable.

GLD1220I  Replication to replica host:port has stalled.

Explanation:  A failed replication operation is preventing other replication operations from occurring.
In the message text:
host  Replica server host name
port  Replica server port number
Example:  None.
System action:  The LDAP server continues. The replication operation is retried.
Operator response:  None.
User response:  None.
System programmer response:  None.
Administrator response:  Use the information in the replication error log to correct the error on the replica server, then wait for the replication operation to be retried. If necessary, resynchronize the replica server. See the documentation on replication for more information on how to recover from out-of-sync conditions.
Problem determination:  Not applicable.
Source:  LDAP
Module:  None.
Routing code:  None.
Descriptor code:  None.
Automation:  Not applicable.
GLD1221E Unable to write to file filename:
   error_code/reason_code - error_text.

Explanation: The LDAP server is unable to write to the indicated file. Refer to the description of fprintf() in z/OS XL C/C++ Run-Time Library Reference for more information on the error. If the indicated file is a replication error log, the file name is specified by the ibm-slapdLog attribute in the replica entry.

In the message text:
   filename
      File name
   error_code
      Error code from fprintf()
   reason_code
      Reason code from fprintf()
   error_text
      Error text corresponding to the error code

Example: None.

System action: The LDAP server continues. If the error occurs while writing to the replication error log, some information about the replication error cannot be written to the log and the replication operation cannot be set aside. In this case, the replication operation is retried and replication stalls if it is not successful.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the information in the message to correct the error. Verify that the file can be used in the indicated way by the LDAP server.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1222E File operation routine failed to file filename:
   error_code/reason_code - error_text.

Explanation: The LDAP server is unable to perform the indicated operation to the indicated file. Refer to the description of the routine in z/OS XL C/C++ Run-Time Library Reference for more information on the error. If the indicated file is a replication error log, the file name is specified by the ibm-slapdLog attribute in the replica entry.

In the message text:
   routine
      File operation routine name
   filename
      File name
   error_code
      Error code from the C API
   reason_code
      Reason code from the C API
   error_text
      Error text corresponding to the error code

Example: None.

System action: The LDAP server continues. If the error occurs while writing to the replication error log, some information about the replication error cannot be written to the log and the replication operation cannot be set aside. In this case, the replication operation is retried and replication stalls if it is not successful.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the information in the message to correct the error. Verify that the file can be used in the indicated way by the LDAP server.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1223I End display output

Explanation: This message is displayed at the end of the output of the LDAP server DISPLAY AUDIT operator modify command.

Example: None.

System action: The LDAP server continues.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.
GLD1224I Backend status
Explanation: This message is displayed at the beginning of the output of the LDAP server DISPLAY BACKEND operator modify command. The status of each backend follows.
Example: None.
System action: The LDAP server continues.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1225I Maintenance Mode status
Explanation: This message is displayed at the beginning of the output of the LDAP server DISPLAY MAINTMODE operator modify command. The maintenance mode status follows.
Example: None.
System action: The LDAP server continues.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1226I Debug settings
Explanation: This message is displayed at the beginning of the output of the LDAP server DISPLAY DEBUG operator modify command. The debug settings follow.
Example: None.
System action: The LDAP server continues.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1228E Unable to cancel thread: error_code/reason_code - error_text
Explanation: The LDAP server is unable to cancel a thread. Refer to the description of pthread_cancel() in z/OS XL C/C++ Run-Time Library Reference for more information on the error.
In the message text:

- error_code
  - Error code from pthread_cancel()
- reason_code
  - Reason code from pthread_cancel()
error_text

Error text corresponding to the error code

Example: None.

System action: The program continues. The request fails.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the information in the message to correct the error. Then retry the request. If the problem persists, contact the service representative.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1229E Environment variable file cannot be opened. Processing is continuing without setting additional environment variables.

Explanation: The LDAP server or utility is unable to open the environment variables file. The name of the file is specified in the current value of the LDAP_DS_ENVVARS_FILE environment variable. If this environment variable is not defined, the file name is specified by the //DD:ENVVAR statement in the procedure used to start the program. If the file in the DD statement cannot be opened, the program attempts to open /etc/ldap/ds.envvars.

Example: None.

System action: The program continues without setting additional environment variables.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Verify that an environment variables file exists and can be read. Restart the program if the environment variable must be set.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1230E Environment variable ignored because line is too long: line line_number, file filename.

Explanation: The LDAP server or utility encountered a line that is too long in its environment variables file. The total length of a line (including any continuation lines) must be less than 1024 characters.

In the message text:

\[\text{line_number}\]

Line number

\[\text{filename}\]

Environment variables file name

Example: None.

System action: The program continues without setting the environment variable on the indicated line.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Verify the contents of the environment variables file. Restart the program if the environment variable must be set.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1231E Environment variable ignored because '=' is missing: line line_number, file filename.

Explanation: The LDAP server or utility encountered an incorrect line in its environment variables file. An environment variable line consists of name=value but the indicated line does not contain an = sign.

In the message text:

\[\text{line_number}\]

Line number

\[\text{filename}\]

Environment variables file name

Example: None.

System action: The program continues without setting the environment variable on the indicated line.

Operator response: None.

User response: None.

System programmer response: None.
Administrator response: Verify the contents of the environment variables file. Restart the program if the environment variable must be set.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1232E Environment variable is ignored because name is NULL: line line_number, file filename.

Explanation: The LDAP server or utility encountered an incorrect line in its environment variables file. An environment variable line consists of name=value but the indicated line does not contain a name.

In the message text:

- line_number
  Line number

- filename
  Environment variables file name

Example: None.

System action: The program continues without setting the environment variable on the indicated line.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Verify the contents of the environment variables file. Restart the program if the environment variable must be set.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1234E Last line in environment variables file ignored because it has a continuation character: line line_number, file filename.

Explanation: The LDAP server or utility found a continuation character at the end of the last line in its environment variables file. The line is ignored because there is no continuation line.

In the message text:

- line_number
  Line number

- filename
  Environment variables file name

Example: None.

System action: The program continues without setting the environment variable on the indicated line.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.
GLD1237E Incomplete shutdown processing; some data may not be committed and resources not freed.

Explanation: During LDAP server shutdown processing, the server waits 60 seconds for all requests in process to complete. If the requests complete, then each backend is notified to free any resources and commit any data. The requests did not complete, thus the backends are not notified.

Example: None.

System action: The backends do not commit data and do not free resources.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Restart the LDAP server and re-issue any incomplete requests.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1238E Unable to decode a request for type backend named name because attribute type attribute is not defined.

Explanation: The LDAP server found an attribute type used in a request for the indicated backend but the attribute type is not defined in the LDAP server schema.

In the message text:

\[
\begin{align*}
type & \quad \text{Backend type} \\
name & \quad \text{Backend name} \\
attribute & \quad \text{Undefined attribute type}
\end{align*}
\]

Example: None.

System action: The request fails. The program may end depending on when the error occurs. A follow-on message indicates the effect on the program.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the information in the follow-on message to resolve the problem. If the LDAP server is in a sysplex and is not the owner of the backend in the sysplex, then restart the server.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1240E A type backend named name database update is missing.

Explanation: The LDAP server has received a database update for the indicated backend from the cross-system group owner but the update cannot be processed because the previous update has not been received.

In the message text:

\[
\begin{align*}
type & \quad \text{Backend type} \\
name & \quad \text{Backend name}
\end{align*}
\]

Example: None.

System action: The LDAP server ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the information in the follow-on message to resolve the problem. If the LDAP server is in a sysplex and is not the owner of the backend in the sysplex, then restart the server.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1241W The option1 configuration option value value cannot be used because it is not in a configured suffix and option2 is not specified.

Explanation: The distinguished name specified for the LDAP administrator (adminDN), peer server (peerServerDN), or master replica server (masterServerDN) in the LDAP server configuration file
cannot be used to bind to the LDAP server because the password is not specified, either in the corresponding password configuration option (adminPW, peerServerPW, or masterServerPW) or in an entry for the distinguished name in the directory. The DN does not fall under any of the suffixes in the LDAP server configuration file or any of the suffixes added by plug-in extensions to the LDAP server, thus there cannot be an entry for the DN in the directory.

In the message text:

option1
   LDAP server configuration option
value
   LDAP server configuration option value
option2
   LDAP server configuration option

Example: None.

System action: The program continues but the LDAP administrator, peerServer DN, or masterServer DN cannot access the LDAP server.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: If intending to bind to the LDAP server using the administrator, peer server, or master server distinguished name, it is recommended that the configuration option value be changed so that the DN falls under one of the suffixes in the LDAP server configuration file or one of the suffixes added by plug-in extensions to the LDAP server. Then restart the server and add an entry for the distinguished name containing a userPassword value to the directory. As a less-secure alternative, the corresponding password configuration option can be added to LDAP server configuration file and then restart the LDAP server.

Problem determination: Not applicable.

Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1242W SDBM update operations and some search operations cannot be performed because there is no RACF address space.

Explanation: The SDBM backend has detected that the RACF address space is not running. SDBM operations that use the RACF address space fail and return a return code of decimal 52 (LDAP_UNAVAILABLE - 'Directory server function is unavailable'). These operations include all update operations and any search operation that scans the RACF database using the RACF SEARCH command. Binding to SDBM and searches for a specific RACF user, group, or connection can be performed because they do not use the RACF address space.

Example: None.

System action: The program continues but SDBM update operations and some search operations cannot be performed.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: If SDBM is needed to update RACF user, group, or connection profiles or to search the RACF database, start the RACF address space. These SDBM operations can then be performed.

Problem determination: Not applicable.

Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1243I Analysis of DB2 RUNSTATS utility output complete.

Explanation: The output of the DB2 RUNSTATS utility in the DB2 catalog has been reexamined. This message is displayed upon completion of the LDAP server REFRESH DB2RUNSTATS operator modify command.

Example: None.

System action: The LDAP server continues.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.
GLD1244E Unable to analyze DB2 RUNSTATS utility output.

Explanation: The LDAP server cannot analyze DB2 RUNSTATS utility output in the DB2 catalog.

Example: None.

System action: If the error occurs during backend initialization, the tcpTerminate option in the LDAP server configuration file determines what the server does. If the tcpTerminate option is set to recover (this is the default if the configuration option is not specified), LDAP server initialization continues. In this case, DB2 catalog statistics are not available until the error is corrected and the server is restarted or the REFRESH DB2RUNSTATS operator modify command is run. If the tcpTerminate option is set to terminate, the program ends. If the error occurs while processing the LDAP server REFRESH DB2RUNSTATS operator modify command, the program continues, using the existing DB2 catalog statistics.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the information in the message to correct the error. Then retry the request. If the problem persists, contact the service representative.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1245E Unable to set thread-specific value: error_code/reason_code - error_text

Explanation: The LDAP server is unable to store thread specific information for a thread. Refer to the description of pthread_setspecific in z/OS XL C/C++ Run-Time Library Reference for more information on the error.

In the message text:

error_code
  Error code from pthread_setspecific()
reason_code
  Reason code from pthread_setspecific()
error_text
  Error text corresponding to the error code

Example: None.

System action: The program continues. The request fails.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the information in the message to correct the error. Then retry the request. If the problem persists, contact the service representative.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.
Explanation: The LDAP server is unable to store thread specific information for a thread. Refer to the description of `pthread_key_create()` in [z/OS XL C/C++ Run-Time Library Reference](https://publib.boulder.ibm.com/infocenter/collcenter/ptref/ccb4e3a7-1488-4495-8878-5b6cd11f9d39) for more information on the error.

In the message text:
- `error_code`: Error code from `pthread_key_create()`
- `reason_code`: Reason code from `pthread_key_create()`
- `error_text`: Error text corresponding to the error code

Example: None.

System action: The program will end.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the information in the earlier message to correct the error. Then restart the LDAP server and retry the request. If the problem persists, contact the service representative.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.
GLD1250E The option configuration option requires the type backend.

**Explanation:** The LDAP server or utility found an option in the LDAP server configuration file whose processing requires that a specific type of backend also be configured, but a backend of that type is not contained in the configuration file.

In the message text:

- **option**
  - LDAP server configuration option
- **type**
  - Backend type

**Example:** None.

**System action:** The program ends.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Either remove the option from the LDAP server configuration file or add a backend of the specified type to the configuration file. Then restart the program.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

GLD1251E Schema database version version1 not supported, highest supported version2.

**Explanation:** The LDAP server or utility found that the schema database is at a higher version level than is currently supported by the LDAP server. This indicates that the schema database may contain values that the LDAP server cannot process. The schema cannot be loaded from the database.

In the message text:

- **version1**
  - Database schema version
- **version2**
  - Server schema version

**Example:** None.

**System action:** The program ends.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** If a command utility such as `ldapsearch` is being used and secure communications are intended, ensure that the `-Z` (use secure communications) option is specified. If secure communications are not wanted, then make sure that a non-secure port is specified on the command utility. If another LDAP client application is being used and secure communications are intended, verify that the application calls `ldap_ssl_init()` and `ldap_ssl_client_init()`. If secure communications are not wanted, make sure that the application uses a non-secure port.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.
GLD1253W  The 'old_option' configuration option value has been replaced by 'new_option'.

Explanation: The LDAP server or utility found an option or value in the LDAP server configuration file that is no longer supported. That option or value has been replaced by another option or value. To facilitate migration, the program still accepts the old option or value, but internally converts it to the new option or value, as displayed in the message. The LDAP server configuration file is not changed.

In the message text:

old_option
Old LDAP server configuration option

new_option
New LDAP server configuration option

Example: None.

System action: The program continues, using the replacement configuration option or value. Support for the old configuration option or value may be removed in a future release of the LDAP server, resulting in the program ending.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the information in the earlier message to correct the error. Restart the program if it didn't initialize or if the plug-in is needed.

Problem determination: Not applicable.

Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1255E  Plug-in start-up failed for plug-in named 'name'.

Explanation: The indicated plug-in failed to start. A previous message indicates the reason for the failure.

In the message text:

name
Plug-in name

Example: None.

System action: The plug-in does not start. If the srvStartUpError option in the LDAP server configuration file is set to ignore, the LDAP server continues to run with those backends and plug-ins that successfully initialize. If the srvStartUpError option is set to terminate (this is the default if the configuration option is not specified), the program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the information in the earlier message to correct the error. Restart the program if it didn't start or if the plug-in is needed.

Problem determination: Not applicable.

Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.
GLD1256E Replication entry 'name' is not supported in this replication configuration.

Explanation: The LDAP server or utility found a replication entry that is not supported in this replication configuration. If useAdvancedReplication on is specified in the LDAP server configuration file, then entries with an object class of replicaObject are not supported. If useAdvancedReplication off is specified in the LDAP server configuration file, then entries with object classes of ibm-replicationAgreement, ibm-replicationContext, ibm-replicationGroup, and ibm-replicationSubEntry are not supported. If running the ldif2ds utility, a previous message indicates the name of the LDIF file containing the entry.

In the message text:

name
Entry distinguished name

Example: None.

System action:
- If the error occurs while running an LDAP utility, the program ends.
- If the error occurs during initialization of an LDAP server backend, then the backend does not start. If the srvStartUpError option in the LDAP server configuration file is set to ignore, the LDAP server continues to run with those backends that successfully start. If the srvStartUpError option is set to terminate (this is the default if the configuration option is not specified), the program ends.

Operator response: None.
User response: None.
System programmer response: None.
Administrator response:
- If useAdvancedReplication on is specified in the LDAP server configuration file, the LDAP server or utility does not support basic replication entries with an object class of replicaObject. If configuring a basic replication environment, change the useAdvancedReplication option setting from on to off to allow basic replication entries to be used. If configuring an advanced replication environment, then all entries with object classes of ibm-replicationAgreement, ibm-replicationContext, ibm-replicationGroup, and ibm-replicationSubEntry must be removed from the LDAP server backend or the input LDIF file.
- If useAdvancedReplication off is specified in the LDAP server configuration file, the LDAP server or utility does not support advanced replication entries with object classes of ibm-replicationAgreement, ibm-replicationContext, ibm-replicationGroup, and ibm-replicationSubEntry. If configuring an advanced replication environment, change the

useAdvancedReplication option setting from off to on to allow advanced replication entries to be used. If configuring a basic replication environment, then all entries with object classes of ibm-replicationAgreement, ibm-replicationContext, ibm-replicationGroup, and ibm-replicationSubEntry must be removed from the LDAP server backend or the input LDIF file.

Then restart the program.

Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1257E Multi-server configuration backend (CDBM) support is required.

Explanation: The multiserver option in the CDBM section of the LDAP server configuration file must be set on because there is another backend section that has multiserver set on. When CDBM is configured, all CDBM, TDBM, LDBM, and GDBM backends must have the same setting for the multiserver option.

Example: None.

System action: The program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Either set the multiserver option in the CDBM section of the LDAP server configuration file on or set all the multiserver options off. Then restart the program.

Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1258E The operations monitor ID 'opid' is already associated with WLM transaction name 'name'.

Explanation: The operations monitor ID (OPID) specified on the WLMEXCEPT operator modify command is already associated with a WLM transaction name. An OPID is only allowed to be associated with one WLM transaction name at a time.
In the message text:

**opid**
- operations monitor identifier

**name**
- WLM transaction name

**Example:** None.

**System action:** The LDAP server continues. The request fails.

**Operator response:** Contact the LDAP Administrator or see the Administrator response.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Verify the OPID specified on the WLMEXCEPT operator modify command is correct and is not already associated with a WLM transaction name. If the correct OPID was specified, issue the RESET WLMEXCEPT operator modify command to remove the current OPID mapping. Then re-issue the same WLMEXCEPT operator modify command to set the OPID mapping to the WLM transaction name specified in the message.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

GLD1259I The operations monitor ID ‘opid’ is now associated with WLM transaction name ‘name’.

**Explanation:** The operations monitor ID (OPID) is successfully associated with the WLM transaction name specified. Future client requests in the LDAP server that match the search pattern identified by the OPID are routed to the specified WLM transaction name.

In the message text:

**opid**
- operations monitor identifier

**name**
- WLM transaction name

**Example:** None.

**System action:** The LDAP server continues.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** None.

---

GLD1260I Now setting the LDAP server health value to ‘value’ percent.

**Explanation:** The LDAP server has adjusted its internal health value to accurately reflect the number of errors that have occurred in the LDAP server. The health value is the number of failures that have occurred in the last 5000 client operations. The health value is only updated if one minute has passed since the internal health value was last set. The internal health value is used by the sysplex distributor to help distribute incoming client requests to the LDAP servers within the sysplex.

In the message text:

**value**
- LDAP health percentage

**Example:** None.

**System action:** The LDAP server continues.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** If activity or audit logging is active, analyze the activity or audit logs to determine the client application errors. If activity or audit logging is not active, consider turning on ERROR tracing on the LDAP server to determine the client application errors. Correct any client applications that are resulting in errors in the LDAP server. After the client application errors are fixed, continue to monitor the LDAP server’s health value.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

GLD1261E The operations monitor ID ‘opid’ is not valid.

**Explanation:** The operations monitor ID (OPID) value entered on the WLMEXCEPT operator modify command is not valid.

In the message text:

**opid**
- operations monitor identifier

**name**
- WLM transaction name

**Example:** None.

**System action:** The LDAP server continues.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** None.
**opid**

operations monitor identifier

**Example:** None.

**System action:** The LDAP server continues. The request fails.

**Operator response:** Issue a WLMEXCEPT operator modify command with a valid OPID value. Contact the LDAP Administrator to determine a valid OPID value.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Verify the OPID specified on the WLMEXCEPT operator modify command is a valid number and exists as an id value in a search pattern returned on the cn=operations,cn=monitor entry.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD1262E** The WLM transaction name 'name' is not valid.

**Explanation:** The WLM transaction name entered on the WLMEXCEPT operator modify command is not valid.

In the message text:

name

WLM transaction name

**Example:** None.

**System action:** The LDAP server continues. The request fails.

**Operator response:** Issue a WLMEXCEPT operator modify command with a valid WLM transaction name. Contact the LDAP Administrator to determine a valid WLM transaction name.

**User response:** None.

**System programmer response:** None.

**Administrator response:** None.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD1263I** wlmExcept reset completed.

**Explanation:** If an operations monitor ID (OPID) has been specified on the RESET WLMEXCEPT operator modify command, then that OPID is no longer associated with a WLM transaction name. If an OPID has not been specified on the RESET WLMEXCEPT operator modify command, then the LDAP server defaults to using only the configured wlmexcept options for routing incoming client requests to WLM transaction names.

**Example:** None.

**System action:** The LDAP server continues.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** None.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD1264E** The Extended Operation owner cannot be contacted.

**Explanation:** The LDAP server is unable to contact the Extended Operation owner to request a function be performed.

**Example:** None.

**System action:** An error is returned to the client requesting the operation, the LDAP server continues.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Determine the owning system by issuing the LDAP server DISPLAY XCF operator modify command for the LDAP server reporting the error. The command output indicates which LDAP server is the group owner. Then issue the DISPLAY XCF operator modify command for the owning LDAP server and verify that this server is really the group owner. Restart the owning LDAP server if there is no response to the DISPLAY XCF operator modify command. Restart this LDAP server if it ended. Retry the Extended Operation request if it failed. If the problem persists, contact the service representative.

**Source:** LDAP

**Module:** None.

**Routing code:** None.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1265E  The Extended Operation owner is busy, retrying.
Explanation: The LDAP server which is the owner of Extended Operations in the LDAP cross-system group in the sysplex is currently busy and cannot respond to the Extended Operation request.
Example: None.
System action: The LDAP server continues and retries the request.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.

GLD1266E  The serverCompatLevel is set to value. The configuration option 'option' requires minimum serverCompatLevel min_value.
Explanation: The serverCompatLevel value specified in the configuration file for this server is set to an unsupported level for the configuration option indicated in the message.
In the message text:
value
    serverCompatLevel option value
option
    LDAP server configuration option
min_value
    serverCompatLevel option minimum value
Example: None.
System action: The LDAP server ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Set the serverCompatLevel configuration option to the minimum level specified in the message or remove the configuration option indicated in the message.

GLD1267E  The LDAP server must have READ access to the BPX.WLMSERVER profile.
Explanation: The user ID that runs the LDAP server must have READ access to the BPX.WLMSERVER profile in the RACF FACILITY class so that the LDAP server can connect to Workload Manager (WLM).
Example: None.
System action: If the srvStartUpError option in the LDAP server configuration file is set to ignore, the LDAP server continues to run with no WLM support. If the srvStartUpError option is set to terminate (this is the default if the configuration option is not specified), the LDAP server ends.
Operator response: None.
User response: None.
System programmer response: None.
GLD1268E The serverCompatLevel is set to value.  
The sysplex owner has a different serverCompatLevel setting.

Explanation:  The serverCompatLevel value specified in the configuration file for this server is incompatible with the serverCompatLevel value established by the sysplex owner.

In the message text:

value
serverCompatLevel option value

Example:  None.

System action:  The LDAP server ends.

Operator response:  None.

User response:  None.

System programmer response:  None.

Administrator response:  Obtain the serverCompatLevel value established by the sysplex owner. Either migrate the server to an appropriate level for the sysplex or upgrade this LDAP server to a level that is compatible with the sysplex.

Problem determination:  Not applicable.

Source:  LDAP

Module:  None.

Routing code:  None.

Descriptor code:  None.

Automation:  Not applicable.

GLD1269A Unable to initialize WLM support.

Explanation:  The LDAP server is unable to initialize WLM support.

Example:  None.

System action:  If the srvStartUpError option in the LDAP server configuration file is set to ignore, the LDAP server continues to run with no WLM support. If the srvStartUpError option is set to terminate (this is the default if the configuration option is not specified), the LDAP server ends.

Operator response:  None.

User response:  None.

System programmer response:  None.

Administrator response:  If WLM support is needed, restart the LDAP server with ERROR tracing turned on to determine the failure. The WLM routine that has failed is present in the LDAP ERROR traces. Refer to the z/OS XL C/C++ Run-Time Library Reference for more information on the error. If the problem persists, contact the service representative.

Problem determination:  Not applicable.

Source:  LDAP

Module:  None.

Routing code:  None.

Descriptor code:  None.

Automation:  Not applicable.

GLD1270E IP address 'value' for configuration option 'option' is not valid.

Explanation:  The LDAP server or utility found an option in the LDAP server configuration file that has a value that is not supported for that option.

In the message text:

value
LDAP server configuration option IP address
option
LDAP server configuration option

Example:  None.

System action:  The program ends.

Operator response:  None.

User response:  None.

System programmer response:  None.

Administrator response:  Correct the LDAP server configuration file. If the option value looks correct, check that the option on the next line after this option line starts in column 1. A blank in column 1 of the next line indicates that it is a continuation line. The next line is then appended to the preceding option line and thus can result in a value that is not supported for the option. Then restart the program.

Problem determination:  Not applicable.

Source:  LDAP

Module:  None.

Routing code:  None.

Descriptor code:  None.

Automation:  Not applicable.

GLD1271E The operations monitor ID parameter is missing.

Explanation:  The operations monitor id (OPID) parameter is missing on the WLMEXCEPT operator modify command.

Example:  None.

System action:  The LDAP server continues. The request fails.

Operator response:  Issue a WLMEXCEPT operator modify command with an OPID value. Contact the
LDAP Administrator to determine the correct OPID value.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Verify that an OPID is specified on the `WLMEXCEPT` operator modify command.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD1272E** The WLM transaction name parameter is missing.

**Explanation:** The WLM transaction name parameter is missing on the `WLMEXCEPT` operator modify command.

**Example:** None.

**System action:** The LDAP server continues. The request fails.

**Operator response:** Issue a `WLMEXCEPT` operator modify command with a WLM transaction name. Contact the LDAP Administrator to determine the correct WLM transaction name.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Verify that a WLM transaction name is specified on the `WLMEXCEPT` operator modify command.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD1274W** Replication conflict: a conflict is detected on a modify operation of `dn` in type backend named `name`. A refresh of the entry will be requested because the entry has been modified on this server before it was modified on the supplier.

**Explanation:** The LDAP server detected a replication conflict on a modify operation of an entry in the specified backend. The conflict occurred because an incoming modification has a newer timestamp than the timestamp on the target entry. This server will request the supplier to send a refreshed entry in order to resolve this conflict.

In the message text:

- `dn` Entry distinguished name
- `type` Backend type
- `name` Backend name

**Example:** None.

**System action:** The LDAP server converts the add operation into a modify operation, and continues to process the request.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Inspect the recorded entry in the lost and found log file and verify that the attributes and their values are modified correctly.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.
name
 Backend name

Example: None.

System action: The LDAP server sends a entry refresh request to the supplier.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1276W Replication conflict: a conflict is detected on a modify operation of 'dn' in type backend named name. The modification was rejected because it had a timestamp older than or equal to the target entry timestamp.

Explanation: The LDAP server detected a replication conflict on a modify operation of an entry in the specified backend. The conflict occurred because an incoming modification has the same or older timestamp than the timestamp on the target entry. This server will ignore the modify request.

In the message text:

name Backend name

type Backend type

dn Entry distinguished name

Example: None.

System action: The LDAP server ignores the incoming request.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Check that no updates have been lost by comparing the contents of this entry among all the servers in the replication topology.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.
name1
   Entry distinguished name of conflicting entry

name2
   Entry distinguished name of conflicting entry

value
   Master server distinguished name

Example:  None.

System action:  The LDAP server continues, but the consumer server credential entry settings are ignored.

Operator response:  None.

User response:  None.

System programmer response:  None.

Administrator response:  Update the ibm-masterServerPW attribute values on the conflicting consumer server entries to use the same value or use a different ibm-masterServerDN attribute value for one of the conflicting consumer server credential entries. A consumer server credential entry has an object class value of ibm-slapdSupplier or ibm-slapdReplication.

Problem determination:  Not applicable.

Source:  LDAP

Module:  None.

Routing code:  None.

Descriptor code:  None.

Automation:  Not applicable.

GLD1278E  Routine 'routine' failed: Return code

   return_code - error_text.

Explanation:  An internal programming error has been detected by the routine identified in the message.

In the message text:

   routine
      Routine name

   return_code
      Return code from routine

   error_text
      Error text corresponding to the return code

Example:  None.

System action:  The current ldap operation being handled by the server ends.

Operator response:  None.

User response:  None.

System programmer response:  None.

Administrator response:  Use the information in the earlier message to correct the error, then restart the program.

Problem determination:  Not applicable.

Source:  LDAP

Module:  None.

Routing code:  None.

Descriptor code:  None.

Automation:  Not applicable.

GLD1800E  Command options option1 and option2 are mutually exclusive.

Explanation:  The two command-line parameters indicated in the message cannot both be specified at the same time.

In the message text:

   option1
      Command-line parameter name

   option2
      Command-line parameter name
Example: None.
System action: The program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1801E Option option is specified more than once with different values.
Explanation: The command-line parameter indicated in the message can only have one value.
In the message text:

option
Command-line parameter name

Example: None.
System action: The program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Remove one or both of the parameters from the command line. Then restart the program.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1802E The summary message frequency must be a non-negative decimal integer.
Explanation: The frequency value specified for the -q command-line parameter is not a positive integer.
In the message text:

Example: None.
System action: The program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1803E At least one phase option (-c, -p, -l) must be specified.
Explanation: At least one phase command-line parameter must be specified for the ldif2ds utility.
Example: None.
System action: The program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Specify one or more phase parameters on the command line. Then restart the program.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1804E Unable to open name:
error_code/reason_code - error_text
Explanation: The file indicated in the message cannot be opened in the required way: read for an input file, write for an output file. The file can be a filesystem file or a dataset. If the file name is //DD:INTRDR, the failure occurred while opening the internal reader to submit the load jobs. Refer to the description of fopen() in [Z/OS XL C/C++ Run-Time Library Reference](https://www.ibm.com/support/docview.wss?uid=swg21285712) for more information on the error.
In the message text:

name
Dataset name or file name
error_code
Error code from fopen()
**GLD1805E** Unable to read name:

error_code/reason_code - error_text

**Explanation:** The file indicated in the message cannot be read. The file can be a filesystem file or a dataset. Refer to the description of fgets() in [z/OS XL C/C++ Run-Time Library Reference](https://www-01.ibm.com/support/knowledgecenter/SSEKJ0_1.11.0/com.ibm.zos.v1r11Samples.LDIF2DS/gl친1805E.html) for more information on the error.

In the message text:

name

Dataset name or file name

error_code

Error code fgets()

reason_code

Reason code from the routine

error_text

Error text corresponding to the error code

**Example:** None.

**System action:** The program ends.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Use the information in the message to correct the error. Then restart the program.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD1806E** Unable to write name using routine:

error_code/reason_code - error_text

**Explanation:** An attempt to write to the file indicated in the message failed. The file can be a filesystem file or a dataset. If the file name is //DD:INTRDR, the failure occurred while submitting a load job to the internal reader. The routine used to perform the write is also indicated in the message. Refer to the description of the routine in [z/OS XL C/C++ Run-Time Library Reference](https://www-01.ibm.com/support/knowledgecenter/SSEKJ0_1.11.0/com.ibm.zos.v1r11Samples.LDIF2DS/gl친1806E.html) for more information on the error.

In the message text:

name

Dataset name or file name

routine

Routine that failed

error_code

Error code from the routine

reason_code

Reason code from the routine

error_text

Error text corresponding to the error code

**Example:** None.

**System action:** The program ends.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Use the information in the message to correct the error. Then restart the program.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD1807E** At least one LDIF file must be specified.

**Explanation:** The -c (check) or -p (prepare) parameter is specified on the ldif2ds command line but no LDIF files are specified with either the -i or -e command-line parameter.

**Example:** None.

**System action:** The program ends.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Use the information in the message to correct the error. Then restart the program.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.
System action: The program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1808I Preparing LDIF file filename.
Explanation: The ldif2ds utility is starting the prepare phase for the directory entries in the indicated LDIF file.
In the message text:
filename
    LDIF file name
Example: None.
System action: The program continues.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1810I ldif2ds utility version version.release, Service level level.
Explanation: The ldif2ds utility with version, release, and service level indicated in the message is running.
In the message text:
version
    Utility version
release
    Utility release
Example: None.
System action: The program continues.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1811E ldif2ds utility terminating due to error condition.
Explanation: The ldif2ds utility is ending due to an error. Previous messages indicate the reason for the failure.
Example: None.
System action: The program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Use the information in the earlier messages to correct the problem. Then restart the program.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1812I name: number entries have been processed, error of them encountered.
Explanation: The utility has processed the indicated number of entries. This message is issued when all of the directory entries have been processed for the check or prepare phase. It is also issued as an intermediate status message as determined by the -q command-line parameter. Note that the program might have encountered errors during this processing. If so, processing of some entries may not have completed.
successfully. Additional messages are issued to indicate these errors.

In the message text:

name
Program name

number
Number of entries

error
Number of error entries

Example: None.

System action: The program continues.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1813E Incorrect continuation at line line_number of filename.

Explanation: The ldif2ds utility has encountered a continuation line at the start of a directory entry definition in the input LDIF file indicated in the message. A directory entry cannot begin with a continuation line.

In the message text:

line_number
Line number

filename
LDIF file name

Example: None.

System action: The utility skips the rest of this entry and continues the check phase with the next complete entry. No entries are prepared or loaded.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1814E Syntax error at line line_number of filename.

Explanation: The ldif2ds utility has encountered a syntax error in the LDIF statement beginning at the indicated line of the LDIF file.

In the message text:

line_number
Line number

filename
LDIF file name

Example: None.

System action: The utility skips the rest of this entry and continues the check phase with the next complete entry. No entries are prepared or loaded.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: If the statement is within an entry, verify that it has the correct format for specifying an entry record (either name: value or name:: value). If the statement is not within an entry and is not a comment, then it must be a version or dn statement. See IBM Tivoli Directory Server Client Programming for z/OS for more information on the format of LDIF statements. Correct the LDIF file. Then restart the program.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1815E Unable to decode binary value at line line_number of filename.

Explanation: The ldif2ds utility is unable to decode a base64-encoded value in the LDIF statement beginning at the indicated line of the LDIF file. The LDIF statement format is name:: value, indicating that the value must be base64-encoded.

In the message text:

line_number
Line number

filename
LDIF file name

Example: None.
GLD1816E LDIF version is not supported.

Explanation: The version directive in an LDIF file specifies a version number that is not supported by the ldif2ds utility. The utility only supports LDIF version 1. A previous message indicates the name of the LDIF file.

Example: None.

System action: The program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Either correct the changetype directive by specifying add or remove the directive from the LDIF file. Then restart the program.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1817E Unsupported 'changetype' directive at line line_number of filename.

Explanation: An entry in the LDIF file at the indicated line contains a changetype directive that does not specify an add operation. Only changetype: add is supported by the ldif2ds utility. Note that the changetype directive is not required.

In the message text:

Example: None.

System action: The utility skips the rest of this entry and continues the check phase with the next complete entry. No entries are prepared or loaded.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Specify a distinguished name for the entry in the LDIF file. Then restart the program.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1818E Zero-length distinguished name found at line line_number of filename.

Explanation: The ldif2ds utility found a zero-length distinguished name for a directory entry at the indicated line of the LDIF file. Every entry must have a distinguished name.

In the message text:

Example: None.

System action: The utility skips the rest of this entry and continues the check phase with the next complete entry. No entries are prepared or loaded.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1819E Unable to normalize value at line
  line_number of filename: error_text.
Explanation: The ldif2ds utility cannot convert the
  value on the indicated line to normalized format. The
  value is either a distinguished name or an attribute
  value.
In the message text:
  line_number
    Line number
  filename
    LDIF file name
  error_text
    Error message text
Example: None.
System action: The utility skips the rest of this entry
  and continues the check phase with the next complete
  entry. No entries are prepared or loaded.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Use the information in the
  message to correct the value in the LDIF file. Then
  restart the program.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1820E Unable to resolve attribute type
  attribute: error_text.
Explanation: The ldif2ds utility is unable to find the
  attribute indicated in the message in the LDAP server
  schema. Every attribute contained in the entry, including
  the attributes in the relative distinguished name (RDN)
  of the entry, must be already defined in the schema. A
  previous message indicates the name of the LDIF file
  containing the entry.
In the message text:
  attribute
    Attribute type
  error_text
    Error message text
Example: None.
System action: The utility skips the rest of this entry
  and continues the check phase with the next complete
  entry. No entries are prepared or loaded.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Either add the missing
  attribute to the LDAP server schema or remove the
  attribute from the LDIF file. Then restart the program.
  To add the attribute to the schema, start the LDAP server
  and issue an appropriate modify operation to the
  schema entry. Make sure to stop the LDAP server
  before using the ldif2ds utility to load entries into the
  server. See the documentation on schema for more
  information on modifying the LDAP server schema.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1821E Unable to resolve object class
  objectclass: error_text.
Explanation: The ldif2ds utility is unable to find the
  object class indicated in the message in the LDAP
  server schema. Every object class contained in the
  entry, including any object classes in the relative
  distinguished name (RDN) of the entry, must be already
  defined in the schema. A previous message indicates
  the name of the LDIF file containing the entry.
In the message text:
  objectclass
    Object class
  error_text
    Error message text
Example: None.
System action: The utility skips the rest of this entry
  and continues the check phase with the next complete
  entry. No entries are prepared or loaded.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Either add the missing
  object class to the LDAP server schema or remove the
  object class from the LDIF file. Then restart the
program. To add the object class to the schema, start
the LDAP server and issue an appropriate modify
operation to the schema entry. Make sure to stop the
LDAP server before using the ldif2ds utility to load
entries into the server. See the documentation on
schema for more information on modifying the LDAP
server schema.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD1822E**  No backend configured for DN 'name'.

**Explanation:** The ldif2ds utility encountered an entry
whose distinguished name (DN) does not belong to any
suffix in the backends contained in the LDAP server
configuration file. A previous message indicates the
name of the LDIF file containing the entry.

In the message text:

```plaintext
name
   Entry distinguished name
```

**Example:** None.

**System action:** The utility skips the rest of this entry
and continues the check phase with the next complete
entry. No entries are prepared or loaded.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Either add the appropriate
suffix option in the backend section of the LDAP server
configuration file or change the distinguished name of
the entry in the LDIF file. Then restart the program.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD1824E**  Entry 'name' is not in a TDBM backend.

**Explanation:** The ldif2ds utility encountered an entry
whose distinguished name (DN) belongs to a backend
in the LDAP server configuration file that is not a TDBM
backend. The ldif2ds utility can only load entries into a
single TDBM backend, thus all entries in the LDIF files
must belong to the same TDBM backend. A previous
message indicates the name of the LDIF file containing
the entry.

In the message text:

```plaintext
name
   Entry distinguished name
```

**Example:** None.

**System action:** The utility skips the rest of this entry
and continues the check phase with the next complete
entry. No entries are prepared or loaded.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Either change the
distinguished name of the entry so that it has one of the
suffixes of the TDBM backend being loaded or remove
the entry from the LDIF file. Then restart the program.

---

**GLD1823E**  Entry 'name' is not in the same
backend as previous entries.

**Explanation:** The ldif2ds utility encountered an entry
whose distinguished name (DN) belongs to a different
backend in the LDAP server configuration file than the
entries processed before this one. Each invocation of
the ldif2ds utility can only load entries into one
backend. All the entries in the LDIF files must belong to
the same backend, using any of the suffixes listed for
that backend in the LDAP server configuration file. A
previous message indicates the name of the LDIF file
containing the entry.

In the message text:

```plaintext
name
   Entry distinguished name
```

**Example:** None.

**System action:** The utility skips the rest of this entry
and continues the check phase with the next complete
entry. No entries are prepared or loaded.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Either change the
distinguished name of the entry so that it has one of the
suffixes of the TDBM backend being loaded or remove
the entry from the LDIF file. Then restart the program.

Other types of backends can only be loaded using an
add operation while the LDAP server is running.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1825I Using TDBM backend name.

Explanation: The ldif2ds utility is processing entries for the TDBM backend whose name is indicated in the message. This name is either the name specified in the database option for this TDBM backend in the LDAP server configuration file or, if no name is specified in the option, is a name generated by LDAP based on the position of the backend section in the LDAP server configuration file.

In the message text:

name
   Backend name

Example: None.

System action: The utility continues.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the information in the message to correct the error. Then restart the program.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1828E An internal error has occurred.

Explanation: An internal programming error has been detected by the utility.

Example: None.

System action: The program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Try running the utility again with -D ERROR specified on the command line. The debug output may assist in locating and correcting the error. If the problem persists, contact the service representative.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1826E Unable to encrypt attribute value for entry 'name': error_text.

Explanation: The ldif2ds utility encountered an error while trying to encrypt a value of an attribute that requires encryption: userPassword, secretKey, replicaCredentials, ibm-replicaKeyPw, or ibm-slapdMasterPw. The type of encryption in use is determined by the values of the pwEncryption and secretEncryption options within the TDBM backend section in the LDAP server configuration file. A previous message indicates the name of the LDIF file containing the entry.

In the message text:

attribute
   Attribute type

name
   Entry distinguished name

error_text
   Error message text

Example: None.

System action: The program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1829E Entry 'name' already exists.

Explanation: The ldif2ds utility encountered an entry that already exists, either as a prior entry in this LDIF file or in an LDIF file processed before this file, or as an existing entry in the TDBM backend being loaded. The
duplicate entry cannot be added to the directory. A previous message indicates the name of the LDIF file containing the entry.

In the message text:

*name*  
Entry distinguished name

**Example:** None.

**System action:** The utility skips the rest of this entry and continues the check phase with the next complete entry. No entries are prepared or loaded.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** If the entry is a duplicate of a previous entry in an LDIF file, remove one of these entries. If the entry is a duplicate of an entry in the TDBM directory, remove the entry from the LDIF file. Then restart the program.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

GLD1832I Checking LDIF file *filename*.

**Explanation:** The ldif2ds utility is starting the check phase for the directory entries in the indicated LDIF file. The check phase is performed when the `-c` or `-p` option is specified on the command line.

In the message text:

*filename*  
LDIF file name

**Example:** None.

**System action:** The utility continues.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** None.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

GLD1833E Unable to get directory entry 'name': *error_text*.

**Explanation:** The ldif2ds utility has detected an entry whose parent entry cannot be retrieved from the TDBM database. The distinguished name of the parent entry is indicated in the message. A previous message indicates the name of the LDIF file containing the child entry.

In the message text:

*name*  
Parent entry distinguished name

*error_text*  
Error message text

**Example:** None.

**System action:** The program ends.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** If appropriate, review the output of each load job to determine if it is successful. If a load job fails, use the information in the description of the ldif2ds utility to determine how to proceed.

**Note:** If a load job fails, do not run the ldif2ds utility again because this can add duplicate data to the database.

**Problem determination:** Not applicable.
Administrator response: Use the information in the message to correct the error. Then restart the program.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1834E The parent for entry 'name' does not exist.

Explanation: The ldif2ds utility encountered a non-suffix entry for which there is no parent. Every non-suffix entry must have a parent entry, either as a prior entry in this LDIF file or in an LDIF file processed before this file, or as an existing entry in the TDBM database being used. The entry cannot be added to the directory. A previous message indicates the name of the LDIF file containing the entry.

In the message text:

name
    Child entry distinguished name

Example: None.

System action: If the error occurs during the check phase, the utility skips the rest of this entry and continues the check phase with the next complete entry. No entries are prepared or loaded.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Remove the child entry from the LDIF file or change the distinguished name of the child entry to one for which the parent is not an alias or referral entry. Then restart the program.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1836E DN 'name' exceeds the maximum length of length.

Explanation: The ldif2ds utility encountered an entry for which the normalized distinguished name (DN) is longer than the maximum length allowed. The maximum length of a DN is determined by the size of the DN column in the DIR_ENTRY table, set when creating the TDBM database tables. The normalized DN is stored in this column. The normalized DN may not be same as the DN specified for the entry in the LDIF file. A previous message indicates the name of the LDIF file containing the entry.

In the message text:

name
    Entry normalized distinguished name

length
    Maximum DN length

Example: None.

System action: The utility skips the rest of this entry and continues the check phase with the next complete entry. No entries are prepared or loaded.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1835E Parent entry 'name' is a referral or an alias.

Explanation: The ldif2ds utility encountered an entry for which the parent entry is an alias entry or a referral entry. Alias and referral entries cannot have children.

The parent entry can be a prior entry in this LDIF file or in an LDIF file processed before this file or it can be an existing entry in the TDBM backend being loaded. A previous message indicates the name of the LDIF file containing the child entry.

In the message text:

name
    Parent entry distinguished name

Example: None.

System action: The utility skips the rest of this entry and continues the check phase with the next complete entry. No entries are prepared or loaded.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.
**System programmer response:** None.

**Administrator response:** Shorten the distinguished name of the entry. Then restart the program.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD1837E** No base structural object class specified for ’name’.

**Explanation:** The ldif2ds utility encountered an entry which does not contain a base structural object class. Every entry must have a single base structural object class, specified on the objectclass attribute within the entry. A previous message indicates the name of the LDIF file containing the entry.

In the message text:

```
name
Entry distinguished name
```

**Example:** None.

**System action:** The utility skips the rest of this entry and continues the check phase with the next complete entry. No entries are prepared or loaded.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Add a base structural object class to the entry. Then restart the program.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD1838E** Multiple base structural object classes specified for ’name’.

**Explanation:** The ldif2ds utility encountered an entry which contains more than one base structural object class. Every entry must have a single base structural object class, specified on the objectclass attribute within the entry. A previous message indicates the name of the LDIF file containing the entry.

In the message text:

```
name
Entry distinguished name
```

**Example:** None.

**System action:** The utility skips the rest of this entry and continues the check phase with the next complete entry. No entries are prepared or loaded.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Either remove the obsolete attribute type from the entry in the LDIF file or modify the LDAP server schema to remove the obsolete specification from the attribute definition. Then restart the program. To modify the schema, start the LDAP server and issue an appropriate modify operation to the schema entry. Make sure to stop the LDAP server before using the ldif2ds utility to load entries into the server. See the documentation for schema for more information.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.
information on modifying the LDAP server schema.

Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1840E Entry 'name' contains abstract class 'objectclass' as a base object class.

Explanation: The ldif2ds utility encountered an entry which contains an abstract object class as one of its base object classes. An abstract class cannot be a base object class; it must be derived from another object class specified in the entry. A previous message indicates the name of the LDIF file containing the entry.

In the message text:

name
Entry distinguished name
objectclass
Abstract object class name

Example: None.
System action: The utility skips the rest of this entry and continues the check phase with the next complete entry. No entries are prepared or loaded.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Remove the abstract object class from the entry. Verify that all the attributes used in the entry and in the relative distinguished name (RDN) of the entry are included in the remaining object classes. Then restart the program.

Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1841E Entry 'name' contains restricted attribute type 'attribute'.

Explanation: The ldif2ds utility encountered an entry which contains an attribute that cannot be set when adding an entry. The value for this attribute is instead generated by the LDAP server itself. With the exception of the ibm-EntryUUID, creatorsName, createTimestamp, modifiersName, and modifyTimestamp attribute types, an attribute type that is marked as NO-USER-MODIFICATION in its definition in the LDAP schema may not be used to create a new directory entry. A previous message indicates the name of the LDIF file containing the entry.

In the message text:

name
Entry distinguished name
attribute
Attribute type

Example: None.
System action: The utility skips the rest of this entry and continues the check phase with the next complete entry. No entries are prepared or loaded.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Remove the restricted attribute from the entry. Then restart the program.

Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1842E Entry 'name' contains obsolete object class 'objectclass'.

Explanation: The ldif2ds utility encountered an entry which contains an object class that is marked as obsolete in the LDAP server schema. Obsolete object classes cannot be used in an entry. A previous message indicates the name of the LDIF file containing the entry.

In the message text:

name
Entry distinguished name
objectclass
Obsolete object class name

Example: None.
System action: The utility skips the rest of this entry and continues the check phase with the next complete entry. No entries are prepared or loaded.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Either remove the obsolete
object class from the entry in the LDIF file or modify the LDAP server schema to remove the obsolete specification from the object class definition. Then restart the program. If the object class is removed, verify that all the attributes used in the entry and in the relative distinguished name (RDN) of the entry are included in the remaining object classes. If modifying the schema, start the LDAP server and issue an appropriate modify operation to the schema entry. Make sure to stop the LDAP server before using the ldif2ds utility to load entries into the server. See the documentation for schema for more information on modifying the LDAP server schema.

Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1843E The userPassword attribute is not allowed for entry 'name'.

Explanation: The ldif2ds utility encountered an entry which is set up for using native authentication but which also contains the userPassword attribute. This attribute cannot be included in an entry which is using native authentication. A previous message indicates the name of the LDIF file containing the entry.

In the message text:
name
  Entry distinguished name

Example: None.
System action: The utility skips the rest of this entry and continues the check phase with the next complete entry. No entries are prepared or loaded.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Change the entry so that it is an alias entry or a referral entry but not both. Then restart the program.

Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1844E Entry 'name' cannot be both an alias and a referral.

Explanation: The ldif2ds utility encountered an entry which is both an alias entry and a referral entry. This combination is not supported by the LDAP server. A previous message indicates the name of the LDIF file containing the entry.

In the message text:
name
  Entry distinguished name

Example: None.
System action: The utility skips the rest of this entry and continues the check phase with the next complete entry. No entries are prepared or loaded.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.

Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1845E The aclPropagate attribute for entry 'name' requires the aclEntry attribute.

Explanation: The ldif2ds utility encountered an entry which contains the aclPropagate attribute but does not contain the aclEntry attribute. aclEntry must be specified along with aclPropagate. A previous message indicates the name of the LDIF file containing the entry.

In the message text:
name
  Entry distinguished name

Example: None.
System action: The utility skips the rest of this entry and continues the check phase with the next complete entry. No entries are prepared or loaded.
Operator response: None.
User response: None.
**System programmer response:** None.

**Administrator response:** Either add the aclEntry attribute to the entry or remove the aclPropagate attribute. Then restart the program.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD1846E** The ownerPropagate attribute for entry 'name' requires the entryOwner attribute.

**Explanation:** The ldif2ds utility encountered an entry which contains the ownerPropagate attribute but does not contain the entryOwner attribute. entryOwner must be specified along with ownerPropagate. A previous message indicates the name of the LDIF file containing the entry.

In the message text:

```
name
   Entry distinguished name
```

**Example:** None.

**System action:** The utility skips the rest of this entry and continues the check phase with the next complete entry. No entries are prepared or loaded.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

---

**GLD1848E** The -o option must be specified if -p or -l is specified.

**Explanation:** The ldif2ds utility cannot be invoked without the -o command-line parameter when either the -p (prepare) or -l (load) command-line parameter is specified. -o is required during the prepare and load phases to identify the prefix part of the name of the output datasets.

**Example:** None.

**System action:** The program ends.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

---

**GLD1847E** Schema check failed for entry 'name': error_text.

**Explanation:** The ldif2ds utility encountered an entry whose attributes or object classes violate the LDAP server schema. The entry cannot be added to the directory. A previous message indicates the name of the LDIF file containing the entry.

In the message text:

```
name
   Entry distinguished name
erro_text
   Error message text
```

**Example:** None.

**System action:** The utility skips the rest of this entry and continues the check phase with the next complete entry. No entries are prepared or loaded.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

---

**GLD1848E** The -o option must be specified if -p or -l is specified.

**Explanation:** The ldif2ds utility cannot be invoked without the -o command-line parameter when either the -p (prepare) or -l (load) command-line parameter is specified. -o is required during the prepare and load phases to identify the prefix part of the name of the output datasets.

**Example:** None.

**System action:** The program ends.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

---

**GLD1847E** Schema check failed for entry 'name': error_text.

**Explanation:** The ldif2ds utility encountered an entry whose attributes or object classes violate the LDAP server schema. The entry cannot be added to the directory. A previous message indicates the name of the LDIF file containing the entry.

In the message text:

```
name
   Entry distinguished name
erro_text
   Error message text
```

**Example:** None.

**System action:** The utility skips the rest of this entry and continues the check phase with the next complete entry. No entries are prepared or loaded.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.
GLD1849E The output dataset name prefix is too long.

Explanation: The maximum length of the output dataset set name prefix is 22. The prefix is specified by the -o parameter on the ldif2ds utility command line.

Example: None.

System action: The program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Specify a value that is at most 22 characters long for the -o parameter on the command line. Then restart the program.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1850E Unable to assign database attribute identifier: error_text.

Explanation: The ldif2ds utility could not assign an internal identifier for an attribute type that is not currently known to the TDBM backend being loaded.

In the message text:

error_text

Example: None.

System action: The program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the information in the message to correct the problem. Then restart the program. If more information is needed, try running the utility again with -d ERROR specified on the command line. The debug output may assist in locating and correcting the error. If the problem persists, contact the service representative.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1851E Unable to assign database entry identifier: error_text.

Explanation: The ldif2ds utility could not assign an internal entry identifier for a new entry.

In the message text:

error_text

Example: None.

System action: The program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the information in the message to correct the problem. Then restart the program. If more information is needed, try running the utility again with -d ERROR specified on the command line. The debug output may assist in locating and correcting the error. If the problem persists, contact the service representative.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1852E Alias entry "name" points to itself.

Explanation: The ldif2ds utility encountered an alias entry in which a value of the aliasedObjectName attribute is the same as the distinguished name of the alias entry. This would cause an infinite loop when dereferencing the entry, thus is not allowed. A previous message indicates the name of the LDIF file containing the entry.

In the message text:

name

Example: None.

System action: The utility skips the rest of this entry and continues the check phase with the next complete entry. No entries are prepared or loaded.

Operator response: None.

User response: None.

System programmer response: None.
Administrator response: Change the aliasedObjectName attribute value so that it is not the distinguished name of the entry or remove the alias entry from the LDIF file. Then restart the program.

Problem determination: Not applicable.

Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1853E The JCL dataset must contain fixed-length 80-byte records.

Explanation: The ldif2ds utility has determined that the format of the JCL dataset is not correct. The JCL dataset must be a PDS or PDSE with a record format of RECFM=F or RECFM=FB and with a logical record length of LRECL=80. The name of this dataset is dsprefix.BULKLOAD.JCL, where dsprefix is the value of the -o command-line parameter.

Example: None.

System action: The program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Re-allocate the JCL dataset. Then restart the program.

Problem determination: Not applicable.

Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1854E Unable to retrieve file information:

error_code
reason_code - error_text

Explanation: The utility could not obtain file information for an open file. For the ldif2ds utility, the file is dsprefix.BULKLOAD.JCL, where dsprefix is the value of the -o command-line parameter. Refer to the description of fldata() in [IBM Z/OS XL C/C++ Run-Time Library Reference] for more information on the error.

In the message text:

error_code
    Error code from fldata()

reason_code
    Reason code from fldata()

Example: None.

System action: The program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Use the information in the message to correct the error. Then restart the program.

Problem determination: Not applicable.

Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD1855E The SYSTEM member contains an unrecognized directive: value.

Explanation: The ldif2ds utility has found a record that it does not support in the SYSTEM member of the JCL dataset. The supported records begin with # (a comment), HLQ, or JOBCARD. The name of the JCL dataset is dsprefix.BULKLOAD.JCL, where dsprefix is the value of the -o command-line parameter.

In the message text:

value
    Unrecognized directive

Example: None.

System action: The program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Remove the unsupported record from the SYSTEM member of the JCL dataset. Then restart the program.

Problem determination: Not applicable.

Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.
The HLQ directive in the SYSTEM member is not valid.

**Explanation:** The ldif2ds utility has found a value that is not supported for the HLQ record in the SYSTEM member of the JCL dataset. This value is the high-level-qualifier of the DB2 datasets, and must be at most 35 characters long. The name of the JCL dataset is `dsprefix.BULKLOAD.JCL`, where `dsprefix` is the value of the `-o` command-line parameter.

**Example:** None.

**System action:** The program ends.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Correct the value specified on the HLQ record in the SYSTEM member of the JCL dataset. Then restart the program.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

The load datasets are not in the correct state.

**Explanation:** The ldif2ds utility has been invoked with the `-l` (load) command-line parameter to submit the database load jobs, but it has determined that the load datasets may not be valid. ldif2ds expects the load phase to be run after the new entries in the LDIF files have been prepared, by specifying the `-p` (prepare) command-line parameter. When the utility successfully completes the prepare phase, it sets the first record in the STATUS member of the JCL dataset to STATUS P. When the utility begins the load phase, it checks that the first record of the STATUS member is STATUS P. The load phase fails if there is no record or if the value is not correct. If the status is correct and the load phase completes successfully, then the status is reset to STATUS L. This prevents the ldif2ds utility from being run again to load the same data, which can result in a corrupted DB2 database that is not usable by the LDAP server. The name of the JCL dataset is `dsprefix.BULKLOAD.JCL`, where `dsprefix` is the value of the `-o` parameter.

**Note:** If the load phase is successful, then the load jobs have been successfully submitted, but this does not indicate that the load jobs have ended successfully. The processing of the load jobs by DB2 is outside the scope of the ldif2ds utility. Review the output generated by each load job to determine if it is successful.

**Example:** None.

**System action:** The program ends.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** If the ldif2ds utility has not yet be run with the `-p` command-line parameter to prepare the entries for loading, do that before using the `-l` command-line parameter to load the data. Both parameters can also be specified at the same time. Otherwise, if it is certain that the load datasets contain valid load data, prepared using the current LDAP server schema and the current TDBM database to be loaded, and that the data has not already been loaded into the database, then edit the STATUS member of the JCL dataset, set the first record to STATUS P, and then restart the program to load the entries.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.
GLD1859E Unable to allocate an internal reader:

Error error_code, Reason reason_code.

Explanation: The ldif2ds utility has been unable to allocate an internal reader, needed to submit the load jobs. Refer to the description of dynalloc() in [Z/OS XL C/C++ Run-Time Library Reference for more information on the error.

In the message text:

error_code

   Error code from dynalloc()

reason_code

   Reason code from dynalloc()

Example: None.

System action: The program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the information in the message to correct the error. Then restart the program. If the problem persists, contact the service representative.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1860I

[-?]where:

-b creator
Creator/modifier DN to use in entries without them.

-c Check that LDIF entries are complete and acceptable.

-d debugLevel Level of debug messages to be created.

-e ldifListfile
Name of a file containing a list of LDIF input files.

-f confFile Name of the LDAP configuration file.

-g LDIF files unloaded in genealogical (parent/child) order.

-i ldifFile Names of one or more LDIF input files.

-j Use DB2 logging when adding the entries.

-k keyFile Name of a file containing LDAP encryption keys.

-l Invoke the DB2 Load utility to load the TDBM
database.

-o outHlq Highlevel qualifier of the output datasets.

-p Prepare DB2 load files and JCL from the LDIF
entries.

-q summaryFreq Number of entries handled between status
messages.

-? Display the usage

Explanation: The ldif2ds utility help and usage menu.

In the message text:

utility_name

   Utility name

Example: None.

System action: The program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1862E The following specified options are ignored when performing phase:

options.

Explanation: The ldif2ds utility has been invoked with one or more command-line parameters that do not apply to the requested phases of processing (check,
prepare, and/or load). The extraneous parameters are ignored.

In the message text:

**phase**
Processing phases

**options**
Ignored command-line parameters

**Example:** None.

**System action:** The utility continues.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** None.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD1863E** The '**attribute**' attribute cannot be used in the entry distinguished name for entry '**name**'.

**Explanation:** The ldif2ds utility encountered an entry which contains an attribute in its relative distinguished name (RDN) that is not allowed to be part of the RDN.

The following attributes cannot be used in an entry RDN: aclEntry, aclPropagate, entryOwner, ownerPropagate, ibm-EntryUUID, creatorsName, createTimestamp, modifiersName, modifyTimestamp. A previous message indicates the name of the LDIF file containing the entry.

In the message text:

**attribute**
Attribute type

**name**
Enter distinguished name

**Example:** None.

**System action:** The utility skips the rest of this entry and continues the check phase with the next complete entry. No entries are prepared or loaded.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Correct the RDN of the entry so that all its attributes are allowed in an RDN. Then restart the program.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD1864E** The dynamic group URL '**url**' on entry '**name**' is not valid.

**Explanation:** The ldif2ds utility encountered a dynamic group entry with a memberURL attribute value that is not supported. The value is indicated in the message. A previous message indicates the name of the LDIF file containing the entry. The format of a dynamic group URL is 'ldap:///?dn=scope?filter', where **dn** is the distinguished name of the base entry for the search, **scope** is the search scope, and **filter** is the search filter. The valid values for the search scope are base, one, and sub. All of the attribute types specified in the search filter must be defined in the LDAP server schema and each assertion value must conform to the matching rule for the associated attribute type. BINARY attribute types cannot be specified in a search filter.

In the message text:

**url** Dynamic group URL

**name**
Enter distinguished name

**Example:** None.

**System action:** The utility skips the rest of this entry and continues the check phase with the next complete entry. No entries are prepared or loaded.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Ensure that the memberURL attribute contains valid values for the distinguished name, search scope and search filter. Then restart the program.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.
GLD1865E  An unsupported value 'value' is specified for attribute 'ref' in entry 'name'.

Explanation: The ldif2ds utility encountered a referral entry with a ref attribute value that is not supported by the LDAP server. The value is indicated in the message. A previous message indicates the name of the LDIF file containing the entry. If the value is '', then a 0-length string was specified for the value. This could occur if the attribute is specified without a value.

In the message text:

value
  Unsupported attribute value

name
  Entry distinguished name

Example: None.

System action: The utility skips the rest of this entry and continues the check phase with the next complete entry. No entries are prepared or loaded.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the error text in the message to ensure that the indicated parent entry contains the correct objectclasses for the indicated child entry. Then restart the program.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1867E  Entry 'cn=localhost' is not allowed an objectclass value of 'ibm-replicationContext'.

Explanation: The ldif2ds utility encountered a cn=localhost entry with the ibm-replicationContext objectclass attribute value. The cn=localhost entry is not allowed to be the root of a replication context. A previous message indicates the name of the LDIF file containing the entry.

Example: None.

System action: The utility skips the rest of this entry and continues the check phase with the next complete entry. No entries are prepared or loaded.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Remove the objectclass value ibm-replicationContext from the cn=localhost entry. Then restart the program.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1866E  Entry 'parent_name' which is a parent of 'child_name' failed objectclass checking: error_text.

Explanation: The ldif2ds utility encountered a child entry with an objectclass that is not allowed based on the objectclass attribute values in the parent entry. A previous message indicates the name of the LDIF file containing the entry.

In the message text:

parent_name
  Parent entry distinguished name

child_name
  Child entry distinguished name

error_text
  Error message text

Example: None.

System action: The utility skips the rest of this entry and continues the check phase with the next complete entry. No entries are prepared or loaded.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Remove the objectclass value ibm-replicaSubEntry and ibm-replicaGateway objectclass values.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1868E  Entry 'name' must be a master server if it has both 'ibm-replicaSubEntry' and 'ibm-replicaGateway' objectclass values.

Explanation: The ldif2ds utility encountered an entry
with **objectclass** attribute values of **ibm-replicaSubEntry** and **ibm-replicaGateway** and the **ibm-replicaServerIsMaster** attribute value is not set to **TRUE**. A previous message indicates the name of the LDIF file containing the entry.

In the message text:

**name**

   Entry distinguished name

**Example:** None.

**System action:** The utility skips the rest of this entry and continues the check phase with the next complete entry. No entries are prepared or loaded.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Remove either the **ibm-replicaSubEntry** or **ibm-replicaGateway** **objectclass** attribute value or set the **ibm-replicaServerIsMaster** attribute value to **TRUE** to designate this server as a master server. Then restart the program.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD1871E** The replication filter *value* on entry *name* is not valid.

**Explanation:** The ldif2ds utility encountered a replication filter entry with an **ibm-replicationFilterAttr** attribute value that is not supported. The value is indicated in the message. A previous message indicates the name of the LDIF file containing the entry. The format of a replication filter is:

```
(objectclass=objectclass):[](attr1,attr2...) where
objectclass is an objectclass attribute value and attr1, attr2, and etc. is a list of attribute values to filter separated by a comma. The objectclass and any attribute values specified in the replication filter entry must be defined in the LDAP server schema.
```

In the message text:

**value**

   Filter value

**name**

   Entry distinguished name

**Example:** None.

**System action:** The utility skips the rest of this entry and continues the check phase with the next complete entry. No entries are prepared or loaded.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Ensure that the **ibm-replicationFilterAttr** attribute value in the replication filter entry is in the correct format. Then restart the program.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD1871E** Entry *name* has a duplicate replication consumer URL *url*.

**Explanation:** The ldif2ds utility encountered an entry with an **objectclass** attribute value of **ibm-replicationAgreement** that contains a value for **ibm-replicaURL** that already exists for the replication context. The values for the **ibm-replicaURL** attribute type under an advanced replication context must all be unique.

In the message text:

**name**

   Entry distinguished name

**url**

   Replication consumer URL

**Example:** None.

**System action:** The utility skips the rest of this entry and continues the check phase with the next complete entry. No entries are prepared or loaded.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Ensure that the **ibm-replicaURL** attribute contains a unique value within the scope of the replication context. Then restart the program.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.
GLD1873E Unable to decode the 'replicateOperationalAttributes' control found at line line_number of filename.

Explanation: The ldif2ds utility encountered an error decoding the replicateOperationalAttributes control. The replicateOperationalAttributes control contains base64 encoded values for the creatorsName, createTimestamp, modifiersName, and modifyTimestamp operational attribute values.

In the message text:

line_number
Line Number

filename
LDIF file name

Example: None.

System action: The utility skips the rest of this entry and continues the check phase with the next complete entry. No entries are prepared or loaded.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Ensure that the replicateOperationalAttributes control is properly encoded. Then restart the program. If the problem persists, remove the replicateOperationalAttributes control from the LDIF file.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD1874E Replication context entry 'name' missing explicit propagating type specification.

Explanation: The ldif2ds utility encountered an error processing the entry specified in the message. If a replication context entry is not a suffix level entry, an aclEntry and entryOwner attribute value must be defined explicitly in that entry.

In the message text:

name
Entry distinguished name

type
Attribute type

Example: None.

System action: The utility skips the rest of this entry and continues the check phase with the next complete entry. No entries are prepared or loaded.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Add the specified attribute type to the entry. Then restart the program.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

Utility messages (2000)

GLD2001I No Directory Server service has been configured.

Explanation: No LDAP server backends have been configured.

Example: None.

System action: The program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Configure appropriate backends as needed in the configuration profile. Then restart the program.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD2002I Directory Server configuration utility has started.

Explanation: The dsconfig utility has started.

Example: None.

System action: The program continues.

Operator response: None.

User response: None.
GLD2003I Directory Server configuration utility has ended.

Explanation: The dsconfig utility has ended.

Example: None.

System action: The program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD2004D

Attention: The output data set name has been previously used. Existing members may be overwritten and data lost. Do you wish to continue? (yes/no)

Explanation: The output data set specified on the dsconfig command already contains output from a previous run. This prompt is asking the user if they wish to overwrite existing members in the output data set. If the output data set is currently being used for an LDAP server, a different output data set should be used for this invocation of the dsconfig utility.

In the message text:
name

Output data set name

Example: None.

System action: The utility continues after a yes response has been entered. The utility ends if the response is no.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD2005I Terminating upon user request.

Explanation: The dsconfig utility is terminating upon user request.

Example: None.

System action: The program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD2006I dsconfig usage message.

Explanation: The dsconfig utility help and usage menu.

Example: None.

System action: The program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD2007E A blank option was found in file filename and is not allowed.
Explanation: The dsconfig utility has detected a blank option in the input profile.
In the message text:
filename File that contains the blank option
Example: None.
System action: The program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Correct the blank option. Then restart the program.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD2008E The value for option name contains non-printable characters.
Explanation: The dsconfig utility has detected that the value of the indicated option contains characters that cannot be printed.
In the message text:
name Option name
Example: None.
System action: The program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Correct the value. Then restart the program.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD2016E ADDRMODE 'value' must be either 31 or 64.
Explanation: The ADDRMODE option must have a value of 31 or 64.
In the message text:
value ADDRMODE option value
Example: None.
System action: The program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Change the ADDRMODE value to either 31 or 64. Then restart the program.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
GLD2017E  *option with value 'current_value' in file filename was previously set to 'original_value'.*

**Explanation:** An option is specified more than once in the file. This option can only be specified once.

In the message text:

- **option**
  - Option name
- **current_value**
  - Current option value
- **filename**
  - File that contains the duplicate options
- **original_value**
  - Original option value

**Example:** None.

**System action:** The program ends.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Remove all but one of the duplicate options. Then restart the program.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

GLD2018E  *File filename: error_text*

**Explanation:** An error occurred while processing a file or data set.

In the message text:

- **filename**
  - File or data set associated with the error
- **error_text**
  - Error message text

**Example:** None.

**System action:** The program ends.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** None.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

GLD2019E  *option is not allowed in file filename.*

**Explanation:** The indicated option is not allowed in the indicated profile.

In the message text:

- **option**
  - Option name not allowed
- **filename**
  - File containing the incorrect option

**Example:** None.

**System action:** The program ends.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Remove the option. Then restart the program.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

GLD2020E  *Unable to allocate storage.*

**Explanation:** An attempt to allocate storage was unsuccessful.

**Example:** None.

**System action:** The program ends.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Increase the storage available for use by the utility. Then restart the program.
If the problem persists, contact the service representative.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD2021E** The configuration profile has not been specified.

**Explanation:** The configuration profile name was not specified in the `dsconfig` command.

**Example:** None.

**System action:** The program ends.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Re-issue the `dsconfig` command and specify the configuration profile using the `-i` command-line parameter.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD2022E** No network interface has been configured.

**Explanation:** No `LISTEN` option was found in the configuration profile.

**Example:** None.

**System action:** The program ends.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Add one or more appropriate `LISTEN` options to the configuration profile. Then restart the program.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD2023E** `option` in file `filename` has no input value.

**Explanation:** A required option is missing.

In the message text:

`option`

Option name having no input value

`filename`

File in which option value must be included

**Example:** None.

**System action:** The program ends.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Add the required option to the indicated profile. Then restart the program.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD2024E** The value for `option1` must be different from the value for `option2`.

**Explanation:** Values for the indicated options must be unique.

In the message text:

`option1`

Option name

`option2`

Option name

**Example:** Database directory names cannot be the same for LDBM and file-based GDBM backends. DB2 database user IDs cannot be the same for TDBM and DB2-based GDBM backends. Similarly, DB2 database names must be unique.

**System action:** The program ends.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Change one of the option
values so that the two option values are different. Then restart the program.

Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD2025E Operand missing for command parameter 'parameter'.

Explanation: No value was specified for the indicated dsconfig command-line parameter. This parameter must have a value.

In the message text:

parameter
Command-line parameter

Example: None.
System action: The program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Edit the option value such that its length does not exceed the maximum allowed. Then restart the program.

Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD2027E Command input 'value', is not valid.

Explanation: The dsconfig utility detected an incorrect command-line parameter. Either the parameter is not known or the value specified for the parameter is not supported. See the usage notes in the documentation for the dsconfig utility for more information.

In the message text:

value
Incorrect command input

Example: None.
System action: The program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Re-issue the dsconfig command and either specify a valid value for the parameter or remove the parameter from the command (if it is optional).

Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD2026E Value 'value' for option option is too long. It must be number characters or less.

Explanation: The dsconfig utility has detected that an option value is longer than the maximum characters allowed for that option.

In the message text:

value
Option value

option
Option name

number
Maximum option length

Example: None.
System action: The program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Re-issue the dsconfig command and either specify a valid value for the parameter or remove the parameter from the command (if it is optional).

Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.
GLD2028E An internal program error occurred.
Explanation: The dsconfig utility detected an internal program error.
Example: None.
System action: The program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Contact the service representative.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD2256I utility_name: number entries have been processed.
Explanation: The utility has processed the number of entries indicated in the message. If errors are encountered during processing, additional messages are issued to indicate these errors. The number of entries processed may not match the number of entries present in the output LDIF file if errors are encountered during processing.

GLD2257I utility_name has completed successfully.
Explanation: The utility has successfully completed.
In the message text:
utility_name
Utility name
number
Number of entries processed
Example: None.
System action: The program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD2258A utility_name has failed.
Explanation: The utility has ended after encountering a severe error. Previous messages issued by the utility indicate the actual problem.
In the message text:
utility_name

Utility name

Example:  None.

System action:  The program ends.

Operator response:  None.

User response:  None.

System programmer response:  None.

Administrator response:  Use the information provided by the previous error messages to correct the error. To obtain additional debug information, specify -d ALL on the command line of the utility. Then restart the program.

Problem determination:  Not applicable.

Source:  LDAP

Module:  None.

Routing code:  None.

Descriptor code:  None.

Automation:  Not applicable.

GLD2259I  utility_name has terminated because there are no entries to process.

Explanation:  The utility found no entries in the LDBM, TDBM, or CDBM backend to unload.

In the message text:

utility_name

Utility name

Example:  None.

System action:  The program ends.

Operator response:  None.

User response:  None.

System programmer response:  None.

Administrator response:  None.

Problem determination:  Not applicable.

Source:  LDAP

Module:  None.

Routing code:  None.

Descriptor code:  None.

Automation:  Not applicable.

GLD2260I  ds2ldif_usage_message

Explanation:  The ds2ldif utility help and usage menu.

In the message text:

utility_name

Utility name

Example:  None.

System action:  The program ends.

Operator response:  None.

User response:  None.

System programmer response:  None.

Administrator response:  None.

Problem determination:  Not applicable.

Source:  LDAP

Module:  None.

Routing code:  None.

Descriptor code:  None.

Automation:  Not applicable.

GLD2262A  There are no TDBM, LDBM, CDBM, or schema backends which contain a subtree or filter DN entry for name.

Explanation:  The ds2ldif utility or the LDAP server is unable to find the subtree entry in a TDBM, LDBM, CDBM, or schema backend or the filter entry cannot be found within a TDBM, LDBM, or CDBM backend. The DN is the subtree or filter DN value specified on the -s (subtree DN) or the -q (filter DN) command-line parameter of the utility.

In the message text:

name

Distinguished name of entry

Example:  None.

System action:  The program ends.

Operator response:  None.

User response:  None.

System programmer response:  None.

Administrator response:  Ensure that the LDAP server configuration file used by the ds2ldif utility includes the subtree or filter entry within a TDBM, LDBM, CDBM, or schema backend. Verify that the TDBM, LDBM, CDBM, or schema backend is configured correctly and that the syntax of the DN specified on the -s (subtree DN) or the -q (filter DN) command-line parameter of the ds2ldif utility is correct. Then restart the program.

Problem determination:  Not applicable.
GLD2263E  
utility_name found more than one backend section. Either use the -s or -n option to specify which TDBM, LDBM, or CDBM section to process or remove all but one of the database sections from the configuration file.

Explanation: If there are more than one TDBM, LDBM, or CDBM backends present in the LDAP server configuration file, it is necessary to specify which backend needs to be unloaded by using the -n or -s command-line parameter on the ds2ldif utility. The utility is unable to determine which TDBM, LDBM, or CDBM backend needs to be unloaded if there are multiple TDBM, LDBM, or CDBM backends in the LDAP server configuration file.

In the message text:

utility_name
Utility name

Example: None.

System action: The program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: The ds2ldif utility provides two command-line parameters that are used to specify which TDBM, LDBM, or CDBM backend to process. These two parameters cannot be specified at the same time.

- The -s parameter specifies a subtree DN (distinguished name) whose entries are to be unloaded. The ds2ldif utility selects the TDBM, LDBM, or CDBM backend database section which contains this subtree from the LDAP server configuration file.

- The -n parameter indicates the name of a TDBM, LDBM, or CDBM backend whose entries are to be unloaded. This name is the optional fourth parameter that can be specified on the database option in the LDAP server configuration file. The ds2ldif utility selects the TDBM, LDBM, or CDBM database section with this name from the LDAP server configuration file.

Alternatively, modify the LDAP server configuration file and remove all of the TDBM, LDBM, or CDBM database sections except the one that needs to be unloaded.

Then restart the program.

Problem determination: Not applicable.

Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD2264E  
Unable to write record to output file

filename: error_code/reason_code - error_text

Explanation: The ds2ldif utility or the LDAP server encountered an error while attempting to write a record to the output file. The output file is either a Unix System Services file, a partitioned dataset, or a sequential dataset specified on -o command-line parameter of the ds2ldif utility. The error code, reason code, and error text are returned from one of the following: fputs(), fflush(), or fclose(). Refer to the descriptions of these routines in z/OS XL C/C++ Run-Time Library Reference for more information on the error.

In the message text:

filename
Output file name

error_code
Error code from function

reason_code
Reason code from function

error_text
Text corresponding to the error code

Example: None.

System action: The program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the information in the message to correct the error. Then restart the program.

Problem determination: Not applicable.

Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.
GLD2265A There are no TDBM, LDBM, or CDBM backends with name 'name'.

Explanation: The ds2ldif utility is unable to find a TDBM, LDBM, or CDBM backend in the LDAP server configuration file with the backend name indicated in the message. This backend name is the value specified on the -n command-line parameter of the ds2ldif utility.

In the message text:
name
  Backend name

Example: None.
System action: The program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD2266I Unloading directory data from backend named backend_name under subtree DN: entry_name.

Explanation: The ds2ldif utility is unloading the entries under the indicated subtree DN (distinguished name) in the TDBM, LDBM, or CDBM backend with the indicated name.

In the message text:
backend_name
  Backend name
entry_name
  Distinguished name

Example: None.
System action: The program continues.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD2267I Unloading the cn=schema entry.

Explanation: The ds2ldif utility is unloading the LDAP server schema entry.

Example: None.
System action: The program continues.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD2268E Unable to open file filename: error_code/reason_code - error_text

Explanation: The ds2ldif utility encountered an error while attempting to open the indicated file for writing. Refer to the description of fopen() in z/OS XL C/C++ Run-Time Library Reference for more information on the error.

In the message text:
filename
  File name
error_code
  Error code from fopen()
reason_code
  Reason code from fopen()
error_text
  Text corresponding to the error code

Example: None.
System action: The program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Use the information in the message to correct the error. Then restart the program.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD2269I ds2ldif utility is starting.
Explanation: The ds2ldif utility has started.
Example: None.
System action: The program continues.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD2270E Option 'option' is specified more than once with different values.
Explanation: The ds2ldif utility encountered an error because it detected multiple specifications of the same command-line parameter with different values. The utility is not able to determine which value to use for the parameter.
Example: None.
System action: The program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD2271E Do not specify both of the following options: 'option1' and 'option2'.
Explanation: The ds2ldif utility encountered an error because it detected that the two indicated parameters are both specified on the utility command line, but they are mutually exclusive. These two command-line parameters cannot be specified at the same time. Refer to the documentation of the ds2ldif utility for additional information.
Example: None.
System action: The program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Remove one of the mutually exclusive parameters from the command line of the ds2ldif utility. Then restart the program.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD2272I ds2ldif version version.release, Service level level.
Explanation: The ds2ldif utility with version, release, and service level indicated in the message is running.
Example: None.
System action: The program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Correct the command line of the ds2ldif utility to ensure that a parameter is only specified one time. Then restart the program.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.
GLD2273D Enter the LDAP administrator password to unload the directory:

Explanation: The ds2ldif utility has determined that it is necessary to perform an unloadRequest extended operation (OID 1.3.18.0.2.12.62) to unload the desired data. This is either because the -r parameter is specified on the ds2ldif command line or because the backend to be unloaded cannot successfully be started by the ds2ldif utility. Before attempting the unloadRequest extended operation, a connection must be established with the targeted LDAP server. This ensures that the LDAP server with the backend that is to be unloaded is running.

Example: None.

System action: The program continues.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD2274I Connecting to the LDAP server with the backend to unload.

Explanation: The ds2ldif utility has determined that it is necessary to perform an unloadRequest extended operation (OID 1.3.18.0.2.12.62) to unload the desired data. This is either because the -r parameter is specified on the ds2ldif command line or because the backend to be unloaded cannot successfully be started by the ds2ldif utility. Before attempting the unloadRequest extended operation, a connection must be established with the targeted LDAP server. This ensures that the LDAP server with the backend that is to be unloaded is running.

Example: None.

System action: The program continues.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD2275I Unloading directory data by using the unloadRequest extended operation.

Explanation: The ds2ldif utility is sending the unloadRequest extended operation (OID 1.3.18.0.2.12.62) to the LDAP server that is running to directly unload the desired data directory. The utility sends the necessary information, including the values of the subtree DN (-s utility command-line parameter), the backend name (-n parameter), and the LDIF output filename (-o parameter) on the unloadRequest extended operation to the LDAP server.

Example: None.

System action: The program continues.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.
GLD2276A The unloadRequest extended operation encountered an error: error_code - error_message.

Explanation: The unloadRequest extended operation (OID 1.3.18.0.2.12.62) encountered an error on the targeted LDAP server while attempting to unload the desired directory data.

In the message text:

error_code
unloadRequest server error code

error_message
unloadRequest server error message

Example: None.

System action: The program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Examine the ds2ldif command line options and verify that all values are printable characters. To obtain additional debug information, specify -d ALL on the command line of the utility. Then restart the utility. The error code is an internal error code that occurred during the encoding of the unloadRequest extended operation. If the problem persists, contact the service representative and provide the ds2ldif debug trace, the LDAP server configuration file, and the ds2ldif command that was attempted.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD2277A Unexpected error error_code occurred while decoding the unloadResponse extended operation.

Explanation: An unexpected error was encountered while attempting to BER decode the unloadResponse extended operation (OID 1.3.18.0.2.12.63) that was received from the targeted LDAP server. The unloadResponse extended operation is not valid.

In the message text:

error_code
unloadResponse decode error

Example: None.

System action: The program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Examine the ds2ldif command line options and verify that all values are printable characters. To obtain additional debug information, specify -d ALL on the command line of the utility. Then restart the utility. The error code is an internal return code. To determine why the LDAP server constructed an unloadResponse extended operation that is not valid, turn on LDAP debug tracing by specifying -d DEBUG on both the ds2ldif utility and the LDAP server command lines (or use the LDAP server DEBUG operator modify command). Then restart the utility and the LDAP server if it is not running. If the problem persists, contact the service representative and provide the LDAP server configuration file, the ds2ldif debug trace, the ds2ldif command that was attempted, and the LDAP server command line options and verify that all values are printable characters.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD2277A Unexpected error error_code occurred while encoding the unloadRequest extended operation.

Explanation: An unexpected error was encountered while attempting to BER encode the unloadRequest extended operation (OID 1.3.18.0.2.12.62).

In the message text:

error_code
unloadRequest encode error

Example: None.

System action: The program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: The error code is an internal return code. To determine why the LDAP server constructed an unloadResponse extended operation that is not valid, turn on LDAP debug tracing by specifying -d DEBUG on both the ds2ldif utility and the LDAP server command lines (or use the LDAP server DEBUG operator modify command). Then restart the utility and the LDAP server if it is not running. If the problem persists, contact the service representative and provide the LDAP server configuration file, the ds2ldif debug trace, the ds2ldif command that was attempted, and the LDAP server command line options and verify that all values are printable characters.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.
provide the LDAP server and ds2ldif debug traces, the LDAP server configuration file, and the ds2ldif command that was attempted.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD2279A** Error encountered in getpass():

<code>error_code/reason_code - error_text</code>

**Explanation:** An error was encountered while attempting to obtain the password of the administrator DN so that an <code>unloadRequest</code> extended operation can be attempted. Refer to the description of <code>getpass()</code> in <i>z/OS XL C/C++ Run-Time Library Reference</i> for more information on the error.

In the message text:

- error_code
  - Error code from <code>getpass()</code>
- reason_code
  - Reason code from <code>getpass()</code>
- error_text
  - Text corresponding to the error code

**Example:** None.

**System action:** The program ends.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Verify that the LDIF output filename that is specified on the -o command-line parameter is in one of the following formats:

- Fully-qualified Unix System Services file name
  - Example: /var/ldap/output.ldif
- Fully-qualified file in a sequential dataset
  - Example: //USER.OUTPUT.LDIF
- Fully-qualified file in a partitioned dataset
  - Example: //USER.OUTPUT(LDIF)
- Fully-qualified file specified as a DD card in JCL
  - Example: DD:OUTNAME

The fully-qualified Unix System Services file names must start with an / and represent the pathname from the root directory. The fully-qualified sequential and partitioned dataset names must start with the following two characters: // When ds2ldif is invoked from the shell, quotes must be used around the dataset name. For example, the sequential dataset name above would be specified as "//USER.OUTPUT.LDIF". The DD card specified in JCL must start with the following three characters: DD:. However, a DD card cannot be used to specify the LDIF output filename if an <code>unloadRequest</code> extended operation is to be performed. Correct the -o parameter on the command line of the ds2ldif utility. Then restart the program.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD2280A** A fully qualified LDIF output filename must be specified on the -o option.

**Explanation:** The ds2ldif utility encountered an error in the LDIF output filename specified on the -o parameter on the utility command line. The <code>ds2ldif</code> utility requires a fully-qualified file name on the -o parameter.

**Example:** None.

**System action:** The program ends.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Verify that the LDIF output filename that is specified on the -o command-line parameter is in one of the following formats:

- Fully-qualified Unix System Services file name
  - Example: /var/ldap/output.ldif
- Fully-qualified file in a sequential dataset
  - Example: //USER.OUTPUT.LDIF
- Fully-qualified file in a partitioned dataset
  - Example: //USER.OUTPUT(LDIF)
- Fully-qualified file specified as a DD card in JCL
  - Example: DD:OUTNAME

The fully-qualified Unix System Services file names must start with an / and represent the pathname from the root directory. The fully-qualified sequential and partitioned dataset names must start with the following two characters: // When ds2ldif is invoked from the shell, quotes must be used around the dataset name. For example, the sequential dataset name above would be specified as "//USER.OUTPUT.LDIF". The DD card specified in JCL must start with the following three characters: DD:. However, a DD card cannot be used to specify the LDIF output filename if an <code>unloadRequest</code> extended operation is to be performed. Correct the -o parameter on the command line of the ds2ldif utility. Then restart the program.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.
GLD2281A  Error return_code reported by ldap_extended_operation().

Explanation: The ds2ldif utility encountered an error in ldap_extended_operation() while sending the unloadRequest extended operation to the targeted LDAP server.

In the message text:

return_code
Return code from ldap_extended_operation()

Example: None.

System action: The program ends
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: An error occurred in the ldap_extended_operation() routine. The following are the common client errors:

81  The network connection to the targeted LDAP server has failed
89  A parameter specified on ldap_extended_operation is not valid
90  Insufficient storage is available
92  The LDAP protocol version must be V3 in order to initiate the unloadRequest extended operation
252 An unbind request has been issued for the LDAP handle

The following are the common server error that are returned on the ldap_extended_operation() routine:

2  The server does not support the unloadRequest extended operation
12 A critical server control is either not recognized or is not supported for the unloadRequest extended operation.
53 The server is unable to perform the requested unloadRequest extended operation.

Depending upon the return code from ldap_extended_operation(), it may be necessary to correct the LDAP server configuration file, restart the LDAP server, and restart the ds2ldif utility. To obtain additional debug information, turn on LDAP debug tracing by specifying -d ALL on both the ds2ldif utility and the LDAP server command lines (or use the LDAP server DEBUG operator modify command). Then restart the utility and the LDAP server if it is not running. If the problem persists, contact the service representative and provide the LDAP server and ds2ldif debug traces, the LDAP server configuration file, and the ds2ldif command that was attempted.

Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD2282A  An unexpected error occurred during the running of ds2ldif.

Explanation: The ds2ldif utility encountered an unexpected error during its processing.

Example: None.

System action: The program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: To obtain additional debug information, specify -d ALL on the command line of the utility. Then restart the program. Analyze the debug trace output and correct the error. If the problem persists, contact the service representative and provide the ds2ldif debug trace, the LDAP server configuration file, and the ds2ldif command that was attempted.

Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD2283A  A DD card is not allowed to be specified when performing an unloadRequest extended operation.

Explanation: The ds2ldif utility does not allow a DD: card to be specified on the -o option when performing an unloadRequest extended operation. The ds2ldif utility will perform an unloadRequest extended operation when the LDAP server is already running or because the -r option is specified on the ds2ldif command line.

Example: None.

System action: The program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Correct the -o parameter on
the command line of the *ds2ldif* utility to no longer specify the DD: card. In order to use the same LDIF output filename that is specified on the DD card, update the -o option to specify that filename. If the *unloadRequest* extended operation is not desired, stop the LDAP server and do not specify the -r option on the *ds2ldif* command line. Then restart the program.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD2284A** There is no TDBM, LDBM, or CDBM backend present in the LDAP server configuration file to unload.

**Explanation:** When the *ds2ldif* utility is invoked without the -n or -s option, the utility searches for a TDBM, LDBM, or CDBM backend in the LDAP server configuration file. The utility was unable to find a TDBM, LDBM, or CDBM backend to unload.

**Example:** None.

**System action:** The program ends.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Ensure that the correct LDAP server configuration file has been specified on the -f option or the CONFIG DD card in JCL. The only backends that can be unloaded with the *ds2ldif* utility are the schema, LDBM, TDBM, and CDBM backends. If the schema backend is to be unloaded, specify the -s option with a value of cn=schema. If a TDBM, LDBM, or CDBM backend is to be unloaded, either make sure that there is only a single TDBM, LDBM, or CDBM backend in the LDAP server configuration file or use the -n or -s option to indicate which one of the TDBM, LDBM, or CDBM backends to unload. Then restart the program.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD2285E** The "option1" option requires the "option2" option.

**Explanation:** The *ds2ldif* utility encountered an error because it detected that *option1* was missing required *option2*. If *option1* is specified then *option2* must be specified. Refer to the documentation of the *ds2ldif* utility for additional information.

In the message text:

*option1*  
Command-line parameter

*option2*  
Command-line parameter

**Example:** None.

**System action:** The program ends.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Either remove *option1* from the *ds2ldif* command line or specify both *option1* and *option2* on the *ds2ldif* command line. Then restart the program.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD2286I** Filtering directory data being unloaded using filters in filter DN: *entry_name*.

**Explanation:** The *ds2ldif* utility is using filters that are specified in the *ibm-replicationfilterattr* attribute values contained in the indicated filter DN (distinguished name). These filters may prevent some entries from being unloaded or may remove some attribute types and values from some unloaded directory entries.

In the message text:

*entry_name*  
Distinguished name

**Example:** None.

**System action:** The program continues.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** None.

**Problem determination:** Not applicable.
GLD2401E Encrypt all passwords that are presently unencrypted, AES encrypted, or DES encrypted (yes/no)?

Explanation: The db2pwden utility replaces any clear text (unencrypted), AES encrypted, or DES encrypted userPassword attribute values that exist in the directory with encrypted userPassword values based upon the setting of the pwEncryption option in the LDAP server configuration file. This message prompts the user of the db2pwden utility to ensure that encryption of the userPassword attribute values is really desired.

Example: None.

System action: If the response is yes, y, or Y, the utility continues. Otherwise, the utility ends without encrypting userPassword attribute values.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: If it is desired to encrypt any clear text (unencrypted), AES encrypted, or DES encrypted userPassword attribute values in the directory, enter yes, y, or Y. Otherwise, enter any other response to end the program.

Problem determination: Not applicable.

Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD2402A No base is defined.

Explanation: The utility encountered an error because a base DN (distinguished name) is not specified for the utility. The base DN can be specified on the -b command-line parameter of the utility or set on the LDAP_BASEDN environment variable. If set both ways, the command line value is used.

Example: None.

System action: The program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Specify the base DN (distinguished name) either by using the -b command-line parameter of the utility or by setting it on the LDAP_BASEDN environment variable. Then restart the program.

Problem determination: Not applicable.

Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD2403A db2pwden ends without encrypting passwords.

Explanation: The db2pwden utility has ended and no passwords have been encrypted. Either the user responded to the utility prompt to end the utility or an error occurred during utility processing.

Example: None.

System action: The program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: If the utility ended due to an error, refer to any previous error messages and correct any errors that are identified. To obtain additional debug information, specify -d ALL on the command line of the utility. Then restart the program.

Problem determination: Not applicable.

Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD2410A Memory allocation failed.

Explanation: An attempt to allocate storage was unsuccessful.

Example: None.

System action: The program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Increase the storage...
available for use by the utility. Then restart the program. If the problem persists, contact the service representative.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD2411A** The only supported mechanisms are EXTERNAL, GSSAPI, CRAM-MD5, and DIGEST-MD5.

**Explanation:** The utility encountered an error because an incorrect authentication mechanism is specified on the -m or -S command-line parameter.

**Example:** None.

**System action:** The program ends.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Correct the -m or -S command-line parameter by specifying a valid authentication mechanism. The only supported authentication mechanisms are EXTERNAL, GSSAPI, CRAM-MD5, or DIGEST-MD5. Then restart the program.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD2417A** Error return_code reported parsing LDAP results.

**Explanation:** The ds2ldif utility encountered an error while attempting to parse LDAP results from the targeted LDAP server when performing the unloadRequest extended operation. The return code is from the ldap_result() routine. Refer to the description of this routine in [IBM Tivoli Directory Server Client Programming for z/OS](http://www.ibm.com) for more information on the error.

In the message text:

```
return_code
   Return code from parsing results
```

**Example:** None.

**System action:** The program ends.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Use the information in the message to correct the error. Verify that the targeted LDAP server is still running. To obtain additional debug information, specify -d ALL on the command line of the program. Then restart the utility. If the problem persists, contact the service representative.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.
GLD2425A  A user name is required when doing a DIGEST-MD5 bind.

Explanation:  When the -m DIGEST-MD5 or -S DIGEST-MD5 command-line parameter of the db2pwden utility is specified, the -U (user name) command-line option must also be specified.

Example:  None.

System action:  The program ends.

Operator response:  None.

User response:  None.

System programmer response:  None.

Administrator response:  Specify the user name on the -U command-line parameter of the db2pwden utility or change the -m or -S command-line parameter. Then restart the utility.

Problem determination:  Not applicable.

Source:  LDAP

Module:  None.

Routing code:  None.

Descriptor code:  None.

Automation:  Not applicable.

GLD2429A  Credentials are not valid for the specified LDAP server.

Explanation:  The ds2ldif utility encountered an error while attempting to perform the LDAP administrator authentication for the unloadRequest extended operation. The credentials specified on the -w command-line parameter are not valid for the adminDN configuration option.

Example:  None.

System action:  The program ends.

Operator response:  None.

User response:  None.

System programmer response:  None.

Administrator response:  Specify the correct credentials for the adminDN in the LDAP server configuration file on the -w command-line parameter of the ds2ldif utility. Then restart the program.

Problem determination:  Not applicable.

Source:  LDAP

Module:  None.

Routing code:  None.

Descriptor code:  None.

Automation:  Not applicable.

GLD2426A  Debug value is not valid.

Explanation:  The utility encountered an error with the debug value that is specified on the -d command-line parameter of the utility. Refer to the documentation for the utility for more information on valid debug values.

Example:  None.

System action:  The program ends.

Operator response:  None.

User response:  None.

System programmer response:  None.

Administrator response:  Specify a valid debug value on the -d command-line parameter of the utility or remove the parameter. Then restart the program.

Problem determination:  Not applicable.

Source:  LDAP

Module:  None.

Routing code:  None.

Descriptor code:  None.

Automation:  Not applicable.

TDBM messages (3000)
GLD3301E Unable to load type backend named name because attribute type attribute is not defined.

**Explanation:** The LDAP server or utility found an attribute type used by an entry in the indicated backend is not defined in the LDAP schema.

In the message text:
- **type** Backend type
- **name** Backend name
- **attribute** Undefined attribute type

**Example:** None.

**System action:** The backend does not start. If the srvStartUpError option in the LDAP server configuration file is set to ignore, the LDAP server continues to run with those backends that successfully start. If the srvStartUpError option is set to terminate (this is the default if the configuration option is not specified), the program ends. The utility ends regardless of the option value.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** If the backend is needed, restart the LDAP server without the backend section in the LDAP server configuration file and add the missing attribute type to the LDAP server schema. Then restore the backend section and restart the program.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

GLD3302E Unable to load type backend named name because object class objectclass is not defined.

**Explanation:** The LDAP server or utility found an object class used by an entry in the indicated backend is not defined in the LDAP schema.

In the message text:
- **type** Backend type
- **name** Backend name
- **objectclass** Undefined object class

**Example:** None.

**System action:** The backend does not start. If the srvStartUpError option in the LDAP server configuration file is set to ignore, the LDAP server continues to run with those backends that successfully start. If the srvStartUpError option is set to terminate (this is the default if the configuration option is not specified), the program ends. The utility ends regardless of the option value.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** None.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

GLD3303E TDBM backend specified for a non-TDBM database.

**Explanation:** The LDAP server or utility found that the TDBM backend DLL, GLDBTD31, is specified on a database option in the LDAP server configuration file but the type parameter on the option is not TDBM.

**Example:** None.

**System action:** The TDBM backend does not start. If the srvStartUpError option in the LDAP server configuration file is set to ignore, the LDAP server continues to run with those backends that successfully start. If the srvStartUpError option is set to terminate (this is the default if the configuration option is not specified), the program ends. The utility ends regardless of the option value.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Correct the database option in the LDAP server configuration file so that the DLL and backend type match. Restart the program if it ended or if the TDBM backend is needed.

**Problem determination:** Not applicable.
GLD3304E  type initialization terminated because DB2 is not available.

Explanation: The LDAP server or utility cannot initialize the indicated backend because DB2 is not available.

In the message text:

- **type**: Backend type

Example: None.

System action: The backend does not start. If the `srvStartUpError` option in the LDAP server configuration file is set to `ignore`, the LDAP server continues to run with those backends that successfully start. If the `srvStartUpError` option is set to `terminate` (this is the default if the configuration option is not specified), the program ends. If the error occurs while processing a client request, the LDAP server continues but the client request fails. The utility ends in all cases.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Correct the DB2 problem and ensure that DB2 is active. Restart the program if it ended or if the backend is needed.

Problem determination: Not applicable.

Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

---

GLD3305E  Native return code code, SQL state state, SQL message: text

Explanation: The LDAP server or utility encountered an error while performing a DB2 database operation. This message provides information about the error. Refer to [DB2 Messages and Codes] for more information on DB2 errors.

In the message text:

- **code**: Native return code
- **state**: SQL state
- **text**: SQL message text

Example: None.

System action: If the error occurs during backend initialization, the backend does not start. If the `srvStartUpError` option in the LDAP server configuration file is set to `ignore`, the LDAP server continues to run with those backends that successfully start. If the `srvStartUpError` option is set to `terminate` (this is the default if the configuration option is not specified), the program ends. If the error occurs while processing a client request, the LDAP server continues but the client request fails. The utility ends in all cases.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the information in the message to correct the error. Then retry the client operation or restart the program if it stopped or if the backend is needed. Contact your DB2 database administrator if you are unable to resolve the problem.

Problem determination: Not applicable.

Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.
(this is the default if the configuration option is not specified), the program ends. If the error occurs while processing a client request, the LDAP server continues but the client request fails. The utility ends in all cases.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Use the information in the message to correct the error. Then retry the client operation or restart the program if it stopped or if the backend is needed. Contact your DB2 database administrator if you are unable to resolve the problem.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

**GLD3307E** Unable to create key identifier: 

**error_code**/ **reason_code** - **error_text**

**Explanation:** The LDAP server or utility encountered an error when creating a key identifier. Refer to the [description of pthread_key_create() in z/OS XL C/C++ Run-Time Library Reference](https://www.ibm.com/support/knowledgecenter/SST7JU_11.1.0/c/stdlib/pthread_key_create.htm) for more information on the error.

In the message text:

- **error_code**
  - Error code from **pthread_key_create()**

- **reason_code**
  - Reason code from **pthread_key_create()**

- **error_text**
  - Error text corresponding to the error code

**Example:** None.

**System action:** If the error occurs during backend initialization, the backend does not start. If the `srvStartUpError` option in the LDAP server configuration file is set to `ignore`, the LDAP server continues to run with those backends that successfully start. If the `srvStartUpError` option is set to `terminate` (this is the default if the configuration option is not specified), the program ends. If the error occurs while processing a client request, the LDAP server continues but the client request fails. The utility ends in all cases.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Retry the client operation or restart the program if it stopped or if the backend is needed. If the problem persists, contact the service representative.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

**GLD3307E** An internal type backend error has occurred.

**Explanation:** The LDAP server or utility has detected an internal programming error.

In the message text:

- **type**
  - Backend type

**Example:** None.

**System action:** If the error occurs during backend initialization, the backend does not start. If the `srvStartUpError` option in the LDAP server configuration file is set to `ignore`, the LDAP server continues to run with those backends that successfully start. If the `srvStartUpError` option is set to `terminate` (this is the default if the configuration option is not specified), the program ends. If the error occurs while processing a client request, the LDAP server continues but the client request fails. The utility ends in all cases.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Retry the client operation or restart the program if it stopped or if the backend is needed. If the problem persists, contact the service representative.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

**GLD3309E** Unable to get thread-specific value:

**error_code**/ **reason_code** - **error_text**

**Explanation:** The LDAP server or utility is unable to retrieve a thread-specific value. Refer to the description of `pthread_getspecific()` in [z/OS XL C/C++ Run-Time Library Reference](https://www.ibm.com/support/knowledgecenter/SST7JU_11.1.0/c/lib/CLibraryReference/pthread_getspecific.htm) for more information on the error.

In the message text:
**error_code**
Error code from `pthread_getspecific()`

**reason_code**
Reason code from `pthread_getspecific()`

**error_text**
Error text corresponding to the error code

**Example:** None.

**System action:** If the error occurs during backend initialization, the backend does not start. If the `srvStartUpError` option in the LDAP server configuration file is set to `ignore`, the LDAP server continues to run with those backends that successfully start. If the `srvStartUpError` option is set to `terminate` (this is the default if the configuration option is not specified), the program ends. If the error occurs while processing a client request, the LDAP server continues but the client request fails. The utility ends in all cases.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Use the information in the message to correct the error. Then retry the client operation or restart the program if it stopped or if the backend is needed. If the problem persists, contact the service representative.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD3310E** Database access unavailable for type backend named *name* because DB2 is terminating.

**Explanation:** The DB2 database manager is terminating and the `db2Terminate` option in the LDAP server configuration file is set to `recovery` or `restore` (this is the default value).

In the message text:

- **type** Backend type
- **name** Backend name

**Example:** None.

**System action:** The LDAP server continues to run but access to the indicated backend is not available until the DB2 database is available. Client requests to that backend are rejected.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Restart the DB2 database manager.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD3312I Database access available for type backend named name because DB2 has restarted.

Explanation: The DB2 database manager is restarting and the LDAP server can once more access the indicated backend.

In the message text:
  type Backend type
  name Backend name

Example: None.

System action: The LDAP server continues. Client requests to that backend are now processed.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.
Source: LDAP
Module: None.

Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD3313E Column name in the owner.table table is not defined correctly.

Explanation: The LDAP server or utility has found that a required column in the indicated DB2 table is not defined correctly. Either a non-modifiable column has the wrong length or a modifiable column has a length less than 8.

In the message text:
  name Column name
  owner Database owner
  table Database table

Example: None.

System action: The backend does not start. If the srvStartUpError option is set to terminate (this is the default if the configuration option is not specified), the program ends. The utility ends regardless of the option value.
Operator response: None.
User response: None.
System programmer response: None.

Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD3314E Column name in the owner.table table is not defined correctly.

Explanation: The LDAP server or utility has found that a required column in the indicated DB2 table is not defined correctly. Either a non-modifiable column has the wrong length or a modifiable column has a length less than 8.

In the message text:
  name Column name
  owner Database owner
  table Database table

Example: None.

System action: The backend does not start. If the srvStartUpError option in the LDAP server configuration file is set to ignore, the LDAP server continues to run with those backends that successfully start. If the srvStartUpError option is set to terminate (this is the default if the configuration option is not specified), the program ends. The utility ends regardless of the option value.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Review the SPUFI script used to create the DB2 database and ensure that the column sizes are acceptable for the IBM Tivoli Directory Server for z/OS LDAP server. A DB2 database created using the SPUFI script shipped in the Integrated Security Services LDAP server may not have acceptable column sizes for the IBM Tivoli Directory Server for z/OS LDAP server. Correct the DB2 database. Then restart the program if it ended or if the backend is needed.

Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD3315E type database version major.minor is not supported.

Explanation: The DB_VERSION value in the DB2 DIR_MISC table for this backend is set to an unsupported database version for this level of the LDAP server.
In the message text:
  type
    Backend type
  major
    database version number
  minor
    database version number

Example: None.
System action: The backend does not start. If the srvStartUpError option in the LDAP server configuration file is set to ignore, the LDAP server continues to run with those backends that successfully start. If the srvStartUpError option is set to terminate (this is the default if the configuration option is not specified), the program ends. The utility ends regardless of the option value.
Operator response: None.
User response: None.
System programmer response: None.

Administrator response: If the program ended or if the backend is needed, restart the LDAP server without the TDBM backend section in the LDAP server configuration file and modify the LDAP server schema to specify a shorter name for the attribute type. Then restore the backend section in the configuration file and restart the program.

Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD3316E Attribute type identifier 'name' is too long.

Explanation: The LDAP server or utility found an attribute type in the schema whose identifier is too long. The TDBM database limits the maximum length of attribute type identifiers to 200 characters.
In the message text:
  name
    Attribute type identifier

Example: None.

System action: The backend does not start. If the srvStartUpError option in the LDAP server configuration file is set to ignore, the LDAP server continues to run with those backends that successfully start. If the srvStartUpError option is set to terminate (this is the default if the configuration option is not specified), the program ends. The utility ends regardless of the option value.
Operator response: None.
User response: None.

System programmer response: None.

Administrator response: Either migrate the DB2 database to an appropriate level for this LDAP server or upgrade the LDAP server to a level that supports the database version. Then restart the program if it ended or if the backend is needed.

Problem determination: Not applicable.
Source: LDAP
Module: None.

GLD3317E Object class identifier 'name' is too long.

Explanation: The LDAP server or utility found an object class in the schema whose identifier is too long. The TDBM database limits the maximum length of object class identifiers to 200 characters.
In the message text:
  name
    Object class name

Example: None.
System action: The backend does not start. If the
**srvStartUpError** option in the LDAP server
configuration file is set to **ignore**, the LDAP server
continues to run with those backends that successfully
start. If the **srvStartUpError** option is set to **terminate**
(this is the default if the configuration option is not
specified), the program ends. The utility ends regardless
of the option value.

Operator response: None.
User response: None.
System programmer response: None.
Administrator response: If the program ended or if
the backend is needed, restart the LDAP server without
the TDBM backend section in the LDAP server
configuration file and modify the LDAP server schema
to specify a shorter name for the object class. Then
restore the backend section in the configuration file and
restart the program.

Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

**GLD3318W** Database suffix 'suffix' is not
configured.

Explanation: A backend directory contains a suffix
entry that is not in the list of suffixes specified by the
**suffix** options for this backend in the LDAP server
configuration file.

In the message text:

```
suffix
  Database suffix distinguished name
```

Example: None.

System action: Backend initialization continues, but
directory entries under this suffix are not accessible.

Operator response: None.
User response: None.
System programmer response: None.

**GLD3319E** Database suffix 'directory_suffix'
overlaps configured suffix
'configured_suffix'.

Explanation: The LDAP server or utility found that a
backend directory contains a suffix entry that is an
ancestor or descendant of a suffix in the list of suffixes
for this backend. The list of suffixes is specified by the
**suffix** options for this backend in the LDAP server
configuration file.

In the message text:

```
directory_suffix
  Distinguished name of existing suffix in directory
configured_suffix
  Configured suffix distinguished name
```

Example: None.

System action: The backend does not start. If the
**srvStartUpError** option in the LDAP server
configuration file is set to **ignore**, the LDAP server
continues to run with those backends that successfully
start. If the **srvStartUpError** option is set to **terminate**
(this is the default if the configuration option is not
specified), the program ends. The utility ends regardless
of the option value.

Operator response: None.
User response: None.
System programmer response: None.

**GLD3320E** **type** backend named name database
XCF data record is not valid.

Explanation: The LDAP server cannot decode a
cross-system notification containing information about a
change to a database.

In the message text:

```
type
  Backend type
```
name
Backend name

Example: None.

System action: The LDAP server continues, but the backend is not notified of the change. As a result, the backend database on this LDAP server may be out of sync with the other LDAP servers in the cross-system group. Thus, this server may return different results for an LDAP request than the other servers. There are two additional consequences for a TDBM backend. A persistent search will not get notified if the database change is within the scope of the search. Also, no replication of the change to replica servers for this backend is performed if this server is the database owner in the cross-system group.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: For a TDBM database, restart the LDAP server to re-synchronize the TDBM backend with the other servers in the cross-system group. If the problem persists, contact the service representative.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD3321E The DIR_REPENTRY.CHNGDN column is smaller than the DIR_ENTRY.DN column.

Explanation: The LDAP server or utility found that a column in a DB2 table is too small. The size of the CHNGDN column in the DIR_REPENTRY table must be at least as large as the size of the DN column in the DIR_ENTRY table in the DB2 database for this backend.

Example: None.

System action: The backend does not start. If the srvStartUpError option in the LDAP server configuration file is set to ignore, the LDAP server continues to run with those backends that successfully start. If the srvStartUpError option is set to terminate (this is the default if the configuration option is not specified), the program ends. The utility ends regardless of the option value.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Increase the size of the CHNGDN column in the DIR_REPENTRY table or decrease the size of the DN column in the DIR_ENTRY table so that the DN column size is not greater than the CHNGDN column size. Then restart the program if it ended or if the backend is needed.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD3322E The attribute attribute for replication entry 'name' is not valid.

Explanation: A replica entry contains an attribute whose value is not supported. Replication is not performed for the replica identified by this entry until the value is corrected.

In the message text:

- attribute
  Attribute type
- name
  Replica entry distinguished name

Example: None.

System action: The LDAP server continues. Directory updates are not replicated to the replica identified by this entry.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Modify the attribute value in the replica entry to correct the error.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD3323E Unable to add 'name' to the replication list.

Explanation: The LDAP server is unable to add the indicated replica entry to the replica list. Replication is not performed for the replica identified by this entry until the error is corrected. A previous message may indicate the cause of the failure.
In the message text:

name
  Replica entry distinguished name

Example: None.

System action: The LDAP server continues. Directory updates are not replicated to the replica identified by this entry.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the information in the earlier message to correct the error. If there is no earlier message, use the LDAP server DEBUG operator modify command to turn on the ERROR debug level and then re-issue the operation. The output may assist in locating and correcting the problem.

Problem determination: Not applicable.

Source: LDAP

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD3324E Unable to wait for network event:

explanation: The LDAP server is unable to wait for a network event. Refer to the description of selectex() in z/OS XL C/C++ Run-Time Library Reference for more information on the error.

In the message text:

error_code
  Error code from selectex()

reason_code
  Reason code from selectex()

erro_text
  Error text corresponding to the error code

Example: None.

System action: The LDAP server continues, however TDBM replication is not available.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the information in the message to correct the error. Restart the program if replication is needed. If the problem persists, contact the service representative.

Problem determination: Not applicable.

Source: LDAP

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD3325E Replication entry 'name' requires SSL but SSL support is not configured.

Explanation: The replicaUseSSL attribute is set to TRUE in the replica entry but SSL support is not configured in the LDAP server configuration file. Replication will not be performed for the replica identified by this entry until the error is corrected.

In the message text:

name
  Replica entry distinguished name

Example: None.

System action: The LDAP server continues. Directory updates are not replicated to the replica identified by this entry.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Either configure SSL support for the LDAP server in the LDAP server configuration file and then restart the LDAP server or modify the replica entry to specify FALSE for the replicaUseSSL attribute.

Problem determination: Not applicable.

Source: LDAP

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD3326E Replication failed with host:port: Error error_code - error_text.

Explanation: The LDAP server is unable to replicate a directory modification to the indicated replica.

In the message text:

host
  Replica server host name

port
  Replica server port number

error_code
  Error code
**error_text**

Error text corresponding to the error code

**Example:** None.

**System action:** The LDAP server continues. The failing replication request is periodically retried until replication is successful.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Use the information in the message to correct the error. The associated replica entry should be deleted from the directory if replication is no longer desired for the failing replica. Deleting and then adding the replica entry will reset the replication status so that only future directory modifications will be replicated to the replica.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD3327E** Additional information: text

**Explanation:** This message provides additional information for a replication error. The text is the error message returned by the replica server.

In the message text:

- **text**
  - Additional text

**Example:** None.

**System action:** The LDAP server continues.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Use the information in the message to assist in correcting the error.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD3328E** Additional information: text

**Explanation:** This message provides additional information for a replication error. The text is the error message returned by the replica server.

In the message text:

- **text**
  - Additional text

**Example:** None.

**System action:** The LDAP server continues.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** None.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD3329E** Group owner for type backend named name cannot be contacted.

**Explanation:** The LDAP server is unable to contact the LDAP server that owns the indicated database in the LDAP cross-system group in the sysplex.

In the message text:

- **type**
  - Backend type
- **name**
  - Backend name

**Example:** None.

**System action:** The LDAP server continues. Changes to the backend database may not be replicated to the replica servers for the backend.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Determine the owning system by issuing the LDAP server DISPLAY XCF.
operator modify command for the LDAP server reporting the error. The command output will indicate which LDAP server is the group owner. Then issue the \texttt{DISPLAY XCF} operator modify command for the owning LDAP server and verify that this server is really the group owner. Restart the owning LDAP server if there is no response to the \texttt{DISPLAY XCF} command. If the problem persists, contact the service representative.

\textbf{Problem determination:} Not applicable.

\textbf{Source:} LDAP

\textbf{Module:} None.

\textbf{Routing code:} None.

\textbf{Descriptor code:} None.

\textbf{Automation:} Not applicable.

\begin{verbatim}
GLD3330E Unable to decrypt change request: 
Error_text.

Explanation: The LDAP server encrypts pending replication requests when the request contains the \texttt{userPassword} or \texttt{secretKey} attribute and the \texttt{secretEncryption} option is specified in the LDAP server configuration file. The request must then be decrypted before it is sent to a replica server. This error indicates the LDAP server is unable to decrypt the request.

In the message text:

\begin{verbatim}
Error_text
\end{verbatim}

\textbf{Example:} None.

\textbf{System action:} The LDAP server continues, however TDBM replication may stall because this replication request cannot be processed.

\textbf{Operator response:} None.

\textbf{User response:} None.

\textbf{System programmer response:} None.

\textbf{Administrator response:} Verify that the encryption key label specified in the \texttt{secretEncryption} option in the LDAP server configuration file has not been changed. If it has been changed, ensure that the previous encryption key label is still defined.

\textbf{Problem determination:} Not applicable.

\textbf{Source:} LDAP

\textbf{Module:} None.

\textbf{Routing code:} None.

\textbf{Descriptor code:} None.

\textbf{Automation:} Not applicable.
\end{verbatim}

\begin{verbatim}
GLD3331E Unable to decrypt replica credentials: 
error_text.

Explanation: The LDAP server encrypts the password specified by the \texttt{replicaCredentials} attribute in a replica entry if the \texttt{secretEncryption} option is specified in the LDAP server configuration file. The password must then be decrypted before the LDAP server can bind to the replica server. This error indicates the LDAP server is unable to decrypt the replica password.

In the message text:

\begin{verbatim}
error_text
\end{verbatim}

\textbf{Example:} None.

\textbf{System action:} The LDAP server continues.

\textbf{Operator response:} None.

\textbf{User response:} None.

\textbf{System programmer response:} None.

\textbf{Administrator response:} Verify that the encryption key label specified in the \texttt{secretEncryption} option in the LDAP server configuration file has not been changed. If it has been changed, ensure that the previous encryption key label is still defined.

\textbf{Problem determination:} Not applicable.

\textbf{Source:} LDAP

\textbf{Module:} None.

\textbf{Routing code:} None.

\textbf{Descriptor code:} None.

\textbf{Automation:} Not applicable.
\end{verbatim}

\begin{verbatim}
GLD3332I type backend named \texttt{name} schema migration has started.

Explanation: A TDBM backend was initially created by an Integrated Security Services LDAP server and the schema used for the TDBM backend is contained in an entry in the TDBM backend. In IBM Tivoli Directory Server for z/OS, there is a single LDAP server schema for all backends. The LDAP server has begun to merge the TDBM backend schema into the LDAP server schema.

In the message text:

\begin{verbatim}
type
  Backend type
name
  Backend name
\end{verbatim}

\textbf{Example:} None.

\textbf{System action:} The LDAP server continues.

\textbf{Operator response:} None.
\end{verbatim}
GLD3333I  type backend named name schema migration has ended.

Explanation: A TDBM backend was initially created by an Integrated Security Services LDAP server and the schema used for the TDBM backend is contained in an entry in the TDBM backend. In IBM Tivoli Directory Server for z/OS, there is a single LDAP server schema for all backends. The LDAP server has completed merging the TDBM backend schema into the LDAP server schema.

In the message text:

type
  Backend type
name
  Backend name

Example: None.

System action: The LDAP server continues.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD3334E  Schema migration has failed. error_text.

Explanation: A TDBM backend was initially created by an Integrated Security Services LDAP server and the schema used for the TDBM backend is contained in an entry in the TDBM backend. In IBM Tivoli Directory Server for z/OS, there is a single LDAP server schema for all backends. An error occurred while the LDAP server was merging the TDBM backend schema into the LDAP server schema.

In the message text:

error_text
  Error text

Example: None.

System action: The TDBM backend does not start. If the srvStartUpError option in the LDAP server configuration file is set to ignore, the LDAP server continues to run with those backends that successfully start. If the srvStartUpError option is set to terminate (this is the default if the configuration option is not specified), the program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the information in the message to correct the error. Then restart the program if it stopped or if the backend is needed.

Problem determination: Not applicable.

Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD3335I  The option option is not supported in multi-server mode with DB_VERSION less than 4. The option is ignored.

Explanation: The option indicated in the message is specified in the LDAP server configuration file. However, the database for this backend is running in multi-server mode and has a DB_VERSION less than 4. This indicates that the TDBM database is being shared with an earlier version of the LDAP server. The configuration option is not supported in this environment.

In the message text:

option
  LDAP server configuration option

Example: None.

System action: The LDAP server continues. The option is ignored.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Remove the option from the LDAP server configuration file.
Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD3336E Unable to decode a type entry (name) for persistent search, rc=code.

Explanation: The LDAP server is unable to decode an entry passed in a notification from another LDAP server in the sysplex group.

In the message text:

<table>
<thead>
<tr>
<th>type</th>
<th>Backend type</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Entry distinguished name</td>
</tr>
<tr>
<td>code</td>
<td>LDAP return code</td>
</tr>
</tbody>
</table>

Example: None.

System action: The LDAP server continues. Any persistent searches will not receive this notification and thus the entry will not be returned if it would have matched the persistent search.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: To disable persistent search on the server named in the message, set the persistentSearch option to off in the LDAP server configuration file on that server. To enable persistent search on this server, set the persistentSearch option to on in the LDAP server configuration file on this server. The LDAP server containing the configuration file that is changed must be restarted to put the change into effect.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD3337E Backend backend_name on LDAP server server_name in the sysplex group has different persistent search settings than this server.

Explanation: This LDAP server does not have persistent search enabled but another server in the sysplex group does have persistent search enabled. All LDAP servers in a sysplex group must have the same persistent search settings.

In the message text:

backend_name
  Backend name

server_name
  LDAP server name

Example: None.

System action: The LDAP server continues.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: To disable persistent search on the server named in the message, set the persistentSearch option to off in the LDAP server configuration file on that server. To enable persistent search on this server, set the persistentSearch option to on in the LDAP server configuration file on this server. The LDAP server containing the configuration file that is changed must be restarted to put the change into effect.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD3338E The option option is not supported in multi-server mode with DB_VERSION less than 4.

Explanation: The option indicated in the message is specified in the LDAP server configuration file. However, the TDBM database for this backend is running in multi-server mode and has a DB_VERSION less than 4. This indicates that the TDBM database is being shared with an earlier version of the LDAP server. The configuration option or its value is not supported in this environment. If the option is:

- pwEncryption - AES encryption cannot be used. DES encryption can be used if the DES keys are stored in ICSF. All other encryption methods are supported.
- secretEncryption - cannot be used. Even when this option is not specified, there can be problems using the secretKey and replicaCredentials attributes in this environment. See the migration documentation for more information.
- pwCryptCompat - cannot be set to no. The earlier LDAP server only supports the EBCDIC version of the crypt() algorithm.

In the message text:
option

LDAP server configuration option

Example: None.

System action: The backend does not start. If the srvStartUpError option in the LDAP server configuration file is set to ignore, the LDAP server continues to run with those backends that successfully start. If the srvStartUpError option is set to terminate (this is the default if the configuration option is not specified), the program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Remove the option or change its value in the LDAP server configuration file. Restart the program if it ended or if the backend is needed.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

---

GLD3339W No DB2 statistics gathered on table 'owner.table'.

Explanation: The LDAP server did not find statistics for the specified TDBM table. This may be an indication that the DB2 RUNSTATS utility has not been successfully run.

In the message text:

owner
Database owner
table
Database table

Example: None.

System action: The LDAP server continues, however database queries may not have optimal performance.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Verify that the DB2 RUNSTATS utility has been successfully run. Refer to the "Performance tuning" chapter of the IBM Tivoli Directory Server Administration and Use for z/OS for more information on running the RUNSTATS utility for the LDAP server.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

---

GLD3340I Found num_values frequent values for column column of table owner.table.
Table cardinality is cardinality and least frequent value has a frequency of frequency.

Explanation: The LDAP server has successfully found RUNSTATS information in the DB2 catalog for the indicated TDBM database. The number of frequent values found is governed by the options given to the RUNSTATS utility. The table cardinality indicates the number of rows in the table. The frequency of the least frequent value indicates how many times that value appears in the table. If the frequency is a large percentage of the table cardinality, it may indicate that there are more frequent values to be gathered. In this case the DB2 RUNSTATS utility can be re-run, with updated options to gather more frequent values for the column indicated above. Refer to the "Performance tuning" chapter of IBM Tivoli Directory Server Administration and Use for z/OS for more information on running the RUNSTATS utility for the LDAP server.

In the message text:

num_values
Number of frequent values
column
Database column
owner
Database owner
table
Database table
cardinality
Database table cardinality
frequency
Lowest frequency

Example: None.

System action: The LDAP server continues.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None. The information in this message may be useful to tailoring the use of the RUNSTATS utility for improved performance.

Problem determination: Not applicable.

Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD3341W   Insufficient row statistics gathered on column column of table owner.table.

Explanation: The LDAP server did not find any row statistics for the specified columns in the DB2 Catalog. This may indicate that the DB2 RUNSTATS utility was run without the options suggested to gather statistics for the indicated table and columns.

In the message text:

column Database column
owner Database owner
table Database table

Example: None.

System action: The LDAP server continues, however database queries may not have optimal performance.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD3342I   type backend named name is enabled for partition based key assignment.

Explanation: The specified backend of the LDAP server or utility will assign keys to new directory entries using the partition based key assignment algorithm.

In the message text:

type Backend type
name Backend name

Example: None.

System action: None.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD3343E   The partitioning information stored in type backend named name is no longer consistent with the information stored in the DB2 database for this backend.

Explanation: The partitioning information of the DIR_SEARCH tablespace stored in memory by the specified backend is no longer consistent with the information stored in the DIR_EID table by that backend. The user must have re-partitioned the DIR_SEARCH tablespace and then started another LDAP server that is sharing the same DB2 database.

In the message text:

type Backend type
name Backend name

Example: None.

System action: The program continues. The request fails. Future add requests to this backend are not guaranteed to succeed until the LDAP server restarts.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Restarting the LDAP server will update the partition control blocks for the specified backend, making them consistent with the information stored in the DIR_EID table. If the DIR_SEARCH tablespace has indeed been repartitioned by the user, it is recommended to unload the data from the specified backend using ds2ldif, recreate the database belonging to this backend, and reload the data back using ldif2ds. If the problem persists, contact the service representative.

Chapter 30. LDAP server messages  649
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD3344E Unable to add new entries to the DB2 database in type backend named name because all unique keys have been exhausted.

Explanation: The DB2 database in the specified backend is unable to accept new entries because there are no unique keys available.

In the message text:
- type Backend type
- name Backend name

Example: None.

System action: The program continues, but the request fails. Future add requests to this backend will not succeed.

Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Restart the program. If the problem persists, contact the service representative.

Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD3345E Unable to process the DB2 database belonging to type backend named name: error code error_code.

Explanation: The LDAP server or utility is unable to process the DB2 database belonging to the specified backend. The error code has the following values:

1 The DIR_EID table is required by the partitioned entry identifier assignment algorithm, but it is not defined in the DB2 database belonging to the specified backend. This scenario should never happen. If it does happen, this is most likely the result of someone dropping the DIR_EID table from the DB2 database manually. Important entry identifier information will be lost if the DIR_EID table is dropped.

2 The value detected for the PARTITIONED_EID column of the DIR_MISC table in the DB2 database belonging to the specified backend is not valid. The PARTITIONED_EID column, besides being a NULL column, only allows values 'T' or 'F'.

3 The number of entry identifiers that have being assigned and recorded in the DB2 database belonging to the specified backend has exceeded the maximum number of entry identifiers allowed by the LDAP server.

In the message text:
- type Backend type
- name Backend name
- error_code Error code

Example: None.

System action: The specified backend does not start. If the srvStartUpError option in the LDAP server configuration file is set to ignore, the LDAP server continues to run with those backends that successfully start. If the srvStartUpError option is set to terminate (this is the default if the configuration option is not specified), the program ends. The utility ends regardless of the option value.

Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Use the information in the message to correct the problem. For error codes 1 and 3, consider unloading the data from the specified backend using ds2ldif, rebuilding the DB2 database belonging to this backend, and reloading the data back using ldif2ds. Restart the program if it didn't start or if the backend is needed. If the problem persists, contact the service representative.

Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.
GLD3346E Unable to initialize the entry identifier assignment algorithm for type backend named name.

Explanation: The LDAP server or utility is unable to initialize the entry identifier assignment algorithm. A previous message indicates the reason for the failure.

In the message text:
- type Backend type
- name Backend name

Example: None.

System action: The specified backend does not start.

If the srvStartUpError option in the LDAP server configuration file is set to ignore, the LDAP server continues to run with those backends that successfully start. If the srvStartUpError option is set to terminate (this is the default if the configuration option is not specified), the program ends. The utility ends regardless of the option value.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the information in the earlier message to correct the problem. Restart the program if it didn’t start or if the backend is needed.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD3347E Group owner for type backend named name is busy, retrying.

Explanation: A request was sent from this LDAP server to the LDAP server that owns the indicated database in the LDAP cross-system group in the sysplex. The database owner is either too busy to respond to the request or it is waiting for a DB2 connection.

In the message text:
- type Backend type
- name Backend name

Example: None.

System action: The LDAP server continues and retries the request.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Determine the owning system by issuing the LDAP server DISPLAY XCF operator modify command for the LDAP server reporting the error. The command output will indicate which LDAP server is the group owner. Then issue the DISPLAY XCF operator modify command for the owning LDAP server and verify that this server is really the group owner. Either fix the DB2 connection problem on the owning server or restart the owning LDAP server if there is no response to the DISPLAY XCF command. If the problem persists, contact the service representative.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD3350E The name backend requires that the serverCompatLevel (value) match the database version value (major).

Explanation: The TDBM backend has detected that it is a sysplex replica server and that the serverCompatLevel value in the server configuration file for this server does not match the database version for the backend identified in the message.

In the message text:
- name Backend name
- value serverCompatLevel option value
- major database version number

Example: None.

System action: The specified backend does not start.

If the srvStartUpError option in the LDAP server configuration file is set to ignore, the LDAP server continues to run with those backends that successfully start. If the srvStartUpError option is set to terminate (this is the default if the configuration option is not specified), the program ends.

Operator response: None.

User response: None.

System programmer response: None.
Administrator response: Verify the serverCompatLevel configuration option value is the same for all servers sharing the backend. Then restart the program.

Problem determination: Not applicable.

Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD3351E The 'type' backend named 'name' requires an EBCDIC encoding scheme. Encoding Scheme 'codeset' was found for table 'owner.table'.

Explanation: The LDAP server has detected that a DB2 database table for a TDBM or GDBM backend was created with a non EBCDIC encoding scheme. The LDAP server only supports DB2 database tables that are created with an EBCDIC encoding scheme.

In the message text:

- type Backend type
- name Backend name
- codeset Encoding scheme code set

LDBM messages (6000)

GLD6001E Insufficient storage available for database control block.

Explanation: The LDAP server or utility is unable to allocate storage.

Example: None.

System action: Depending on where the error occurs, the LDAP server may end. If the server does not end, the LDBM backend may not start or some LDBM functionality may not be available.

Operator response: Increase the storage available for use by the LDAP server or utility and restart the program. If the problem persists, contact the service representative.

User response: None.

System programmer response: None.

Problem determination: Not applicable.

Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD6004E Unable to get status for database file filename: error_code/reason_code - error_text

Explanation: The LDAP server is unable to get status information on the indicated database file. Refer to the description of fstat() in z/OS XL C/C++ Run-Time Library Reference for more information on the error.

In the message text:

- filename Database file name
- error_code Error code from fstat()
**reason_code**
Reason code from `fstat()`

**error_text**
Error text corresponding to the `error_code`

**Example:** None.

**System action:**
- If the error occurs during backend initialization, the backend does not start. If the `srvStartUpError` option in the LDAP server configuration file is set to `ignore`, the LDAP server continues to run with those backends that successfully start. If the `srvStartUpError` option is set to `terminate` (this is the default if the configuration option is not specified), the program ends. The utility ends regardless of the option value.
- If the error occurs when the database is being reloaded because a moddn operation failed, then the database is marked as disabled. The LDAP server continues, but requests to the affected backend fail.
- Otherwise, the `fileTerminate` option in the backend section of the LDAP server configuration determines what the server does. If the `fileTerminate` option is set to `terminate`, the program ends. If the `fileTerminate` option is set to `recover` (this is the default if the configuration option is not specified), the LDAP server continues processing, but the backend is set to read-only mode.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Verify that the file exists and can be accessed by the LDAP server. Restart the LDAP server if it ended or if the backend is needed. To change the backend to read-write mode, use the LDAP server `BACKEND` operator modify command. Then retry the request.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

**GLD6005E**

**Explanation:** The LDAP server is unable to load the database for a backend and has put the backend in disabled state. A previous message indicates the reason for the failure.

In the message text:

**type**
Backend type

**name**
Backend name

**Example:** None.

**System action:** The LDAP server continues, however the backend can no longer process requests.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Use the information in the earlier message to correct the error. Restart the LDAP server if the backend is needed. If the problem persists, contact the service representative.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.
the backend section in the configuration file and restart the program.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

GLD6007E Unable to load type backend named name because object class objectclass is not defined.

**Explanation:** The LDAP server or utility found an object class used by an entry in the indicated backend is not defined in the LDAP server schema.

In the message text:

- type
  - Backend type
- name
  - Backend name
- objectclass
  - Undefined object class

**Example:** None.

**System action:**

- If the error occurs during backend initialization, the backend does not start. If the `srvStartUpError` option in the LDAP server configuration file is set to `ignore`, the LDAP server continues to run with those backends that successfully start. If the `srvStartUpError` option is set to `terminate` (this is the default if the configuration option is not specified), the program ends. The utility ends regardless of the option value.

- If the error occurs when the database is being reloaded because a moddn operation failed, then the `fileTerminate` option in the backend configuration determines what the server does. If the `fileTerminate` option is set to `terminate`, the program ends. If the `fileTerminate` option is set to `recover` (this is the default if the configuration option is not specified), the LDAP server continues processing, but the backend is marked as disabled and requests to the affected backend fail.

- Otherwise, the `fileTerminate` option in the backend section of the LDAP server configuration determines what the server does. If the `fileTerminate` option is set to `terminate`, the program ends. If the `fileTerminate` option is set to `recover` (this is the default if the configuration option is not specified), the LDAP server continues processing, but the backend is set to read-only mode.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** If the backend is needed, restore the indicated database file from a backup. Then restart the program. If you need to determine which entry in the database file is not valid, restart the LDAP server with -d ERROR specified on the command line.

**Problem determination:** Not applicable.

---

GLD6008E Unable to load type backend named name because database file filename is not valid.

**Explanation:** The LDAP server or utility is unable to decode an entry in the indicated database file. This indicates that the database file has been modified and is no longer usable.

In the message text:

- type
  - Backend type
- name
  - Backend name
- filename
  - Database file name

**Example:** None.

**System action:**

- If the error occurs during backend initialization, the backend does not start. If the `srvStartUpError` option in the LDAP server configuration file is set to `ignore`, the LDAP server continues to run with those backends that successfully start. If the `srvStartUpError` option is set to `terminate` (this is the default if the configuration option is not specified), the program ends. The utility ends regardless of the option value.

- If the error occurs when the database is being reloaded because a moddn operation failed, then the `fileTerminate` option in the backend section of the LDAP server configuration determines what the server does. If the `fileTerminate` option is set to `terminate`, the program ends. If the `fileTerminate` option is set to `recover` (this is the default if the configuration option is not specified), the LDAP server continues processing, but the backend is marked as disabled and requests to the affected backend fail.

- Otherwise, the `fileTerminate` option in the backend section of the LDAP server configuration determines what the server does. If the `fileTerminate` option is set to `terminate`, the program ends. If the `fileTerminate` option is set to `recover` (this is the default if the configuration option is not specified), the LDAP server continues processing, but the backend is set to read-only mode.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** If the backend is needed, restore the indicated database file from a backup. Then restart the program. If you need to determine which entry in the database file is not valid, restart the LDAP server with -d ERROR specified on the command line.

**Problem determination:** Not applicable.
GLD6009E Unable to open directory path: error_code/reason_code - error_text

Explanation: The LDAP server or utility is unable to open the indicated file directory. Refer to the description of opendir() in [z/OS XL C/C++ Run-Time Library Reference] for more information on the error.

In the message text:

path
LDBM database directory path
error_code
Error code from opendir()
reason_code
Reason code from opendir()
error_text
Error text corresponding to the error code

Example: None.

System action: If the error occurs during backend initialization, the backend does not start. If the srvStartUpError option in the LDAP server configuration file is set to ignore, the LDAP server continues to run with those backends that successfully start. If the srvStartUpError option is set to terminate (this is the default if the configuration option is not specified), the program ends. The utility ends regardless of the option value. If the error occurs during an attempt to become the sysplex group owner, the LDAP server ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the information in the message to correct the error. Verify that the directory exists and can be accessed by the LDAP server. Restart the program if it stopped or if the backend is needed.

Problem determination: Not applicable.

Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD6010E Unable to read directory path: error_code/reason_code - error_text

Explanation: The LDAP server or utility is unable to read the indicated file directory. Refer to the description of readdir_r() in [z/OS XL C/C++ Run-Time Library Reference] for more information on the error.

In the message text:

path
LDBM database directory path
error_code
Error code from readdir_r()
reason_code
Reason code from readdir_r()
error_text
Error text corresponding to the error code

Example: None.

System action: If the error occurs during backend initialization, the backend does not start. If the srvStartUpError option in the LDAP server configuration file is set to ignore, the LDAP server continues to run with those backends that successfully start. If the srvStartUpError option is set to terminate (this is the default if the configuration option is not specified), the program ends. The utility ends regardless of the option value. If the error occurs during an attempt to become the sysplex group owner, the LDAP server ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the information in the message to correct the error. Verify that the directory exists and can be accessed by the LDAP server. Restart the program if it stopped or if the backend is needed.

Problem determination: Not applicable.

Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD6011E Unable to delete database file filename: error_code/reason_code - error_text

Explanation: The LDAP server or utility is unable to delete the indicated database file. Refer to the description of remove() in [z/OS XL C/C++ Run-Time Library Reference] for more information on the error.

In the message text:
filename
   LDBM database file name

error_code
   Error code from remove()

reason_code
   Reason code from remove()

error_text
   Error text corresponding to the error code

Example: None.

System action:
- If the error occurs during backend initialization, the backend does not start. If the srvStartUpError option in the LDAP server configuration file is set to ignore, the LDAP server continues to run with those backends that successfully start. If the srvStartUpError option is set to terminate (this is the default if the configuration option is not specified), the program ends. The utility ends regardless of the option value.
- If the error occurs when the database is being reloaded because a moddn operation failed, then the database is marked as disabled. The LDAP server continues, but requests to the affected backend fail.
- Otherwise, the fileTerminate option in the backend section of the LDAP server configuration determines what the server does. If the fileTerminate option is set to terminate, the program ends. If the fileTerminate option is set to recover (this is the default if the configuration option is not specified), the LDAP server continues processing, but the backend is set to read-only mode.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the information in the message to correct the error. Verify that the file exists and can be accessed by the LDAP server. Restart the LDAP server if it ended or if the backend is needed. To change the backend to read-write mode, use the LDAP server BACKEND operator modify command. Then retry the request.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD6012E Unable to open database file filename:
   error_code/reason_code - error_text

Explanation: The LDAP server is unable to open the indicated database file. Refer to the description of open() in z/OS XL C/C++ Run-Time Library Reference for more information on the error.

In the message text:

filename
   Database file name
error_code
   Error code from open()
reason_code
   Reason code from open()
error_text
   Error text corresponding to the error code

Example: None.

System action:
- If the error occurs during backend initialization, the backend does not start. If the srvStartUpError option in the LDAP server configuration file is set to ignore, the LDAP server continues to run with those backends that successfully start. If the srvStartUpError option is set to terminate (this is the default if the configuration option is not specified), the program ends. The utility ends regardless of the option value.
- If the error occurs when the database is being reloaded because a moddn operation failed, then the database is marked as disabled. The LDAP server continues, but requests to the affected backend fail.
- Otherwise, the fileTerminate option in the backend section of the LDAP server configuration determines what the server does. If the fileTerminate option is set to terminate, the program ends. If the fileTerminate option is set to recover (this is the default if the configuration option is not specified), the LDAP server continues processing, but the backend is set to read-only mode.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the information in the message to correct the error. Verify that the file exists and can be accessed by the LDAP server. Restart the LDAP server if it ended or if the backend is needed. To change the backend to read-write mode, use the LDAP server BACKEND operator modify command. Then retry the request.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD6013E Unable to read database file filename:
   error_code/reason_code - error_text

Explanation: The LDAP server is unable to read the indicated database file. The error occurred in the read() or readv() routine. Refer to the description of these routines in z/OS XL C/C++ Run-Time Library Reference for more information on the error.

In the message text:

filename
   Database file name
error_code
   Error code from routine
reason_code
Reason code from routine

error_text
Error text corresponding to the error code

Example: None.

System action:
- If the error occurs during backend initialization, the backend does not start. If the `srvStartUpError` option in the LDAP server configuration file is set to `ignore`, the LDAP server continues to run with those backends that successfully start. If the `srvStartUpError` option is set to `terminate` (this is the default if the configuration option is not specified), the program ends. The utility ends regardless of the option value.
- If the error occurs when the database is being reloaded because a `moddn` operation failed, then the database is marked as disabled. The LDAP server continues, but requests to the affected backend fail.
- Otherwise, the `fileTerminate` option in the backend section of the LDAP server configuration determines what the server does. If the `fileTerminate` option is set to `terminate`, the program ends. If the `fileTerminate` option is set to `recover` (this is the default if the configuration option is not specified), the LDAP server continues processing, but the backend is set to read-only mode.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the information in the message to correct the error. Verify that the file can be accessed by the LDAP server. Restart the LDAP server if it ended or if the backend is needed. To change the backend to read-write mode, use the LDAP server `BACKEND` operator modify command. Then retry the request.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

---

GLD6014E Unable to write database file filename: error_code/reason_code - error_text

Explanation: The LDAP server is unable to write the indicated database file. The error occurred in the `write()`, `writev()`, or `close()` routine. Refer to the description of these routines in the `z/OS XL C/C++ Run-Time Library Reference` for more information on the error.
GLD6015E  type version number file_type files are not supported.

Explanation: The indicated file format is not supported by the current level of the LDAP server.

In the message text:

type
  Backend type

number
  File version number

file_type
  File type

Example: None.

System action: If the error occurs during initialization, the backend does not start. If the srvStartUpError option in the LDAP server configuration file is set to ignore, the LDAP server continues to run with those backends that successfully start. If the srvStartUpError option is set to terminate (this is the default if the configuration option is not specified), the program ends. The utility ends regardless of the option value. If the error occurs after initialization, the program continues but replication is not performed.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: If access to the entries using this suffix is needed, add a suffix option specifying this suffix to the backend section of the LDAP server configuration file. Then restart the program.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD6016I  Suffix suffix in database file filename is not configured and will be ignored.

Explanation: A backend directory contains a suffix entry that is not in the list of suffixes specified by the suffix options for this backend in the LDAP server configuration file.

In the message text:

suffix
  Database suffix distinguished name

filename
  Database file name

Example: None.

System action: Backend initialization continues, but directory entries under this suffix are not accessible.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD6020E  Suffix suffix is not configured but the checkpoint file is not empty.

Explanation: A suffix has been removed from the LDBM backend section of the LDAP server configuration file, but there are one or more checkpoint records to be processed for that suffix.

In the message text:

suffix
  Database suffix distinguished name

Example: None.

System action: The backend does not start. If the srvStartUpError option in the LDAP server configuration file is set to ignore, the LDAP server continues to run with those backends that successfully start. If the srvStartUpError option is set to terminate (this is the default if the configuration option is not specified), the program ends. The utility ends regardless of the option value.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Either restore the LDAP server to the level used to create the database file or remove the indicated backend from the LDAP server configuration file. Restart the LDAP server if it ended or if the backend is needed.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.
GLD6021W Group 'name' contains an incorrect memberURL attribute value.

Explanation: The dynamic group membership URL cannot be evaluated. The format of a dynamic group URL is ldap:///dn?scope?filter, where dn is the distinguished name of the base entry for the search, scope is the search scope, and filter is the search filter. The valid values for the search scope are base, one, and sub. All of the attribute types specified in the search filter must be defined in the LDAP server schema and each assertion value must conform to the matching rule for the associated attribute type. BINARY attribute types cannot be specified in a search filter.

In the message text:
name  Dynamic group distinguished name

Example: None.

System action: The LDAP server continues. The dynamic group is not used in determining group memberships.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Determine the owning system for this backend by issuing the LDAP server DISPLAY XCF operator modify command for the LDAP server reporting the error. The command output indicates which LDAP server is the group owner. Then issue DISPLAY XCF for the owning LDAP server and verify that this server is really the group owner. Restart the owning LDAP server if there is no response to the DISPLAY XCF command. If the problem persists, contact the service representative.

Problem determination: Not applicable.

Source: LDAP

GLD6023E Database owner for type backend named name cannot be contacted rc=return_code.

Explanation: The LDAP server is unable to obtain a current copy of the database indicated in the message from the LDAP server that owns the database in the cross-system group. The return code displayed in the message is either from the attempt to send an XCF message to the database owner or from the reply from the owner if it could not send back the database.

In the message text:
type  Backend type
name  Backend name

Example: None.

System action: The backend does not start. If the srvStartUpError option in the LDAP server configuration file is set to ignore, the LDAP server continues to run with those backends that successfully start. If the srvStartUpError option is set to terminate (this is the default if the configuration option is not specified), the program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Determine the owning system for this backend by issuing the LDAP server DISPLAY XCF operator modify command for the LDAP server reporting the error. The command output indicates which LDAP server is the group owner. Then issue DISPLAY XCF for the owning LDAP server and verify that this server is really the group owner. Restart the owning LDAP server if there is no response to the DISPLAY XCF command. If the problem persists, contact the service representative.

Problem determination: Not applicable.

Source: LDAP

GLD6022E Time limit exceeded while loading type backend named name from group owner.

Explanation: The LDAP server waits a maximum of 60 seconds after requesting a copy of the indicated backend database from the LDAP server that owns the database in the cross-system group. This message indicates that the owning LDAP server is not responding to the request.

In the message text:
type  Backend type
name  Backend name

Example: None.
System action: If the error occurs during initialization, the backend does not start. If the `srvStartUpError` option in the LDAP server configuration file is set to `ignore`, the LDAP server continues to run with those backends that successfully start. If the `srvStartUpError` option is set to `terminate` (this is the default if the configuration option is not specified), the program ends. If the error occurs after initialization, the program continues. Update operations to the LDAP server probably fail. Search operations may succeed.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Determine the owning system for this backend by issuing the LDAP server `DISPLAY XCF` operator modify command for the LDAP server reporting the error. The command output indicates which LDAP server is the group owner. Then issue `DISPLAY XCF` for the owning LDAP server and verify that this server is really the group owner. Restart the owning LDAP server if there is no response to the `DISPLAY XCF` command. If the problem persists, contact the service representative.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD6024E type backend named name database XCF data record is not valid.

Explanation: A cross-system database record received by the indicated backend is not valid.

In the message text:

```
type
  Backend type

name
  Backend name
```

Example: None.

System action: If the error occurs during initialization, the backend does not start. If the `srvStartUpError` option in the LDAP server configuration file is set to `ignore`, the LDAP server continues to run with those backends that successfully start. If the `srvStartUpError` option is set to `terminate` (this is the default if the configuration option is not specified), the program ends. If the error occurs after initialization, the program continues but some operations may fail. A follow-on message indicates the effect on the program.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Ensure that the same suffixes are specified in the `suffix` option in the backend section of the LDAP server configuration file of each LDAP server in the cross-system group. Restart the program if it ended or if the backend is needed.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD6025E Suffix list for type backend named name does not match owner suffix list.

Explanation: The suffix list defined for the indicated backend in the LDAP server configuration file is not the same as the suffix list defined in the LDAP server configuration file of the LDAP server that owns the cross-system group resources.

In the message text:

```
type
  Backend type

name
  Backend name
```

Example: None.

System action: The backend does not start. If the `srvStartUpError` option in the LDAP server configuration file is set to `ignore`, the LDAP server continues to run with those backends that successfully start. If the `srvStartUpError` option is set to `terminate` (this is the default if the configuration option is not specified), the program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the information in the follow-on message to resolve the problem. Restart the program if it ended or if the backend is needed. If the problem persists, contact the service representative.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.
GLD6026E Unable to send type backend named name database update to group members.

Explanation: The LDAP server is unable to send a database update to the other members of the cross-system group. The indicated backend directory has been successfully updated on this LDAP server but the other members in the group will not have the updated database entry. A previous message indicates the reason for the failure.

In the message text:

- type
  Backend type
- name
  Backend name

Example: None.

System action: The LDAP server continues, but an LDAP operation for the indicated backend may return different results depending on if it is processed by this LDAP server or by another LDAP server in the cross-system group.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD6028E type directory path does not match group owner directory owner_path.

Explanation: The directory path specified by the databaseDirectory option in the backend section of the LDAP server configuration file is not correct. When multi-server mode is active, the directory path must be the same for this backend in each LDAP server in the cross-system group.

In the message text:

- type
  Backend type
- path
  Directory path in the LDAP server
- owner_path
  Directory path in the owning LDAP server

Example: None.

System action: The backend does not start. If the srvStartUpError option in the LDAP server configuration file is set to ignore, the LDAP server continues to run with those backends that successfully start. If the srvStartUpError option is set to terminate (this is the default if the configuration option is not specified), the program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Correct the databaseDirectory option in the backend section of the LDAP server configuration file to specify the same directory path for each LDAP server in the cross-system group. Restart the program if it ended or if the backend is needed.

Problem determination: Not applicable.

Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD6029E LDBM backend specified for a non-LDBM database.

Explanation: The LDAP server or utility found that the LDBM backend DLL, GLDBLD31 or GLDBLD64, is specified on a database option in the LDAP server configuration file but the type parameter on the option is not LDBM.

Example: None.

System action: The LDBM backend does not start. If the srvStartUpError option in the LDAP server configuration file is set to ignore, the LDAP server continues to run with those backends that successfully start. If the srvStartUpError option is set to terminate (this is the default if the configuration option is not specified), the program ends. The utility ends regardless of the option value.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Correct the database option in the LDAP server configuration file so that the DLL and backend type match. Restart the program if it ended or if the LDBM backend is needed.

Problem determination: Not applicable.
GLD6030E Unable to truncate database file

filename: error_code/reason_code - error_text

Explanation: The LDAP server is unable to truncate the indicated database file. Refer to the description of ftruncate() in z/OS XL C/C++ Run-Time Library for more information on the error.

In the message text:

filename
Database file name
error_code
Error code from ftruncate()
reason_code
Reason code from ftruncate()
error_text
Error text corresponding to the error code

Example: None.

System action: If the fileTerminate option in the backend section of the LDAP server configuration file is set to terminate, the program ends. If the fileTerminate option is set to recover (this is the default if the configuration option is not specified), the LDAP server continues processing, but the backend is set to read-only mode.

Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Correct the file system problem. Verify that the file can be accessed by the LDAP server. Then restart the program if it has ended or change the backend to read-write mode using the LDAP server BACKEND operator modify command.

Problem determination: Not applicable.

Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD6031E Setting backend named name to read-only because fileTerminate option is set to 'recover'.

Explanation: A write error is encountered while an LDAP server file-based backend is writing to the file system. Since the fileTerminate option in the backend section of the LDAP server configuration file is set to recover or the option is not specified at all, the LDAP server forces the backend directory into read-only mode.

In the message text:

name
Backend name

Example: None.

System action: The LDAP server continues to run. The backend contents cannot be modified.

Operator response: Verify that there is enough free space on the file system. Also verify that the LDAP server has read and write permissions to the database directory and files. A previous message indicates the reason for the failure.
User response: None.
System programmer response: None.
Administrator response: Use the information in the earlier message to correct the file system error. Then issue the LDAP server BACKEND operator modify command to change the backend to read-write mode.

Problem determination: Not applicable.

Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD6032A Terminating LDAP server because fileTerminate option is set to 'terminate' in backend named name.

Explanation: A write error is encountered while an LDAP server file-based backend is writing to the file system. Since the fileTerminate option in the backend section of the LDAP server configuration file is set to terminate, the program is ending. A previous message indicates the reason for the failure.

In the message text:

name
Backend name

Example: None.

System action: The program ends.
Operator response: Verify that there is enough free space on the file system. Correct the file system problem. Verify that the file can be accessed by the LDAP server. Then restart the program if it has ended or change the backend to read-write mode using the LDAP server BACKEND operator modify command.
space on the file system. Also verify that the LDAP server has read and write permissions on the database directory and files.

User response: None.
System programmer response: None.
Administrator response: Use the information in the earlier message to correct the file system error. Then restart the program.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD6033I Committing changes to database for type backend named name.

Explanation: The indicated LDAP server backend is going to commit changes to its database. This can occur periodically during normal processing or when the server is shutting down.

In the message text:

type
    Backend type
name
    Backend name

Example: None.
System action: The LDAP server continues.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD6034I Completed committing changes to database for type backend named name.

Explanation: The indicated LDAP server backend committed changes to its database. This can occur periodically during normal processing or when the server is shutting down.

In the message text:

type
    Backend type
name
    Backend name

Example: None.
System action: The LDAP server continues.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD6035E Unable to commit changes to database for type backend named name, rc=code.

Explanation: The indicated LDAP server backend failed to commit changes to its database. A previous message indicates the reason for the failure.

In the message text:

type
    Backend type
name
    Backend name
code
    LDAP return code

Example: None.
System action: If the fileTerminate option in the backend section of the LDAP server configuration file is set to terminate, the program ends. If the fileTerminate option is set to recover (this is the default if the configuration option is not specified), the LDAP server continues processing, but the backend is set to read-only mode.
Operator response: Verify that there is enough free space on the file system. Also verify that the LDAP server has read and write permissions on the database directory and files.
User response: None.
System programmer response: None.
Administrator response: Use the information in the earlier message to correct the file system error. Then restart the program if it has ended or change the
backend to read-write mode using the LDAP server
BACKEND operator modify command.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

GLD6036W type backend named name database owner is busy, retrying.

**Explanation:** A request was sent from this LDAP server to the LDAP server that owns the database in the cross-system group. The database owner is either too busy to respond to the request or it is waiting for a DB2 connection.

In the message text:

- **type**
  - backend type

- **name**
  - Backend name

**Example:** None.

**System action:** The LDAP server continues and retries the request.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Modify the replica entry to correct the error.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

GLD6037E Value value for attribute type in replication entry 'name' is not valid.

**Explanation:** The replica entry contains an attribute value that is not supported. Refer to the documentation on replication for more information about the attribute and its values.

In the message text:

- **value**
  - Attribute value

- **type**
  - Attribute type that contains value

- **name**
  - Replica entry distinguished name

**Example:** None.

**System action:** The LDAP server continues to run. Replication is not performed to the replica server identified by this entry.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** None.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.
**GLD6039E** Unable to wait for network event:

**error_code**

**reason_code**

**error_text**

**Explanation:** The LDAP server is unable to wait for a network event. Refer to the description of `selectex()` in the [z/OS XL C/C++ Run-Time Library Reference](#) for more information on the error.

In the message text:

- **error_code**
  - Error code from `selectex()`
- **reason_code**
  - Reason code from `selectex()`
- **error_text**
  - Error text corresponding to the error code

**Example:** None.

**System action:** The LDAP server continues, but replication is not performed.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Either configure SSL support in the LDAP server configuration file and then restart the LDAP server, or modify the value of the `replicaUseSSL` attribute to `FALSE`.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD6041E** Replication failed with host:port: Error

**error_code**

**error_text**

**Explanation:** The LDAP server is unable to replicate a directory modification to the indicated replica server.

In the message text:

- **host**
  - Replica server host name
- **port**
  - Replica server port number
- **error_code**
  - Error code
- **error_text**
  - Error text corresponding to the error code

**Example:** None.

**System action:** The LDAP server periodically retries the failing replication request until replication is successful.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:**

- If the replica server is down, then restart the replica server.
- If the replica credentials are not valid, then correct either the LDAP server configuration file on the replica server or the replica entry on this LDAP server. The distinguished name and password specified in the replica entry must match the values specified in the backend section of the LDAP server configuration file on the replica server. See the documentation on replication for more information on replica credentials.
- If the operation cannot take place because the replica server is missing updates, then resynchronize the

---

**GLD6040E** Replication entry 'name' requires SSL but SSL support is not configured.

**Explanation:** The indicated replica entry specifies TRUE for the `replicaUseSSL` attribute but SSL support is not configured in the LDAP server configuration file.

In the message text:

- **name**
  - Replica entry distinguished name

**Example:** None.

**System action:** The LDAP server continues.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.
replica server. See the documentation on replication for more information on how to recover from out-of-sync conditions.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD6042E** Unable to create LDAP handle for replication with `host:port`.

**Explanation:** The LDAP server is unable to create an LDAP handle for use with the indicated replica server. Either the `ldap_init()` or the `ldap_ssl_init()` routine failed.

In the message text:

- `host` Replica server host name
- `port` Replica server port number

**Example:** None.

**System action:** The LDAP server continues. Replication does not occur for the indicated replica server.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Use the information in the message to correct the error.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD6043E** Additional information: `text`

**Explanation:** This message provides additional information for a replication error. The text is the error message returned by the replica server.

In the message text:

- `text` Additional text

**Example:** None.

**System action:** The LDAP server continues.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** None.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD6044E** Unable to decrypt replica credentials: `text`.

**Explanation:** The LDAP server encrypts the password specified by the `replicaCredentials` attribute in a replica entry if the `secretEncryption` option is specified in the LDAP server configuration file. The password must then be decrypted before the LDAP server can bind to the replica server. The LDAP server is unable to decrypt the replica password.

In the message text:

- `text` Error text

**Example:** None.

**System action:** The LDAP server continues. Replication does not occur to this replica server. If the error occurs while adding or modifying a replica entry, the operation fails.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Verify that the encryption key label specified in the `secretEncryption` option in the LDAP server configuration file has not been changed. If it has been changed, ensure that the previous encryption key label is still defined.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.
GLD6045E  Replica object name does not have corresponding progress table entry.

Explanation: The LDAP server has found a replica entry for which there is no replication progress information in the replica progress file.

In the message text:

name
    Replica entry distinguished name

Example: None.

System action: The LDAP server continues. Replication does not occur to the replica server identified by this replica entry.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Resynchronize the replica server. See the documentation on replication for more information on how to recover from out-of-sync conditions.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

---

GLD6046E  Progress table entry uuid does not have corresponding Replica object.

Explanation: The LDAP server detected data for a replica in the progress file that does not have a corresponding replica entry in the directory for this backend. The ibm-entryUUID of each replica entry is kept in the progress file to associate the progress data with the replica to which it pertains.

In the message text:

uuid
    Replica entry ibm-entryUUID

Example: None.

System action: The LDAP server continues. The data for the replica is deleted from the progress file and replication to that replica server does not occur.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

---

GLD6047E  Unexpected state of replica type file, attempting recovery.

Explanation: The LDBM backend maintains several types of files containing replication information. When one of these files is changed, the current version of the file is renamed and eventually deleted when the new version of the file is created. The backend has found an old version of the file when it is not expected.

In the message text:

type
    Type of replica file

Example: None.

System action: The LDAP server tries to correct the error and continues.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

---

GLD6048E  Partial record read from file filename length=amount.

Explanation: The LDAP server is not able to read an entire record from the replication operations progress file.

In the message text:

filename
    Replication operations progress file

amount
    The amount of the record that was read

Example: None.
**System action:** If the error occurs during backend initialization, the backend does not start. If the `srvStartUpError` option in the LDAP server configuration file is set to `ignore`, the LDAP server continues to run with those backends that successfully start. If the `srvStartUpError` option is set to `terminate` (this is the default if the configuration option is not specified), the program ends. If the error occurs during an attempt to become the sysplex group owner, the LDAP server ends.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** If the backend is needed with replication, the replica servers for this backend need to be resynchronized. See the documentation on replication for more information on how to recover from out-of-sync conditions. You may have to first delete the replication queue file. If replication is not needed, then stop the LDAP server if it is running, delete the replication queue file, and restart the server.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

**Explanation:** The replication progress file is not present when the replication queue file is present. Both files are needed for replication processing.

**In the message text:**

```
filename
```

**Example:** None.

**System action:** The LDAP server continues.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** None.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

**Explanation:** The replication progress file is not present when the replication queue file is present. Both files are needed for replication processing.

**In the message text:**

```
filename
```

**Example:** None.

**System action:** If the error occurs during backend initialization, the backend does not start. If the `srvStartUpError` option in the LDAP server configuration file is set to `ignore`, the LDAP server continues to run with those backends that successfully start. If the `srvStartUpError` option is set to `terminate` (this is the default if the configuration option is not specified), the program ends. If the error occurs during an attempt to become the sysplex group owner, the LDAP server ends.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** If the backend is needed with replication, the replica servers for this backend need to be resynchronized. See the documentation on replication for more information on how to recover from out-of-sync conditions. You may have to first delete the replication queue file. If replication is not needed, then stop the LDAP server if it is running, delete the replication queue file, and restart the server.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

**Explanation:** The indicated backend has no changes to commit.

**In the message text:**

```
type
name
```

**Example:** None.

**System action:** The LDAP server continues.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** None.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

**Explanation:** The `ds2ldif` utility found a .new or .old version of the LDBM database file indicated in the message, but not the .db version of the file. This indicates that part of the directory in the backend to be unloaded may be missing. The unload cannot proceed. The .new and .old files are temporary versions of the database file created during checkpoint replay processing, when the database is updated using the...
contents of the checkpoint file. The temporary files are removed when the final updated .db database file is created. For some reason, this process must not have completed the last time it occurred.

In the message text:

**name**

Backend name

**file**

Database file name

**Example:** None.

**System action:** The program ends.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Verify that the LDAP server has write access to the file directory specified by the **databaseDirectory** option located in the named LDBM backend section of the LDAP server configuration file. Start the LDAP server, which will attempt to fix the problems in the database files. Then restart **ds2dif** using the -r option to force **ds2dif** to use an **unloadRequest** extended operation to unload the requested entries.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GDBM, advanced replication, and ldapdiff messages (8000)**

<table>
<thead>
<tr>
<th>GLD8001E</th>
<th>Unable to load the GDBM database because attribute type 'attribute' is not defined.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong> An attribute type used by an entry in the GDBM directory is not defined in the LDAP server schema.</td>
<td></td>
</tr>
<tr>
<td>In the message text:</td>
<td></td>
</tr>
</tbody>
</table>
| **attribute**
| Undefined attribute type |
| **Example:** None. |
| **System action:** The GDBM backend does not start. If the **srvStartUpError** option in the LDAP server configuration file is set to **ignore**, the LDAP server continues to run with those backends that successfully start. If the **srvStartUpError** option is set to **terminate** (this is the default if the configuration option is not specified), the program ends. |
| **Operator response:** None. |
| **User response:** None. |
| **System programmer response:** None. |
| **Administrator response:** If the GDBM backend is needed, restart the LDAP server without the GDBM backend section in the LDAP server configuration file and add the missing attribute type to the LDAP server schema. Then restore the GDBM backend section in the configuration file and restart the LDAP server. |
| **Problem determination:** Not applicable. |
| **Source:** LDAP |
| **Module:** None. |
| **Routing code:** None. |
| **Descriptor code:** None. |

<table>
<thead>
<tr>
<th>GLD8002E</th>
<th>Unable to load the GDBM database because object class 'objectclass' is not defined.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong> An object class used by an entry in the GDBM directory is not defined in the LDAP server schema.</td>
<td></td>
</tr>
<tr>
<td>In the message text:</td>
<td></td>
</tr>
</tbody>
</table>
| **objectclass**
| Undefined object class |
| **Example:** None. |
| **System action:** The GDBM backend does not start. If the **srvStartUpError** option in the LDAP server configuration file is set to **ignore**, the LDAP server continues to run with those backends that successfully start. If the **srvStartUpError** option is set to **terminate** (this is the default if the configuration option is not specified), the program ends. |
| **Operator response:** None. |
| **User response:** None. |
| **System programmer response:** None. |
| **Administrator response:** If the GDBM backend is needed, restart the LDAP server without the GDBM backend section in the LDAP server configuration file and add the missing object class to the LDAP server schema. Then restore the GDBM backend section in the configuration file and restart the LDAP server. |
| **Problem determination:** Not applicable. |
| **Source:** LDAP |
| **Module:** None. |
| **Routing code:** None. |
| **Descriptor code:** None. |
GLD8003E  GDBM backend specified for a non-GDBM database.

Explanation:  The GDBM backend DLL, GLDBGD31 or GLDBGD64, is specified on a database option in the LDAP server configuration file but the type parameter on the option is not GDBM.

Example:  None.

System action:  The GDBM backend does not start. If the srvStartUpError option in the LDAP server configuration file is set to ignore, the LDAP server continues to run with those backends that successfully start. If the srvStartUpError option is set to terminate (this is the default if the configuration option is not specified), the program ends.

Operator response:  None.

User response:  None.

System programmer response:  None.

Administrator response:  Correct the database option in the LDAP server configuration file so that the DLL and backend type match. Restart the LDAP server if it ended or if the GDBM backend is needed.

Problem determination:  Not applicable.

Source:  LDAP

Module:  None.

Routing code:  None.

Descriptor code:  None.

Automation:  Not applicable.

GLD8501E  Unable to connect to replica
"host_name" on port port_number. Verify that the replica is started.

Explanation:  The LDAP server is unable to establish a connection to the consumer server and port number specified in the message.

In the message text:

host_name
  LDAP host name

port_number
  LDAP port number

Example:  None.

System action:  The LDAP server continues however replication to the consumer server identified by the replication agreement is not started.

Operator response:  None.

User response:  None.

System programmer response:  None.

Administrator response:  Verify that the supplier server credentials entry distinguished name specified in the replication agreement entry is correct.

Problem determination:  Not applicable.

Source:  LDAP

Module:  None.

Routing code:  None.

Descriptor code:  None.

Automation:  Not applicable.

GLD8503W  The DN of the credential entry
"credential_name" defined for the replication agreement
"agreement_name" cannot be found.

Explanation:  The supplier server credentials entry which contains authentication information used to bind with the consumer server cannot be found. The supplier server credentials entry is specified in the ibm-replicaCredentialsDN attribute value in the replication agreement entry.

In the message text:

credential_name
  Credentials entry distinguished name

agreement_name
  Replication agreement entry distinguished name

Example:  None.

System action:  The LDAP server continues however replication to the consumer server identified by the replication agreement is not started.

Operator response:  None.

User response:  None.

System programmer response:  None.

Administrator response:  Verify that the supplier server credentials entry distinguished name specified in the replication agreement entry by the ibm-replicaCredentialsDN attribute value is correct and the entry exists. See Credentials entries for more information on the supplier server credentials entry.

Problem determination:  Not applicable.

Source:  LDAP

Module:  None.

Routing code:  None.

Descriptor code:  None.

Automation:  Not applicable.
GLD8504E  The credential entry 'credential_name' defined for the replication agreement 'agreement_name' is not valid.

Explanation:  The object class of the supplier server credentials entry defined in the ibm-replicationCredentialsDN attribute value for the replication agreement entry is not valid. The only supported object class values for supplier server credential entries are ibm-replicationCredentialsSimple and ibm-replicationCredentialsExternal.

In the message text:

credential_name
  Credentials entry distinguished name
agreement_name
  Replication agreement entry distinguished name

Example:  None.

System action:  The LDAP server continues however replication to the consumer server identified by the replication agreement is not started.

Operator response:  None.

User response:  None.

System programmer response:  None.

Administrator response:  Verify that the entry being deleted is not already referenced in the ibm-replicaCredentialsDN attribute value of any replication agreement entries. Either delete the replication agreement entry or modify the ibm-replicaCredentialsDN attribute value so that there are no longer any references to the entry that is being deleted. Then retry the delete client operation.

Problem determination:  Not applicable.

Source:  LDAP
Module:  None.
Routing code:  None.
Descriptor code:  None.
Automation:  Not applicable.

GLD8510E  The consumer URL defined in replication agreement entry 'agreement_name' is a duplicate.

Explanation:  In this replication context, the supplier server already has a replication agreement entry that has the same consumer server URL defined in the ibm-replicaURL attribute value. Within a replication context, each replication agreement entry must have a unique ibm-replicaURL attribute value. A supplier server is only allowed one connection to the same consumer server URL.

In the message text:

agreement_name
  Replication agreement entry distinguished name

Example:  None.

System action:  The LDAP server continues however the requested client add or modify operation is not successful.

Operator response:  None.

User response:  None.

System programmer response:  None.

Administrator response:  In the replication agreement entry that is being added or modified, verify that the consumer server URL in the ibm-replicaURL attribute value is correct and does not the same value as any existing replication agreement entry within this replication context. Then retry the requested client add or modify operation.

Problem determination:  Not applicable.

Source:  LDAP
Module:  None.
Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD8516E Internal processing error in server; replication thread cannot start.

Explanation: An internal error occurred while attempting to initialize the thread for advanced replication. A previously issued message specifies the replication agreement entry that had problems initializing.

Example: None.

System action: The LDAP server continues however advanced replication does not initialize.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD8518I Replication starting for replica 'agreement_name'.

Explanation: Replication to the consumer server identified by the replication agreement entry is initializing.

In the message text:

agreement_name

Replication agreement entry distinguished name

Example: None.

System action: The LDAP server continues.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD8519E Unable to create schedule for replica 'agreement_name'; all changes will be replicated immediately.

Explanation: An internal error occurred while attempting to initialize the replication schedule for the replication agreement entry specified in the message.

In the message text:

agreement_name

Replication agreement entry distinguished name

Example: None.

System action: The LDAP server continues however replication updates to the consumer server identified by the replication agreement are immediately replicated.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: This is an internal processing error that occurred while initializing the
replication scheduling support for the replication agreement entry. If the problem persists, contact the service representative.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD8520E** Unable to locate schedule entry with DN 'schedule_name'.

**Explanation:** The replication schedule entry which contains scheduling information for the replication agreement cannot be found. The replication schedule entry is specified in the ibm-replicaScheduleDN attribute value in the replication agreement entry which is specified in a previously issued message.

In the message text:

- **schedule_name**
  - Replication schedule entry distinguished name

**Example:** None.

**System action:** The LDAP server continues however replication updates to the consumer server identified by the replication agreement are immediately replicated.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Verify that the ibm-replicaScheduleDN attribute value is correct for the replication agreement entry and that the replication schedule entry exists. Either add the replication schedule entry to the directory or remove the ibm-replicaScheduleDN attribute value from the replication agreement entry. See [Schedule entries](#) for information on replication schedule entries.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD8521E** Error on schedule entry with DN 'schedule_name' attribute attribute_name value 'attribute_value'. Value ignored.

**Explanation:** The daily replication schedule entry does not have the correct time format for the attribute type and value specified in the message. The attribute type and value is ignored in the daily replication entry.

In the message text:

- **schedule_name**
  - Replication schedule entry distinguished name
- **attribute_name**
  - Attribute type
- **attribute_value**
  - Attribute value

**Example:** None.

**System action:** The LDAP server continues however the specified time in the daily replication schedule entry is ignored.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Verify that the time format for the daily replication schedule entry attribute type and value is the following: Thhmmss where time is based on a 24 hour clock. Modify the attribute value in the daily schedule replication entry specified in the message to have the correct time format. See [Schedule entries](#) for information on replication schedule entries.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD8522W** Replica 'agreement_name' missing schedule DN; all changes will be replicated immediately.

**Explanation:** The replication agreement entry does not have a replication schedule entry specified in the ibm-replicaScheduleDN attribute so the replication agreement defaults to replicating all updates immediately.

In the message text:

- **agreement_name**
  - Replication agreement entry distinguished name

**Example:** None.

**System action:** The LDAP server continues with replication updates to the consumer server identified by the replication agreement immediately occurring.

**Operator response:** None.

**User response:** None.
Administrator response: If the consumer server defined by the replication agreement entry should not immediately receive replication updates, modify the replication agreement entry to add an ibm-replicaScheduleDN attribute value. The ibm-replicaScheduleDN attribute value must be a weekly replication schedule entry with an object class value of ibm-replicationWeeklySchedule. See Schedule entries for information on replication schedule entries.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD8523E Error retrieving pending changes for replica 'agreement_name'. Will try again.

Explanation: An internal search error occurred while retrieving the pending replication changes from the backend where the replication agreement entry resides.

In the message text:

agreement_name
  Replication agreement entry distinguished name

changeID
  Replication change identifier

name
  Entry distinguished name

Example: None.

System action: The LDAP server continues however the pending replication changes cannot be retrieved from the backend where the replication agreement entry resides.

Operator response: None.

User response: None.

Administrator response: None.

System programmer response: None.

Administrator response: Verify that the backend where the replication agreement entry resides is functioning and handling requests correctly. If the problem persists, contact the service representative.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD8524E Error retrieving data for replica 'agreement_name' change ID changeID entry 'name'. Will try again.

Explanation: An internal search error occurred while retrieving the replication change ID from the backend replication table where the replication agreement entry resides.

In the message text:

agreement_name
  Replication agreement entry distinguished name
changeID
  Replication change identifier
name
  Entry distinguished name

Example: None.

System action: The LDAP server continues however the replication change ID cannot be retrieved from the backend replication table where the replication agreement entry resides.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Verify that the backend where the replication agreement entry resides is functioning and handling requests correctly. If the problem persists, contact the service representative.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD8525E Error while removing status entry for replica 'agreement_name'.

Explanation: An internal error occurred while deleting the replication agreement entry from the backend replication status table where the replication agreement entry resides.

In the message text:

agreement_name
  Replication agreement entry distinguished name

Example: None.

System action: The LDAP server continues however the replication agreement cannot be deleted from the backend replication status table.

Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Verify that the backend where the replication agreement entry resides is functioning and handling requests correctly. If the problem persists, contact the service representative.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD8526E Error while updating status for replica 'agreement_name' to last change ID changeID.

Explanation: An internal error occurred while updating the last change ID in the backend replication status table where the replication agreement entry resides.
In the message text:
agreement_name
Replication agreement entry distinguished name
changeID
Replication change identifier
Example: None.
System action: The LDAP server continues however the replication agreement status cannot be updated in the backend replication status table.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Verify that the backend where the replication agreement entry resides is functioning and handling requests correctly. If the problem persists, contact the service representative.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD8527W Error while parsing data for change ID changeID for replica 'agreement_name'. Will try again.

Explanation: An internal error occurred while parsing the replication change ID data from the backend replication table where the replication agreement entry resides.
In the message text:
changeID
Replication change identifier
agreement_name
Replication agreement entry distinguished name
Example: None.
System action: The LDAP server continues however the change ID cannot be retrieved from the backend replication table where the replication agreement entry resides.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Verify that the backend where the replication agreement entry resides is functioning and handling requests correctly. If the problem persists, contact the service representative.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD8528I Dropping connection to replica 'agreement_name' on host 'host_name' port port_number.

Explanation: The LDAP server is no longer connected to the consumer server and port number identified by the replication agreement entry in the ibm-replicaURL attribute value.
In the message text:
agreement_name
Replication agreement entry distinguished name
host_name
LDAP host name
port_number
LDAP port number
Example: None.
System action: The LDAP server continues however replication to the consumer server identified by the
replication agreement is no longer occurring.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Verify that the consumer server for the replication agreement entry is still running and that this server can still successfully connect to it.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

---

GLD8529I Established connection for replica 'agreement_name' on host 'host_name' port port_number.

Explanation: The LDAP server has successfully established a non-secure connection to the replica server and port number identified by the replication agreement entry. The replica server and port number are specified in LDAP URL format in the ibm-replicaURL attribute value of the replication agreement entry.

In the message text:

agreement_name
   Replication agreement entry distinguished name
host_name
   LDAP host name
port_number
   LDAP port number

Example: None.

System action: The LDAP server continues with replication to the consumer server identified by the replication agreement.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

---

GLD8530I Established secure connection for replica 'agreement_name' on host 'host_name' port port_number.

Explanation: The LDAP server has successfully established a secure connection to the replica server and port number identified by the replication agreement entry. The replica server and port number are specified in LDAP URL format in the ibm-replicaURL attribute value of the replication agreement entry.

In the message text:

agreement_name
   Replication agreement entry distinguished name
host_name
   LDAP host name
port_number
   LDAP port number

Example: None.

System action: The LDAP server continues with replication to the consumer server identified by the replication agreement.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

---

GLD8531I Replicating all pending changes for replica 'agreement_name'.

Explanation: The replication schedule entry identified by the ibm-replicaScheduleDN in the replication agreement entry indicates that all pending replication changes are now allowed to be replicated to the consumer server identified by the replication agreement.

In the message text:

agreement_name
   Replication agreement entry distinguished name

Example: None.

System action: The LDAP server continues with scheduled replication to the consumer server identified by the replication agreement.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.
GLD8532E  Error *error_text* occurred for replica
'agreement_name': add failed for entry 'name' change ID changeID.

Explanation: During replication from this supplier server to the consumer server defined in the replication agreement, an add operation failed. The error string shows the reason why the operation failed. The change ID is used to record the replication change in the backend where the replication agreement entry resides.

In the message text:

- *error_text*  
  Error message text

- *agreement_name*  
  Replication agreement entry distinguished name

- *name*  
  Entry distinguished name

- *changeID*  
  Replication change identifier

Example: None.

System action: The LDAP server continues however replication to the consumer server identified by the replication agreement may be stalled.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: See Monitoring and diagnosing advanced replication problems for information on recovering from advanced replication problems.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD8533E  Error *error_text* occurred for replica
'agreement_name': modify failed for entry 'name' change ID changeID.

Explanation: During replication from this supplier server to the consumer server defined in the replication agreement, a modify operation failed. The error string shows the reason why the operation failed. The change ID is used to record the replication change in the backend where the replication agreement entry resides.

In the message text:

- *error_text*  
  Error message text

- *agreement_name*  
  Replication agreement entry distinguished name

- *name*  
  Entry distinguished name

- *changeID*  
  Replication change identifier

Example: None.

System action: The LDAP server continues however replication to the consumer server identified by the replication agreement may be stalled.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: See Monitoring and diagnosing advanced replication problems for information on recovering from advanced replication problems.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD8534E  Error *error_text* occurred for replica
'agreement_name': rename failed for entry 'name' change ID changeID.

Explanation: During replication from this supplier server to the consumer server defined in the replication agreement, a rename operation failed. The error string shows the reason why the operation failed. The change ID is used to record the replication change in the backend where the replication agreement entry resides.

In the message text:

- *error_text*  
  Error message text
GLD8535E Error 'error_text' occurred for replica 'agreement_name': delete failed for entry 'name' change ID changeID.

Explanation: During replication from this supplier server to the consumer server defined in the replication agreement, a delete operation failed. The error string shows the reason why the operation failed. The change ID is used to record the replication change in the backend where the replication agreement entry resides.

In the message text:

error_text
   Error message text
agreement_name
   Replication agreement entry distinguished name
name
   Entry distinguished name
changeID
   Replication change identifier
Example: None.

System action: The LDAP server continues however replication to the consumer server identified by the replication agreement may be stalled.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: See [Monitoring and diagnosing advanced replication problems](#) for information on recovering from advanced replication problems.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.
the **ibm-slapdMasterPW** attribute value in the consumer server credentials entry used by the replication context. See [Consumer server entries](#) for more information about the consumer server credentials entry.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD8539E** Error '*error_text*' occurred for replica '*agreement_name*': EXTERNAL bind failed.

**Explanation:** The supplier server was unable to successfully perform a SASL EXTERNAL bind to the consumer server. The supplier server credentials entry specified by the **ibm-replicaCredentialsDN** attribute value in the replication agreement entry contains optionally attribute values for the SSL key database file, RACF keyring, PKCS #11 token, certificate label, and SSL key database file password.

In the message text:

- *error_text* Error message text
- *agreement_name* Replication agreement entry distinguished name

**Example:** None.

**System action:** The LDAP server continues however replication to the consumer server identified by the replication agreement is not started.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Verify that the supplier server credentials entry specified by the **ibm-replicaCredentialsDN** attribute value in the replication agreement entry is using the correct SSL certificate label to perform a SASL EXTERNAL bind to the consumer server. See [Credentials entries](#) for more information on the supplier server credentials entry.

If the consumer server is an IBM Tivoli Directory Server with advanced replication configured, ensure it is properly configured to accept SASL EXTERNAL binds. Verify that the consumer server credentials entry is using the correct distinguished name for the **ibm-slapdMasterDN** attribute value. See [Consumer server entries](#) for more information on the consumer server credentials entry.

**Problem determination:** Not applicable.

---

**GLD8538E** Error '*error_text*' occurred for replica '*agreement_name*': EXTERNAL bind failed.

**Explanation:** The supplier server was unable to successfully perform a SASL EXTERNAL bind to the consumer server. The supplier server credentials entry specified by the **ibm-replicaCredentialsDN** attribute value in the replication agreement entry contains optionally attribute values for the SSL key database file, RACF keyring, PKCS #11 token, certificate label, and SSL key database file password.

In the message text:

- *error_text* Error message text
- *agreement_name* Replication agreement entry distinguished name

**Example:** None.

**System action:** The LDAP server continues with replication to the consumer server identified by the replication agreement.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Verify that the replication agreement on the supplier server is connected to the correct consumer server. If the consumer server is an IBM Tivoli Directory Server with advanced replication, ensure that the **ibm-replicaConsumerID** attribute value in the replication agreement entry has the same value as the **ibm-slapdServerID** attribute defined in the **cn=configuration** entry on the consumer server.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD8540W** Replication for replica '*agreement_name*' will continue to retry the same update after receiving an error.

**Explanation:** The current replication change has failed
to replicate to the consumer server so the supplier
server retries the failed replication change every minute
until it is successful. This error may cause replication
from this replication agreement to be stalled until it is
corrected by the LDAP administrator. The
ibm-replicationState operational attribute in the
replication agreement entry is set to retrying to indicate
the current replication status.

In the message text:
agreement_name
  Replication agreement entry distinguished name
Example: None.
System action: The LDAP server continues however
replication to the consumer server identified by the
replication agreement may be stalled.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Verify that replication
between the supplier and consumer servers is
successfully occurring by querying the replication
agreement entry operational attribute values which
provide replication status. If the ibm-replicationState
operational attribute is set to retrying or the number of
ibm-replicationFailedChanges attribute values is near
the maximum number of replication failures allowed per
backend (as specified by the ibm-slapdReplMaxErrors
attribute value in the cn=Replication,cn=configuration
entry), it may be necessary to compare and
re-synchronize the replication context on both servers.
See Monitoring and diagnosing advanced replication
problems for information on the replication agreement
entry operational attributes.

Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD8541W  Replication for replica
  'agreement_entry' will continue to the
  next update after receiving an error.
Explanation: The current change has failed to
replicate to the consumer server so the supplier server
continues to the next replication change after receiving
this error. This failure may cause replication from this
agreement to be stalled unless it is corrected by the
LDAP administrator.
In the message text:
agreement_name
  Replication agreement entry distinguished name
Example: None.
System action: The LDAP server continues however
replication to the consumer server identified by the
replication agreement may be stalled.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Verify that replication
between the supplier and consumer servers is
successfully occurring by querying the replication
agreement entry operational attribute values which
provide replication status. If the ibm-replicationState
operational attribute is set to retrying or the number of
ibm-replicationFailedChanges attribute values is near
the maximum number of replication failures allowed per
backend (as specified by the ibm-slapdReplMaxErrors
attribute value in the cn=Replication,cn=configuration
entry), it may be necessary to compare and
re-synchronize the replication context on both servers.
See Monitoring and diagnosing advanced replication
problems for information on the replication agreement
entry operational attributes.

Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD8542W  Replication continuing for replica
  'agreement_name' after logging
  update_type for entry 'name' ignoring
  error: return_code 'error_text'
  *additional_error_text*.
Explanation: This change has failed to replicate to the
consumer server for the reason specified so the supplier
server continues to the next replication change after
receiving this error. This failure may cause replication
from this agreement to be stalled unless it is corrected
by the LDAP administrator.
In the message text:
agreement_name
  Replication agreement entry distinguished name
update_type
  Operation type
name
  Entry distinguished name
return_code
  LDAP return code
error_text
  Error text for LDAP return code
additional_error_text
  Additional error text
Example: None.
System action: The LDAP server continues however
replication from this replication agreement may be
stalled.
Operator response: None.
Verify that replication between the supplier and consumer servers is successfully occurring by querying the replication agreement entry operational attribute values which provide replication status from the replication agreement. If the `ibm-replicationState` operational attribute is set to `retrying` or the number of `ibm-replicationFailedChanges` attribute values is near the maximum number of replication failures allowed per backend (as specified by the `ibm-slapdReplMaxErrors` attribute value in the `cn=Replication,cn=configuration` entry), it may be necessary to compare and re-synchronize the replication context on both servers. See Monitoring and diagnosing advanced replication problems for information on the replication agreement entry operational attributes.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

---

GLD8543W Replication continuing for replica 'agreement_entry' after skipping update_type for entry 'name' because of error: return_code 'error_text' 'additional_error_text'.

Explanation: After successfully skipping (deleting) the failed replication change, replication to the consumer server identified by the replication agreement is now continuing.

In the message text:

- `agreement_name`: Replication agreement entry distinguished name
- `update_type`: Operation type
- `name`: Entry distinguished name
- `return_code`: LDAP return code
- `error_text`: Error text for LDAP return code
- `additional_error_text`: Additional error text

Example: None.

System action: The LDAP server continues with replication to the consumer server identified by the replication agreement.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Verify that the consumer server for the replication agreement is running and the replication agreement has the correct `ibm-replicaURL` attribute value specified.

Verify that the supplier server credentials entry specified by the `ibm-replicaCredentialsDN` attribute value in the replication agreement entry is using correct bind information. See [Credentials entries](#) for more information on the supplier server credentials entry.

If the consumer server is an IBM Tivoli Directory Server with advanced replication configured, ensure it is properly configured to accept the supplier server bind credentials. Verify that the consumer server credentials entry is using the correct distinguished name for the `ibm-slapdMasterDN` attribute value. See [Consumer server entries](#) for more information on the consumer server credentials entry.

Problem determination: Not applicable.
GLD8546W  The DN of the credential entry
'credential_name' defined for the
replication agreement
'agreement_name' cannot be found.

Explanation:  The supplier server credentials entry
specified by the ibm-replicaCredentialsDN
attribute in the replication agreement entry cannot be found. This
entry contains the bind information necessary for the supplier server to authenticate with the consumer server.

In the message text:

credential_name  Credentials entry distinguished name
agreement_name  Replication agreement entry distinguished name

Example:  None.

System action:  The LDAP server continues however replication to the consumer server identified by the replication agreement is not started.

Operator response:  None.

User response:  None.

System programmer response:  None.

Administrator response:  Verify that the supplier server credentials entry specified by the ibm-replicaCredentialsDN attribute value in the replication agreement is correct and exists in the directory. Either add the supplier server credentials entry specified by the ibm-replicaCredentialsDN attribute value or modify the ibm-replicaCredentialsDN attribute value to specify a valid supplier server credentials entry. See Credentials entries for more information on the supplier server credentials entry. If the problem persists, contact the service representative.

Problem determination:  Not applicable.

Source:  LDAP
Module:  None.
Routing code:  None.
Descriptor code:  None.
Automation:  Not applicable.

GLD8547E  The DN of the credential entry
'credential_name' defined for the
replication agreement
'agreement_name' cannot be found.

Explanation:  An internal search error occurred while attempting to retrieve the supplier server credentials entry specified by the ibm-replicaCredentialsDN attribute value in the replication agreement entry. This entry contains the bind information necessary for the supplier server to authenticate with the consumer server.

In the message text:

credential_name  Credentials entry distinguished name
agreement_name  Replication agreement entry distinguished name

Example:  None.

System action:  The LDAP server continues however replication to the consumer server identified by the replication agreement is not started.

Operator response:  None.

User response:  None.

System programmer response:  None.

Administrator response:  Verify that the supplier server credentials entry specified by the ibm-replicaCredentialsDN attribute value in the replication agreement is correct and exists in the directory. Either add the supplier server credentials entry specified by the ibm-replicaCredentialsDN attribute value or modify the ibm-replicaCredentialsDN attribute value to specify a valid supplier server credentials entry. See Credentials entries for more information on the supplier server credentials entry. If the problem persists, contact the service representative.

Problem determination:  Not applicable.

Source:  LDAP
Module:  None.
Routing code:  None.
Descriptor code:  None.
Automation:  Not applicable.

GLD8551E  Error 'error_text' occurred for replica
'agreement_name': delete failed for
entry 'name' change ID changeID.

Explanation:  During replication from this supplier server to the consumer server defined in the replication agreement, a delete operation failed. The error string shows the reason why the operation failed. The change ID is used to record the replication change in the backend where the replication agreement entry resides.
In the message text:

<table>
<thead>
<tr>
<th>error_text</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error text</td>
</tr>
</tbody>
</table>

| agreement_name |
| Replication agreement entry distinguished name |

| name |
| Entry distinguished name |

| changeID |
| Replication change identifier |

**Example:** None.

**System action:** The LDAP server continues however replication to the consumer server identified by the replication agreement may be stalled.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** None.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD8553E** Error retrieving pending change count for replica ‘agreement_name’.

**Explanation:** An internal error occurred while attempting to retrieve the number of pending replication changes from the backend where the replication agreement entry resides. The current number of pending replication changes are returned in the `ibm-replicationPendingChanges` operational attribute in the replication agreement entry.

In the message text:

| agreement_name |
| Replication agreement entry distinguished name |

**Example:** None.

**System action:** The LDAP server continues however the search of the replication agreement entry is not successful.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Verify that the backend where the replication agreement entry resides is functioning and handling requests correctly. If the problem persists, contact the service representative.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD8556E** The weekly schedule DN object ‘schedule_name’ specified in the replication agreement cannot be found or is not a weekly schedule entry.

**Explanation:** The weekly schedule entry which contains replication scheduling information for the replication agreement entry cannot be found or is not a valid weekly schedule entry. A valid weekly schedule entry has an object class value of `ibm-replicationWeeklySchedule`. The weekly schedule entry is specified in the `ibm-replicaScheduleDN` attribute value of the replication agreement entry.

In the message text:

| schedule_name |
| Replication schedule entry distinguished name |

**Example:** None.

**System action:** The LDAP server continues however replication updates to the consumer server identified by the replication agreement are immediately replicated.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** See [Monitoring and diagnosing advanced replication problems](#) for information on recovering from advanced replication problems.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.
GLD8559E The daily schedule DN object 'schedule_name' specified in the weekly schedule entry cannot be found or is not a daily schedule entry.

Explanation: The weekly schedule entry contains a daily replication schedule entry that cannot be found or is not a valid daily schedule entry. The weekly schedule entry uses the ibm-scheduleSunday, ibm-scheduleMonday, ibm-scheduleTuesday, ibm-scheduleWednesday, ibm-scheduleThursday, ibm-scheduleFriday, and ibm-scheduleSaturday attribute values to point to daily replication schedule entries.

In the message text:

schedule_name Replication schedule entry distinguished name

Example: None.

System action: The LDAP server continues however replication updates to the consumer server identified by the replication agreement are immediately replicated.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Verify that the object class value of the daily schedule entry specified in the weekly schedule entry is ibm-replicationDailySchedule and that the entry exists. See Schedule entries for information on replication schedule entries.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD8560E Error while updating status for replica 'agreement_name' to last change ID changeID.

Explanation: An internal error occurred while updating the last change ID in the backend replication status table where the replication agreement entry resides.

In the message text:

agreement_name Replication agreement entry distinguished name

changeID Replication change identifier

Example: None.

System action: The LDAP server continues however the replication agreement status cannot be updated in the backend replication status table.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: If an entry needs to be moved from one replication context to another, retrieve the entry by performing a search operation and then delete the entry from the replication context. Then re-add the entry with the new distinguished name to the desired replication context.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD8563E Cannot move 'name' from one replication context to another.

Explanation: An entry is not allowed to be moved into or out of a replication context using the modify DN operation. When a replication context is configured, a modify dn operation is only allowed to occur within the same replication context.

In the message text:

name Entry distinguished name

Example: None.

System action: The LDAP server continues however the requested client modify dn operation is not successful.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Verify that the backend where the replication agreement entry resides is functioning and handling requests correctly. If the problem persists, contact the service representative.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.
GLD8564E  The replication configuration could not be read.

Explanation:  An internal error occurred while performing a search for the advanced replication configuration entries in the CDBM backend.

Example:  None.

System action:  The LDAP server continues however advanced replication configuration is not successful.

Operator response:  None.

User response:  None.

System programmer response:  None.

Administrator response:  Verify that the CDBM backend is configured correctly and that the server has access to the CDBM backend. If the problem persists, contact the service representative.

Problem determination:  Not applicable.

Source:  LDAP

Module:  None.

Routing code:  None.

Descriptor code:  None.

Automation:  Not applicable.

GLD8565E  The replication configuration DN object 'name' cannot be found.

Explanation:  An internal error occurred while performing a search for an advanced replication configuration entry in the CDBM backend.

In the message text:

name  Entry distinguished name

Example:  None.

System action:  The LDAP server continues however advanced replication configuration is not successful.

Operator response:  None.

User response:  None.

System programmer response:  None.

Administrator response:  Verify that the CDBM backend is configured correctly and that the server has access to the CDBM backend. If the problem persists, contact the service representative.

Problem determination:  Not applicable.

Source:  LDAP

Module:  None.

Routing code:  None.

Descriptor code:  None.

Automation:  Not applicable.

GLD8566E  Error on configuration entry with DN 'name'; required attribute attribute_name is missing.

Explanation:  An internal error occurred while performing a search for an advanced replication configuration entry in the CDBM backend.

In the message text:

name  Entry distinguished name

attribute_name  Attribute type

Example:  None.

System action:  The LDAP server continues however advanced replication configuration is not successful.

Operator response:  None.

User response:  None.

System programmer response:  None.

Administrator response:  Verify that the specified replication advanced configuration entry has the required attributes and the attribute value data is correct. If the problem persists, contact the service representative.

Problem determination:  Not applicable.

Source:  LDAP

Module:  None.

Routing code:  None.

Descriptor code:  None.

Automation:  Not applicable.

GLD8567E  Error on configuration entry with DN 'name' attribute attribute_name value 'attribute_value'. Value ignored.

Explanation:  An internal error occurred while performing a search for an advanced replication configuration entry in the CDBM backend.

In the message text:

name  Entry distinguished name

attribute_name  Attribute type

attribute_value  Attribute value

Example:  None.
System action: The LDAP server continues however advanced replication configuration is not successful.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Verify that the specified replication advanced configuration entry has the correct attribute value data. If the problem persists, contact the service representative.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD8568E Error initializing the backend replication table used for replication failures.

Explanation: An internal error occurred while initializing the backend replication table used for storing replication failures. The backend replication table is used by all replication agreements within the backend.

Example: None.

System action: The LDAP server continues however replication failures are not stored in the backend replication table.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD8568E Error initializing the backend replication table used for replication failures.

Explanation: An internal error occurred while initializing the backend replication table used for storing replication failures. The backend replication table is used by all replication agreements within the backend.

Example: None.

System action: The LDAP server continues however replication failures are not stored in the backend replication table.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

In the message text:

host_name

LDAP host name

port_number

LDAP port number

Example: None.

System action: The LDAP server continues.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD8570E Propagation of replication topology entries to host 'host_name' port port_number failed with error code return_code.

Explanation: An error occurred while using the Replication topology extended operation to synchronize the replication topology entries on the specified consumer server.

In the message text:

host_name

LDAP host name

port_number

LDAP port number

return_code

LDAP return code

Example: None.

System action: The LDAP server continues however the replication topology entries are not successfully synchronized on the specified consumer server.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: A previously issued message indicates the reason for the Replication topology extended operation error. Correct the error on the targeted consumer server and then retry the Replication topology extended operation.

Problem determination: Not applicable.

In the message text:

host_name

LDAP host name

port_number

LDAP port number

Example: None.

System action: The LDAP server continues however the replication topology entries are not successfully synchronized on the specified consumer server.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

In the message text:

host_name

LDAP host name

port_number

LDAP port number

Example: None.

System action: The LDAP server continues however the replication topology entries are not successfully synchronized on the specified consumer server.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

In the message text:

host_name

LDAP host name

port_number

LDAP port number

Example: None.

System action: The LDAP server continues however the replication topology entries are not successfully synchronized on the specified consumer server.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.
GLD8571I Propagation of replication topology entries will continue with the next target server.

Explanation: The Replication topology extended operation is continuing to synchronize replication topology entries on the next targeted consumer server.

Example: None.

System action: The LDAP server continues with the Replication topology extended operation on the next targeted consumer server.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD8572I Propagation of replication topology entries is complete.

Explanation: The Replication topology extended operation has successfully synchronized replication topology entries on all consumer servers defined within the replication context.

Example: None.

System action: The LDAP server continues.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD8578W The extended operation cannot replicate to target server 'host_name' port port_number because the target server does not support replication topology entries.

Explanation: An error occurred while using the Replication topology extended operation against a consumer server that is not configured for synchronizing replication topology entries. The Replication topology extended operation cannot synchronize replication topology entries on the specified consumer server.

In the message text:

'host_name' LDAP host name
'port_number' LDAP port number

Example: None.

System action: The LDAP server continues however the replication topology entries are not successfully synchronized on the specified consumer server.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Verify that the targeted consumer server supports the Replication topology extended operation. If the targeted consumer server does not support the Replication topology extended operation, exclude that server as a target of the extended operation.

Problem determination: Not applicable.

Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD8579E The extended operation cannot replicate entries to the target server 'host_name' port port_number because the target server does not have the suffix 'name'.

Explanation: An error occurred while using the Replication topology extended operation against a consumer server that does not have the appropriate suffix configured in its server configuration file. Since the consumer server does not have the appropriate suffix configured, the replication topology entries are not allowed to be added.

Chapter 30. LDAP server messages 687
In the message text:

- **host_name**: LDAP host name
- **port_number**: LDAP port number
- **name**: Entry distinguished name

**Example**: None.

**System action**: The LDAP server continues however the replication topology entries are not successfully synchronized on the specified consumer server.

**Operator response**: None.

**User response**: None.

**System programmer response**: None.

**Administrator response**: None.

**Problem determination**: Not applicable.

**Source**: LDAP

**Module**: None.

**Routing code**: None.

**Descriptor code**: None.

**Automation**: Not applicable.

---

GLD8580I Replication conflict: a conflict has been detected on host 'host_name' port port_number. A request to re-add the entry of DN 'name' has been received.

**Explanation**: A replication conflict occurred between the supplier and the targeted consumer server with the specified entry. Since the consumer server supports replication conflict resolution within this replication context, a request has been received by the supplier server from the consumer server to resend the conflicted entry back to the consumer server.

In the message text:

- **host_name**: LDAP host name
- **port_number**: LDAP port number
- **name**: Entry distinguished name

**Example**: None.

**System action**: The LDAP server continues.

---

GLD8581I Replication conflict: re-add DN 'name' to solve a replication conflict on host 'host_name' port port_number.

**Explanation**: A replication conflict occurred between the supplier and the targeted consumer server with the specified entry. Since the consumer server supports replication conflict resolution within this replication context, the conflicted entry has been resent to the consumer server. The intention is to re-synchronize the entry on the supplier and consumer servers.

In the message text:

- **name**: Entry distinguished name
- **host_name**: LDAP host name
- **port_number**: LDAP port number

**Example**: None.

**System action**: The LDAP server continues.

---

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.
GLD8582I Replication conflict: re-add of DN 'name' to host 'host_name' port port_number succeeded.

**Explanation:** The replication conflict that occurred with the specified entry between the supplier and consumer servers has been resolved successfully. The entry is now synchronized between the supplier and consumer servers.

In the message text:
- **name** Entry distinguished name
- **host_name** LDAP host name
- **port_number** LDAP port number

**Example:** None.

**System action:** The LDAP server continues.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Since the entry is not synchronized between the supplier and consumer servers, future replication conflicts may occur with this entry. See Recovering from advanced replication errors for information on synchronizing the supplier and consumer servers.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

GLD8583W Replication conflict: re-add of DN 'name' to host 'host_name' port port_number failed.

**Explanation:** The replication conflict that occurred with the specified entry between the supplier and consumer servers has not been resolved successfully. The entry is not synchronized between the supplier and consumer servers. The supplier server does not attempt to re-synchronize the conflicted entry on the consumer server.

In the message text:
- **name** Entry distinguished name
- **host_name** LDAP host name
- **port_number** LDAP port number

**Example:** None.

**System action:** The LDAP server continues however the specified entry is not synchronized between the supplier and consumer servers.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Since the entry is not synchronized between the supplier and consumer servers, future replication conflicts may occur with this entry. See Recovering from advanced replication errors for information on synchronizing the supplier and consumer servers.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

GLD8584E Replication conflict: re-add of DN 'name' to resolve a conflict will not continue because the entry is too large.

**Explanation:** A replication conflict occurred between the supplier and the consumer server with the specified entry. Since the consumer server supports replication conflict resolution within this replication context, this supplier server has been requested to resend the conflicted entry to the consumer server however the size of the conflicted entry exceeds the maximum size allowed. The maximum conflicted entry size that a supplier server can resend to the consumer server is specified by the ibm-slapdReplConflictMaxEntrySize attribute value in the cn=Replication,cn=configuration configuration entry.

In the message text:
- **name** Entry distinguished name

**Example:** None.

**System action:** The LDAP server continues however the specified entry is not synchronized between the supplier and consumer servers.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Since the entry is not synchronized between the supplier and consumer servers, future replication conflicts may occur with this entry. See Recovering from advanced replication errors for information on synchronizing the supplier and consumer servers.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD8586E Attribute 'attribute_name' is missing from entry 'name'.

Explanation: If a replication context is created with an entry that is not a suffix level entry, ACLs must be defined explicitly in that entry. The following ACL attribute values must be added to the replication context entry for non-suffix level entries: aclEntry, aclPropogate, entryOwner, and ownerPropogate.

In the message text:

attribute_name
Attribute type
name
Entry distinguished name

Example: None.
System action: The LDAP server continues however the requested client add operation is not successful.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.

GLD8587E The Replication Topology extended operation failed to quiesce the context 'context_name' on host 'host_name' port port_number. The operation will not continue with this server.

Explanation: An error occurred while using the Replication topology extended operation to quiesce a replication context on the specified consumer server. The Replication topology extended operation continues to the next targeted consumer server.

In the message text:

context_name
Replication context entry distinguished name
host_name
LDAP host name
port_number
LDAP port number

Example: None.
System action: The LDAP server continues.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD8589E The Replication Topology extended operation failed to unquiesce the context 'context_name' on host 'host_name' port port_number. The operation will not continue with this server.

Explanation: An error occurred while using the Replication topology extended operation to unquiesce the replication context on the targeted consumer server.

In the message text:
context_name Replication context entry distinguished name
host_name LDAP host name
port_number LDAP port number

Example: None.

System action: The LDAP server continues.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.

Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD8590I The Replication Topology extended operation successfully unquiesced the context 'context_name' on host 'host_name' port port_number.

Explanation: The Replication topology extended operation has successfully unquiesced the replication context on the specified consumer server.

In the message text:
context_name Replication context entry distinguished name
host_name LDAP host name
port_number LDAP port number

Example: None.

System action: The LDAP server continues.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.

Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD8591E The Replication Topology extended operation failed to add a suffix 'name' to the configuration file of target host 'host_name' port port_number. The operation will not continue with this server.

Explanation: An error occurred while using the Replication topology extended operation to add replication topology entries on the targeted consumer server.

If the Replication topology extended operation was targeted against a non-z/OS IBM Tivoli Directory server, the add of an ibm-slapdSuffix attribute value to the cn=Directory, cn=RDBM Backends, cn=IBM Directory, cn=Schemas, cn=Configuration entry was not successful.

If the Replication topology extended operation was targeted against a z/OS IBM Tivoli Directory Server, verify that the targeted consumer server has the suffix in its server configuration file.

In the message text:
name
Entry distinguished name
host_name
LDAP host name
port_number
LDAP port number

Example: None.
System action: The LDAP server continues however
the Replication topology extended operation on the
consumer server identified by the replication agreement
is not successful.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD8592I The Replication Topology extended
operation successfully added a suffix
'target host 'host_name' port
port_number.'

Explanation: If the Replication topology extended
operation was targeted against a non-z/OS IBM Tivoli
Directory Server, it has successfully added an
ibm-slapdSuffix attribute value to the rootDSE entry.
This allows the Replication topology extended
operation to synchronize replication topology entries on
the consumer server identified by the replication
agreement.

In the message text:
name
Entry distinguished name
host_name
LDAP host name
port_number
LDAP port number

Example: None.
System action: The LDAP server continues.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD8593E The Replication Topology extended
operation failed to purge the queue
that is associated with the replication
agreement 'agreement_name' on host
'host_name' port port_number.

Explanation: An error occurred while purging the
replication queue for the specified replication agreement
entry when using the Replication topology extended
operation. The replication queue on the supplier server
is purged when the replication agreement entry already
exists on the consumer server.

In the message text:
agreement_name
Replication agreement entry distinguished name
host_name
LDAP host name
port_number
LDAP port number

Example: None.
System action: The LDAP server continues however
the Replication topology extended operation on the
consumer server identified by the replication agreement
is not successful.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Correct the problem by
purging the replication queue or by deleting the
replication agreement entry on the consumer server.
The replication queue for the agreement can be purged
by using the Control replication queue extended
operation in the ldapexop utility. Then retry the
Replication topology extended operation. See
Recovering from advanced replication errors for information on recovering from advanced replication problems.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

**GLD8594I** There is no replication queue that is associated with the replication agreement 'agreement_name' on host 'host_name' port port_number. Or the Replication Topology extended operation successfully purged the queue.

**Explanation:** The Replication topology extended operation has successfully purged the replication queue associated with the specified replication agreement entry.

In the message text:

*agreement_name*
Replication agreement entry distinguished name

*host_name*
LDAP host name

*port_number*
LDAP port number

**Example:** None.

**System action:** The LDAP server continues.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Verify that the consumer server for the replication agreement is running and the replication agreement has the correct ibm-replicaURL attribute value specified. Then retry the Replication topology extended operation.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

**GLD8595I** Topology successfully replicated to number of maximum_number servers.

**Explanation:** The Replication topology extended operation has successfully synchronized the replication topology entries on the number of consumer servers specified.

In the message text:

*number*
Number of consumer servers successfully replicated

*maximum_number*
Total number of consumer servers

**Example:** None.

**System action:** The LDAP server continues.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.
GLD8598E The Replication Topology extended operation timed out.

Explanation: A timeout error occurred while using the Replication topology extended operation. The time limit specified on the Replication topology extended operation has been exceeded.

Example: None.

System action: The LDAP server continues however the Replication topology extended operation is not successful.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD8601E The update logged as a failure with failure ID failureID for replication agreement DN 'agreement_name' has been removed from the backend replication failure table.

Explanation: The Control replication error log extended operation has successfully removed all replication failures from the backend replication table.

In the message text:

agreement_name

Replication agreement entry distinguished name

Example: None.

System action: The LDAP server continues.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD8602E All updates logged as failures for replication agreement DN 'agreement_name' have been removed from the backend replication failure table.

Explanation: The Control replication error log extended operation has successfully removed all replication failures from the backend replication table.

In the message text:

agreement_name

Replication agreement entry distinguished name

Example: None.

System action: The LDAP server continues.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD8603E Unable to log failure in the backend replication failure table for replication agreement DN 'agreement_name' for entry with change ID changeID.

Explanation: An internal error occurred while adding the replication failure ID in the backend replication table.

In the message text:

failureID

Replication failure change identifier
In the message text:

**agreement_name**
Replication agreement entry distinguished name

**changeID**
Replication change identifier

**Example:** None.

**System action:** The LDAP server continues however the replication failure cannot be added to the backend replication table.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Verify that the backend where the replication agreement entry resides is functioning and handling requests correctly. The supplier and consumer servers may no longer be synchronized.
See [Recovering from advanced replication errors] for information on synchronizing the supplier and consumer servers. If the problem persists, contact the service representative.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD8604W** Reached or exceeded the limit max_replication_errors for the backend replication failure table for replication agreement DN 'agreement_name'.

**Explanation:** The supplier server has reached or exceeded the maximum number of replication errors allowed for all replication agreement entries within this backend. The maximum number of replication errors allowed is controlled by the ibm-slapdReplMaxErrors attribute value in the cn=Replication,cn=configuration entry.

In the message text:

**max_replication_error**
Maximum number of advanced replication errors allowed

**agreement_name**
Replication agreement entry distinguished name

**Example:** None.

**System action:** The LDAP server continues.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** None.

**Problem determination:** None.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD8608I** Replication for DN 'agreement_name' will use the single threaded, synchronous method.

**Explanation:** Replication to the consumer server identified by the replication agreement is using the synchronous, single threaded method. The synchronous method is the only supported replication method on the IBM Tivoli Directory Server for z/OS.

In the message text:

**agreement_name**
Replication agreement entry distinguished name

**Example:** None.

**System action:** The LDAP server continues.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** None.

**Problem determination:** None.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD8610E** Replication for DN 'agreement_name' terminated because of an unsupported replication method.

**Explanation:** Replication to the consumer server identified by the replication agreement is using a replication method that is not supported. The synchronous, single threaded method is the only supported replication method on the IBM Tivoli Directory Server for z/OS.
In the message text:

agreement_name
Replication agreement entry distinguished name

method
Replication method

Example: None.

System action: The LDAP server continues however replication to the consumer server identified by the replication agreement is not started.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: See Monitoring and diagnosing advanced replication problems for information on recovering from advanced replication problems.

Problem determination: Not applicable.

Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD8618E Replication for replica 'agreement_name' will continue to retry the same failed update with change ID changeID until it is successful.

Explanation: An error occurred while replicating the update with the specified change ID to the consumer server identified by the replication agreement entry. The failed change gets retried every minute until it succeeds or the failed change is removed from the replication queue by the LDAP administrator. When this failed change occupies the lead position in the pending replication queue, all other replication updates are blocked and replication is stalled.

In the message text:

agreement_name
Replication agreement entry distinguished name

changeID
Replication change identifier

Example: None.

System action: The LDAP server continues however replication to the consumer server identified by the replication agreement is stalled.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: See Monitoring and diagnosing advanced replication problems for information on recovering from advanced replication problems.

Problem determination: Not applicable.

Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD8620E Error occurred processing the replica URL for replication agreement 'agreement_name'.

Explanation: An error occurred while parsing the ibm-replicaURL attribute value in the replication agreement entry. The value specified is not a valid LDAP URL or is empty.

In the message text:

agreement_name
Replication agreement entry distinguished name

Example: None.

System action: The LDAP server continues however replication to the consumer server identified by the replication agreement is not started.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Update the ibm-replicaURL attribute value in the replication agreement entry so a valid LDAP URL is specified. Verify that the consumer server’s hostname and optional port number are correct.

Problem determination: Not applicable.

Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD8628I Creating surrogate entry 'name' on partial replica 'host_name' port port_number.

Explanation: The specified entry is being created on the consumer server because this parent entry is missing. If the ibm-replicationCreateMissingEntries optional attribute in the replication agreement is set to
true, then missing parent entries on the consumer
server are automatically created.
In the message text:

name
Entry distinguished name
host_name
LDAP host name
port_number
LDAP port number

Example: None.

System action: The LDAP server continues.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD8629E Creation of surrogate entry 'name' on
central replica 'host_name' port
port_number failed.

Explanation: An error occurred while attempting to
automatically create the specified entry on the
consumer server. This parent entry was probably
missing because the replication filter excluded it from
being replicated to the consumer server. If the
ibm-replicationCreateMissingEntries optional attribute
in the replication agreement is set to true, then missing
parent entries on the consumer server are automatically
created.
In the message text:

name
Entry distinguished name
host_name
LDAP host name
port_number
LDAP port number

Example: None.

System action: The LDAP server continues.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD8630I Creation of surrogate entry 'name' on
partial replica 'host_name' port
port_number succeeded.

Explanation: The specified entry has successfully
been created because this parent entry was originally
missing on the consumer server. If the
ibm-replicationCreateMissingEntries optional attribute
in the replication agreement is set to true, then missing
parent entries on the consumer server are automatically
created.
In the message text:

name
Entry distinguished name
host_name
LDAP host name
port_number
LDAP port number

Example: None.

System action: The LDAP server continues.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.
GLD8632E The replication filter entry ‘filter_name’ defined for the replication agreement ‘agreement_name’ cannot be found.

Explanation: An internal search error occurred while querying the replication filter entry specified in the ibm-replicationFilterDN attribute value of the replication agreement entry. The replication filter entry specified in the replication agreement cannot be found or the entry specified does not have an object class value of ibm-replicationFilter.

In the message text:

filter_name

Replication filter entry distinguished name

agreement_name

Replication agreement entry distinguished name

Example: None.

System action: The LDAP server continues however replication to the consumer server identified by the replication agreement is not started.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Update the add or modify of the ibm-replicationFilterAttr attribute value in the replication filter entry to only specify non-operational attributes as part of the inclusion or exclusion list in a replication filter. See [Partial replication] for information on replication filter entries.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD8633W The operational attribute attribute_name specified as part of the replication filter inclusion/exclusion list is not allowed.

Explanation: An error occurred while attempting to add or modify a replication filter that had an operational attribute specified. Operational attributes cannot be specified as part of the filter inclusion or exclusion list. Replication filters are specified in the ibm-replicationFilterAttr attribute value in the replication filter entry.

In the message text:

attribute_name

Attribute type

Example: None.

System action: The LDAP server continues however the add or modify operation of the replication filter entry is not successful.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD8634I Modifications to only the ACL attributes of an entry will not be filtered.

Explanation: Partial replication is configured for the replication agreement however replication filtering has been bypassed since only the ACL attribute values of an entry have been modified. Updates to ACL attribute values are always replicated to a consumer server.

Example: None.

System action: The LDAP server continues.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.
GLD8635I The Replication Topology extended operation is performed against the server at host 'host_name' port 'port_number' that does not support filtered replication. Hence, the filtered replication related attributes will not be sent to this server.

Explanation: The consumer server identified by the replication agreement entry does not support filtered replication. However, the replication agreement has a replication filter entry specified in the ibm-replicationFilterDN attribute value. Although the consumer server does not support partial replication, filtered entries are still replicated to the consumer server.

In the message text:

host_name LDAP host name
port_number LDAP port number

Example: None.

System action: The LDAP server continues.

Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.

Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD8639E The filter entry 'filter_name' is in use and cannot be deleted.

Explanation: The replication filter entry cannot be deleted because a replication agreement entry has a reference to this entry in an ibm-replicationFilterDN attribute value.

In the message text:

filter_name Replication filter entry distinguished name

Example: None.

System action: The LDAP server continues however the requested delete client operation is not successful.

Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Perform a search on all replication agreement entries in the directory to find the entry that has an ibm-replicationFilterDN attribute value with the distinguished name (DN) of the entry being deleted. Modify the ibm-replicationFilterDN attribute value in the replication agreement entry to remove the reference to the entry that is being deleted. Then retry the delete client operation.

Problem determination: Not applicable.

Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD8637I Restricted Access to the replication topology is set to value.

Explanation: The ibm-slapdReplRestrictedAccess attribute value in the cn=Replication,cn=configuration entry has been set to the value specified in this message. If set to true, only the LDAP administrator and the master server DN have access to replication topology entries. If set to false, other users with the proper ACL authority can access the replication topology entries.

In the message text:

value True or false

Example: None.

System action: The LDAP server continues.

Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.

Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD8640I Replication error logging for replication agreement DN 'agreement_name' is unlimited.

Explanation: The number of replication failures stored for the specified replication agreement is unlimited.

Chapter 30. LDAP server messages 699
When the `ibm-slapdReplMaxErrors` attribute value in the `cn=Replication,cn=configuration` entry is set to -1, there is no limit on the number of replication failures stored in the backend where the replication agreement entry resides.

In the message text:

- **agreement_name**
  - Replication agreement entry distinguished name

**Example:** None.

**System action:** The LDAP server continues.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** If the limit needs to be decreased, modify the `ibm-slapdReplMaxErrors` attribute value in the `cn=Replication,cn=configuration` entry and specify a small positive number. The `ibm-slapdReplMaxErrors` attribute value applies to all replication agreements within the backend.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD8642W** Internal search for the parent entry 'name' failed.

**Explanation:** An internal server error occurred while searching for the specified parent entry in this server. This entry is a missing parent entry on the consumer server and is needed for replication to the consumer server to continue.

In the message text:

- **name**
  - Entry distinguished name

**Example:** None.

**System action:** The LDAP server continues however the requested replication filter update client operation is not successful. If the replication filter is an existing replication filter entry, the filter is ignored.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Verify that the attribute type specified in the replication filter exists in the schema. Modify the replication filter to use an attribute type that exists in the schema or update the schema to add the missing attribute type.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD8643E** The attribute `attribute_name` specified in the replication filter is not found in the schema.

**Explanation:** The specified attribute type in the replication filter was not found in the schema. A replication filter was specified in the `ibm-replicationFilterAttr` attribute value in the replication filter entry. Another message identifies the replication filter entry and value that is in error.

In the message text:

- **attribute_name**
  - Attribute type

**Example:** None.

**System action:** The LDAP server continues however the requested replication filter update client operation is not successful. If the replication filter is an existing replication filter entry, the filter is ignored.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Verify that the attribute type specified in the replication filter exists in the schema. Modify the replication filter to use an attribute type that exists in the schema or update the schema to add the missing attribute type.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD8644E** The objectclass `value` specified in the replication filter is not found in the schema.

**Explanation:** The specified object class value in the replication filter was not found in the schema. A replication filter was specified in the `ibm-replicationFilterAttr` attribute value in the...
replication filter entry. Another message identifies the
replication filter entry and value that is in error.
In the message text:

value
Objectclass value

Example: None.

System action: The LDAP server continues however
the requested replication filter update client operation is
not successful. If the replication filter is an existing
replication filter entry, the filter is ignored.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.

Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD8645E A replication filter 'attribute_value'
that
is not valid is specified in entry 'name'.

Explanation: The specified filter in the replication filter
entry that was added or modified was not valid. There is
a specific format required for replication filters that are
specified in the ibm-replicationFilterAttr attribute
value.
In the message text:

attribute_value
Attribute value

Example: None.

System action: The LDAP server continues however
the requested replication filter update client operation is
not successful.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Update the
ibm-replicationFilterAttr attribute value in the
replication filter entry so that it is in the correct format.
See [Partial replication] for information on the replication
filter format.

GLD8647E Kerberos authentication is specified
for the replication agreement
'agreement_name'. Kerberos
authentication is not supported on this
platform.

Explanation: The object class of the supplier server
credentials entry defined in the ibm-
replicaCredentialsDN attribute value for the replication
agreement entry is not valid. The only supported object
class values for supplier server credential entries are
ibm-replicationCredentialsSimple and
ibm-replicationCredentialsExternal. The IBM Tivoli
Directory Server for z/OS does not support a supplier
server credentials entry that has an object class value
of ibm-replicationCredentialsKerberos.
In the message text:

agreement_name
Replication agreement entry distinguished name

Example: None.

System action: The LDAP server continues however
replication to the consumer server identified by the
replication agreement is not started.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Verify that the
ibm-replicaCredentialsDN attribute value in the
replication agreement entry does not reference a
supplier server credentials entry with an
ibm-replicationCredentialsKerberos object class
value. Modify the ibm-replicaCredentialsDN attribute
value in the replication agreement entry to reference a
supplier server credentials entry with an object class
value of ibm-replicationCredentialsSimple or
ibm-replicationCredentialsExternal.

Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.
Unable to open lost and found log file 'filename'.

Explanation: The lost and found log file specified by the ibm-slapdLog attribute value in the cn=Replication,cn=Log Management,cn=Configuration entry cannot be opened. The lost and found log file is created by the consumer server any time a replication conflict occurs. Any entries that are deleted on the consumer server because of a replication conflict are stored in LDIF format in this file.

In the message text:

filename
Replication lost and found log file

Example: None.

System action: The LDAP server continues however replication conflicts are not written to the lost and found log file.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

Replication agreement 'agreement_name' is now suspended because 'filter_name' is not a valid filter entry.

Explanation: The replication agreement is now suspended because the replication filter entry specified in the ibm-replicationFilterDN attribute value does not exist or there are no valid ibm-replicationFilterAttr attribute values specified in the entry. The ibm-replicationOnHold attribute value has been automatically set to true in the replication agreement until the problems can be corrected.

In the message text:

agreement_name
Replication agreement entry distinguished name

filter_name
Replication filter entry distinguished name

Example: None.

System action: The LDAP server continues.

Operator response: None.

User response: None.

System programmer response: None.

Verify that the LDAP server has the appropriate access to the directories and to the file itself. Modify the ibm-slapdLog attribute value in the cn=Replication,cn=Log Management,cn=Configuration entry to specify a fully-qualified z/OS UNIX System Services filename and directory location where the LDAP server can create the lost and found log file.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

Replication agreement 'agreement_name' is active.

Explanation: Replication to the consumer server identified by the replication agreement is active because the ibm-replicationOnHold attribute is set to false or is not present.

In the message text:

agreement_name
Replication agreement entry distinguished name

Example: None.

System action: The LDAP server continues.

Operator response: None.

User response: None.

System programmer response: None.
GLD8651I Replication agreement

agreement_name is suspended.

Explanation: Replication to the consumer server identified by the replication agreement is suspended because the ibm-replicationOnHold attribute is set to true.

In the message text:

agreement_name

Replication agreement entry distinguished name

Example: None.

System action: The LDAP server continues.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD8652I Replication agreement

agreement_name is suspended. The Cascading control replication extended operation will skip this agreement.

Explanation: The Cascading control replication extended operation was attempted for a replication context that has one or more suspended replication agreements. The extended operation will skip this agreement and continue to the next replication agreement.

agreement_name

Replication agreement entry distinguished name

Example: None.

System action: The LDAP server continues.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD8653W Duplicate ibm-replicaServerId value 'replicaServerId' defined in subentry 'name' for replication context 'context_name'.

Explanation: The ibm-replicaServerID value defined in a replica subentry is already in use by another replicasubentry defined under the replication context indicated in the message. All replica subentries in the same replication context should have unique ibm-replicaServerID values defined. Specifications for the ibm-replicationServersMaster and ibm-replicaGateway attribute values are derived from the last replica subentry processed that matches this server’s ibm-slapdServerID. The replication configuration may change when the server is restarted based on the internal processing order of the replica subentries.

In the message text:

replicaServerId

Value of duplicate serverID

name

Entry distinguished name containing duplicate serverID

context_name

Replication context distinguished name related to the subentry

Example: None.

System action: The LDAP server continues.

Operator response: None.
GLD8654I Advanced replication initialization failed.
Explanation: An internal error occurred while attempting to initialize advanced replication support.
Example: None.
System action: If the srvStartUpError option in the LDAP server configuration file is set to ignore, the LDAP server continues to run with those backends that successfully start however advanced replication does not start. If the srvStartUpError option is set to terminate (this is the default if the configuration option is not specified), the LDAP server ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Restart the LDAP server with ERROR debug level set. The LDAP trace debug output may assist in locating and correcting the problem.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD8797E Internal processing error in the server; return code return_code from pthread library.
Explanation: An error occurred in one of the following LE pthread library functions:
- pthread_mutex_lock()
- pthread_mutex_unlock()
- pthread_cond_timedwait()
- pthread_setspecific()
- pthread_cond_broadcast()
The name of the pthread routine is in an LDAP ERROR trace. Refer to the description of the LE pthread routines in z/OS XL C/C++ Run-Time Library Reference for more information on the error.
In the message text:
- return_code
  Return code from pthread library
- Example: None.
- System action: The LDAP server continues.
- Operator response: None.
- User response: None.
- System programmer response: None.
- Administrator response: None.
- Problem determination: Not applicable.
- Source: LDAP
- Module: None.
- Routing code: None.
- Descriptor code: None.
- Automation: Not applicable.

GLD8801I ldapdiff_usage_message
Explanation: The ldapdifft utility help and usage menu.
Example: None.
System action: The program ends.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.
GLD8802I Only operational attributes differ for this entry.
Explanation: The ldapdiff utility has detected a difference in the operational attributes for the entry. The ldapdiff utility is using the ibm-entryCheckSumOp attribute, which contains a checksum value of the operational attribute values, to quickly detect entry differences.
Example: None.
System action: The program continues.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD8803I Schema compare is complete.
Explanation: The ldapdiff utility has completed schema comparison on both LDAP servers.
Example: None.
System action: The program continues.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD8804I Schema compare is in progress...
This might take a few minutes...
Explanation: The ldapdiff utility is performing schema comparison on both LDAP servers.
Example: None.
System action: The program continues.
Operator response: None.
User response: None.
System programmer response: None.

GLD8805I Successfully connected to both servers.
Explanation: The ldapdiff utility has successfully connected to both LDAP servers that are to be compared.
Example: None.
System action: The program continues.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD8806E Error occurred while connecting to server: exception_text.
Explanation: The ldapdiff utility encountered an error while attempting to connect to the LDAP server. The exception occurred while attempting to connect to the LDAP server that is indicated in the exception.
In the message text:
exception_text
Exception text
Example: None.
System action: The program ends.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Ensure that the targeted LDAP servers are running before starting the **ldapdiff** utility. Verify that TCP/IP communication is working properly between the utility and each targeted LDAP server. If additional information is needed to solve the problem, specify `-d ALL` on the command line. Then restart the program.

Problem determination: Not applicable.

Source: LDAP

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

---

**GLD8807E** Incorrect SSL options specified for server `"server_host:server_port"`.

Explanation: The **ldapdiff** utility detected missing or incorrect SSL command line parameters while attempting to initialize the SSL connection to the LDAP server and port number indicated in the message. Either the parameter is not known, the value specified for the parameter is not supported, or a parameter is missing.

In the message text:

- `server_host` Server hostname
- `server_port` Server port number

Example: None.

System action: The program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

---

**GLD8808I** Successfully finished traversing the tree on both the servers.

Explanation: The **ldapdiff** utility has successfully retrieved and compared entries on both LDAP servers that reside under the baseDn specified on the `-b` command line parameter.

Example: None.

System action: The program continues.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

---

**GLD8809I** Successfully finished traversing the tree on both the servers...

Explanation: The **ldapdiff** utility is retrieving and comparing entries on both LDAP servers residing under the baseDn specified on the `-b` command line parameter.

Example: None.

System action: The program continues.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

---

**GLD8810I** Either normal or operational attributes or both differ for this entry.

Explanation: The **ldapdiff** utility has detected a difference in the non-operational or operational attributes for the entry. The **ldapdiff** utility is using the `ibm-entryCheckSum` and `ibm-entryCheckSumOp` attribute values, to quickly detect entry differences. The
**ibm-entryCheckSum** attribute value is a checksum value of non-operational attribute values while the **ibm-entryCheckSumOp** attribute value is a checksum value of the operational attribute values.

**Example:** None.

**System action:** The program continues.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** None.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD8811E** Unable to start subtree comparison.

**Explanation:** The ldapdiff utility is unable to start subtree comparison on both LDAP servers for one of the following reasons:

- The **baseDn** specified on the `-b` command line parameter does not have valid DN syntax.
- The **baseDn** specified on the `-b` command line parameter cannot be found on both the supplier and consumer servers.
- The encryption settings of the supplier and consumer servers cannot be retrieved.

**Example:** None.

**System action:** The program ends.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Verify that the **baseDn** that is specified on the `-b` command line parameter is valid and exists on both LDAP servers. Ensure that the encryption settings for both servers can be retrieved. Then restart the program.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD8812I** Exceeded the specified number of non-matching entries.

**Explanation:** The ldapdiff utility has encountered the maximum number of non-matching entries between the LDAP servers being compared. The number of non-matching entries exceeds the number specified on the `-C` command line parameter.

**Example:** None.

**System action:** The program ends.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Verify that the maximum number of entry mismatches allowed by the ldapdiff utility is correct. Consider increasing the number of entry mismatches allowed or removing the `-C` command line parameter. Then restart the program.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

**GLD8813E** Exception: `exception_text`

**Explanation:** The ldapdiff utility encountered an exception while performing the requested task. This message is usually accompanied by another message indicating the operation that resulted in this exception.

**In the message text:**

`exception_text`

**Exception text**

**Example:** None.

**System action:** The program ends.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Use the information in this message and other messages issued to correct the problem. If additional information is needed to solve the problem, specify `-d ALL` on the command line. Then restart the program. If the problem persists, contact the service representative.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.
GLD8814I No attributes returned for name entry.
Explanation: The ldapdiff utility is unable to retrieve attribute values indicating encryption settings or password policy for the entry indicated in the message.
In the message text:
name Entry distinguished name
Example: None.
System action: The program continues without taking into account the encryption settings or password policy established on the LDAP server.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.
GLD8815E Failed to determine the server version for one of the servers.
Explanation: The ldapdiff utility is unable to determine the LDAP server version because the ibmdirver attribute value does not exist in the rootDSE entry. The ldapdiff utility only properly works with the IBM Tivoli Directory Servers on z/OS and other platforms.
Example: None.
System action: The program continues but the results are unexpected.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.
GLD8816E Cannot open the specified file for LDIF output generation. LDIF file will not be created.
Explanation: The ldapdiff utility encountered an error while attempting to open the output LDIF file specified on the -L command line parameter for writing.
Example: None.
System action: The program continues with the differences between the LDAP servers being written to standard output.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD8818E Exception occurred while closing the LDIF file: exception_text

Explanation: The ldapdiff utility encountered an exception when closing the output LDIF file specified on the -L command line parameter. The exception might have occurred for one of the following reasons:
- The output LDIF file is already closed.
- The output LDIF file does not exist at the specified location.
- The internal file reference is not valid.
- The user running the ldapdiff utility does not have the appropriate permissions to access the output LDIF file.

In the message text:

exception_text

Example: None.
System action: The program continues.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: Correct the command line of the ldapdiff utility to specify a valid value for the command line parameter indicated in the message. Then restart the program.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD8820W The supplier and consumer servers have different encryption seed or salt values. The operation will take longer.

Explanation: The ldapdiff utility has detected that the supplier and consumer servers have different encryption seed or salt values. This message is generally only issued when the ldapdiff utility is targeting a supplier or consumer server that is running against a z/OS IBM Tivoli Directory Server. The comparison operation is faster if the supplier and consumer servers have the same encryption seed and salt values.

Example: None.
System action: The program continues.
Operator response: None.
User response: None.
System programmer response: None.
Administrator response: None.
Problem determination: Not applicable.
Source: LDAP
Module: None.
Routing code: None.
Descriptor code: None.
Automation: Not applicable.

GLD8819E Missing arguments: Value not specified for option.

Explanation: The ldapdiff utility has detected that a value has not been specified for the command line parameter specified in the message.

In the message text:
GLD8821E  Missing required argument; refer to the usage description for valid syntax.

Explanation: The ldapdiff utility has detected one of the following required parameters are missing from the command line:

- `-ch host`
- `-sh host`
- `-b baseDn` or `-S`

Example: None.

System action: The program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Verify the ldapdiff command line has the required parameters specified. Then restart the program.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD8823E  Exception occurred while deleting DN from LDIF file: exception_text

Explanation: The ldapdiff utility encountered an exception while writing the entry to the output LDIF file. The exception that occurred while writing to the output LDIF file is indicated in the message.

In the message text:

exception_text

Exception text

Example: None.

System action: The program continues but the entry is not written to the output LDIF file.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the exception text indicated in the message to correct the problem. Verify the output LDIF file exists and the user has the appropriate permissions to write to the file. If additional information is needed to solve the problem, specify `-d ALL` on the command line. Then restart the program.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD8822E  Exception occurred while adding DN to LDIF file: exception_text

Explanation: The ldapdiff utility encountered an exception while writing the entry to the output LDIF file. The exception that occurred while writing to the output LDIF file is indicated in the message.

In the message text:

exception_text

Exception text

Example: None.

System action: The program continues but the entry is not written to the output LDIF file.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the exception text indicated in the message to correct the problem. Verify the output LDIF file exists and the user has the appropriate permissions to write to the file. If additional information is needed to solve the problem, specify `-d ALL` on the command line. Then restart the program.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD8824E  Exception occurred while modifying DN in LDIF file: exception_text

Explanation: The ldapdiff utility encountered an exception while writing the entry to the output LDIF file. The exception that occurred while writing to the output LDIF file is indicated in the message.

In the message text:

exception_text

Exception text

Example: None.

System action: The program continues but the entry is not written to the output LDIF file.

Operator response: None.
Chapter 30. LDAP server messages
GLD8828E Error occurred while reading rootDSE attributes: exception_text

Explanation: The ldapdiff utility encountered an internal error while reading and parsing the rootDSE entry attribute values.

In the message text:
exception_text
Exception text

Example: None.

System action: The program continues by using a default distinguished name (DN) of cn=schema for schema comparison on the targeted LDAP servers.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD8830E No subschemasubentry found in rootDSE.

Explanation: The ldapdiff utility is unable to find the subschemasubentry attribute on the search of the rootDSE entry. The subschemasubentry attribute specifies the distinguished name of the schema entry. Generally, this error can only occur when targeting the ldapdiff utility against a non-z/OS IBM Tivoli Directory Server.

Example: None.

System action: The program continues schema comparison by using the default distinguished name (DN) of cn=schema on the targeted LDAP server.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Define a subschemasubentry attribute value on the non-z/OS IBM Tivoli Directory Server. If additional information is needed to solve the problem, specify -d ALL on the command line. Then restart the program. If the problem persists, contact the service representative.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD8831E An exception occurred during search: exception_text.

Explanation: The ldapdiff utility encountered an internal error while performing a search operation on the LDAP server. The exception is indicated in the message.

In the message text:
exception_text
Exception text

Example: None.

System action: The program continues.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the exception text indicated in the message to correct the problem. Ensure that the targeted LDAP servers are running when using the ldapdiff utility. Verify that TCP/IP communication is working properly between the utility and each targeted LDAP server. Ensure that the bound user has the necessary permissions to access the search query results. If additional information is needed to solve the problem, specify -d ALL on the command line. Then restart the program. If the problem persists, contact the service representative.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.
The ldapdiff utility encountered an error while parsing through LDAP search results. The exception is indicated in the message.

In the message text:

```
exception_text
```

Example: None.

System action: The program continues.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the exception text indicated in the message to correct the problem. Verify that the user has enough permissions to set the environment settings or the system property settings. If additional information is needed to solve the problem, specify -d ALL on the command line. Then restart the program. If the problem persists, contact the service representative.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

Error occurred while setting environment variables: exception_text.

The ldapdiff utility encountered an error while initializing the environment for a connection to the LDAP server. The exception is indicated in the message.

In the message text:

```
exception_text
```

Example: None.

System action: The program continues without traversing the subtree on the consumer server. If the -F command line parameter is specified, any entries that exist on the supplier server in that subtree are added to the consumer server.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

The specified base DN is not found on the consumer server.

The ldapdiff utility was unable to find the baseDn on the consumer server. The baseDn was specified on the -b command line parameter of the ldapdiff utility.

Example: None.

System action: The program continues without traversing the subtree on the consumer server. If the -F command line parameter is specified, any entries that exist on the supplier server in that subtree are added to the consumer server.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

The specified base DN is not found on the supplier server.

The ldapdiff utility was unable to find the baseDn on the supplier server. The baseDn was specified on the -b command line parameter of the ldapdiff utility.

Example: None.

System action: The program continues without traversing the subtree on the supplier server. If the -F command line parameter is specified, any entries that exist on the consumer server in that subtree are deleted.
GLD8839W The supplier and consumer servers have different encryption settings. The operation may take longer.

Explanation: The ldapdiff utility performed a search of the cn=configuration subtree to obtain the ibm-slapdPwEncryption attribute value. This search is only performed when the ldapdiff utility is used with a non-z/OS IBM Tivoli Directory Server. The ldapdiff utility determined the supplier and consumer servers have different encryption settings so the comparison operation may take a longer period of time. The operation is faster if the supplier and consumer servers have the same encryption settings.

Example: None.

System action: The program continues.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD8840E Error while retrieving attributes of DN: exception_text.

Explanation: The ldapdiff utility encountered an internal error while adding the attributes to an internal hash table. The exception that occurred is indicated in the message.

In the message text:

exception_text

Exception text

Example: None.

System action: The program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD8841E Exception occurred while parsing "name" DN: exception_text.

Explanation: The ldapdiff utility encountered an exception while parsing the string representation of a distinguished name (DN). The exception that occurred is indicated in the message.

In the message text:

name

String representation of a DN

exception_text

Exception text

Example: None.

System action: If the distinguished name that is being parsed is the baseDn specified on the -b command line parameter, the program ends. If other distinguished names are being parsed, the program continues.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the exception text indicated in the message to correct the problem. Ensure the syntax of the DN being parsed is correct. Verify that a valid baseDn is specified on the -b command line parameter. If additional information is needed to solve the problem, specify -d ALL on the command line. Then restart the program. If the problem persists, contact the service representative.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.
GLD8842E Exception occurred while fetching attributes from server: "exception_text".

Explanation: The ldapdiff utility encountered an exception while fetching attributes from the LDAP server. The exception is indicated in the message.

In the message text:

exception_text

Exception text

Example: None.

System action: The program continues.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Verify that the targeted LDAP servers are still running. Use the information in this message and other messages to correct the problem. If additional information is needed to solve the problem, specify -d ALL on the command line. Then restart the program. If the problem persists, contact the service representative.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD8843E Traverse exception occurred: "exception_text".

Explanation: The ldapdiff utility encountered an error while traversing the LDAP servers. The exception is indicated in the message. The error might have occurred for one of the following reasons:

- There is a problem performing a search operation on the LDAP server. Verify that the search request controls are valid.
- There is a problem traversing the search entry results from the LDAP server.
- There is a generic error traversing the LDAP server entries.

In the message text:

exception_text

Exception text

Example: None.

System action: The program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Specify a valid number for the command line parameter indicated in the message. Then restart the program.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.
GLD8846E  Missing JSSE package for SSL connection.

Explanation: The installed version of Java is missing the Java secure socket extension (JSSE) package or the JSEE settings are not correct.

Example: None.

System action: The program continues but connecting with the proper JSSE settings is not successful.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Verify the installed version of Java has the JSSE jar file installed. If additional information is needed to solve the problem, specify -d ALL on the command line. Then restart the program. If the problem persists, contact the service representative.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD8847E  Specified SASL mechanism is not available for server name.

Explanation: The ldapdiff utility encountered an error because the LDAP server does not support the SASL authentication mechanism specified. This message is usually accompanied by another message indicating the exact cause of the error.

In the message text:

name  Server name

Example: None.

System action: The program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Verify the bindDn specified on the -cD command line parameter has the appropriate permissions to add, delete, or modify entries on the consumer server. Use the information in the accompanying message to correct the problem. If additional information is needed to solve the problem, specify -d ALL on the command line. Then restart the program. If the problem persists, contact the service representative.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD8848E  Unable to fix entry

Explanation: The ldapdiff utility encountered an error while attempting to fix an entry on the consumer server because the bindDn specified on the -cD command line parameter does not have the appropriate permissions. This message is usually accompanied by another message indicating the exact cause of the error.

Example: None.

System action: The program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Verify the bindDn specified on the -cD command line parameter has the appropriate permissions to add, delete, or modify entries on the consumer server. Use the information in the accompanying message to correct the problem. If additional information is needed to solve the problem, specify -d ALL on the command line. Then restart the program. If the problem persists, contact the service representative.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD8849E  Error occurred while retrieving filter ACL support of the server.

Explanation: The ldapdiff utility encountered an error while retrieving the rootDSE entry attributes from the LDAP server to determine if it supports filter ACL support. This message is usually accompanied by another message indicating the exact cause of the error.

Example: None.

System action: The program continues however filter ACL support is assumed not to be supported on the LDAP server.

Operator response: None.

User response: None.

System programmer response: None.
Administrator response: Use the information in the accompanying message to correct the problem. If additional information is needed to solve the problem, specify -d ALL on the command line. Then restart the program. If the problem persists, contact the service representative.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD8853I Finished comparing number entries...

Explanation: The ldapdiff utility has processed the number of entries indicated in the message.

In the message text:

number

Number of entries processed

Example: None.

System action: The program continues.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: None.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD8855E Supplier and consumer servers cannot be the same.

Explanation: The ldapdiff utility requires the host and port specified for the supplier and consumer servers be different. The ldapdiff utility does not support comparing and fixing entries on the same server.

Example: None.

System action: The program ends.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Correct the host and port values specified for the supplier and consumer servers. Then restart the program.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.

GLD8856E Internal error occurred.

Explanation: The ldapdiff utility encountered an internal error while removing controls from the previous request. This message is accompanied by another message indicating the exact cause of the error.

Example: None.

System action: The program continues without removing the control from the previous request.

Operator response: None.

User response: None.

System programmer response: None.

Administrator response: Use the information in the accompanying message to correct the problem. If additional information is needed to solve the problem, specify -d ALL on the command line. Then restart the program. If the problem persists, contact the service representative.

Problem determination: Not applicable.

Source: LDAP

Module: None.

Routing code: None.

Descriptor code: None.

Automation: Not applicable.
accompanying message to correct the problem. If additional information is needed to solve the problem, specify -d ALL on the command line. Then restart the program. If the problem persists, contact the service representative.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

GLD8857E Unable to display DNs only for non-matching entries. Servers must be able to calculate entry checksums in order to use this feature.

**Explanation:** If the -O command line parameter is specified on the ldapdiff utility, both servers must support entry checksum calculation with the ibm-entryCheckSum and the ibm-entryCheckSumOp attributes. The utility uses these attribute values to compare each entry on the supplier and consumer servers to quickly detect differences.

**Example:** None.

**System action:** The program ends.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Verify that both servers support the calculation of entry checksums by searching the rootDSE entry. Each LDAP server must have an ibm-supportedCapabilities attribute value of 1.3.18.0.2.32.56 on the rootDSE entry to use the -O command line parameter. If both LDAP servers do not have this support, remove the -O command line parameter. Then restart the program.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

GLD8858E A KeyStorePwd or TrustStorePwd is required.

**Explanation:** A KeyStorePwd or TrustStorePwd is required when the keyStoreType or trustStoreType options on the ldapdiff command line are set to JCEKS. The password must be specified on the -sP, -sY, -cP, or -cY command line parameters to gain access to the keyStore or trustStore.

**Example:** None.

**System action:** The program ends.

**Operator response:** None.

**User response:** None.

**System programmer response:** None.

**Administrator response:** Specify a password value for the keyStore or trustStore. Then restart the program.

**Problem determination:** Not applicable.

**Source:** LDAP

**Module:** None.

**Routing code:** None.

**Descriptor code:** None.

**Automation:** Not applicable.

---

GLD8859E Schema differences will not be automatically fixed.

**Explanation:** The ldapdiff utility does not support automatically fixing schema differences. The -F command line parameter is only supported for automatically fixing non-schema related entries.
GLD8860E  The "name" DN is not valid.

Explanation:  The ldapdiff utility is unable to search the supplier and consumer servers for the distinguished name indicated in the message because the name is not valid.

In the message text:

name  Entry distinguished name

Example:  None.

System action:  If the -S command line parameter is specified to compare the schema on both LDAP servers, the program continues; otherwise the program ends.

Operator response:  None.

User response:  None.

System programmer response:  None.

Administrator response:  If the distinguished name indicated in the message is the baseDn specified on the -b command line parameter, verify that it is valid and exists on both LDAP servers being compared. If the distinguished name is not the baseDn, check the entry on the LDAP server to verify that it is valid. Then restart the program.

Problem determination:  Not applicable.

Source:  LDAP

Module:  None.

Routing code:  None.

Descriptor code:  None.

Automation:  Not applicable.

GLD8862W  The -O option overrides the -F and -L options.

Explanation:  The -O command line parameter was specified with either the -F or -L parameters on the ldapdiff utility command line. When this occurs, the distinguished names (DNs) of entries that differ between the LDAP servers are only displayed to standard output. The differences are not fixed and are not written to the output LDIF file specified on the -L command line parameter.

Example:  None.

System action:  The program continues but entry differences are not fixed and are not written to the output LDIF file specified on the -L command line parameter.

Operator response:  None.

User response:  None.

System programmer response:  None.

Administrator response:  If entry differences need to be fixed or written to the output LDIF file, remove the -O command line parameter. Then restart the program.

Problem determination:  Not applicable.

Source:  LDAP

Module:  None.

Routing code:  None.

Descriptor code:  None.

Automation:  Not applicable.

GLD8863I  LDIF output will be written to filename.

Explanation:  Since the -L command line parameter was specified, the entry differences between the LDAP servers are written to the output LDIF file that is indicated in the message.

In the message text:

filename  Output LDIF file

Example:  None.

System action:  The program continues.

Operator response:  None.

User response:  None.

System programmer response:  None.

Administrator response:  None.

Problem determination:  Not applicable.
GLD8864I Schema LDIF output will be written to filename.

Explanation: Since the -S and -L command line parameters were specified together on the ldapdiff utility, the schema differences between the LDAP servers are written to the schema output LDIF file that is indicated in the message.

In the message text:

filename
Schema output LDIF file
Appendix A. Initial LDAP server schema

This appendix shows the initial schema established when the LDAP server is first started. The initial schema is always part of the LDAP server schema and the elements in the initial schema cannot be deleted. With several exceptions, the initial schema cannot be modified. See Updating the schema for more information.

```
cn=schema
cn=schema
objectclass=ibmSubschema
objectclass=subentry
objectclass=subschema
objectclass=top
subtreespecification=NULL

ldapsyntaxes=( 1.3.10.0.2.8.1 DESC 'IBM attribute type description' )
ldapsyntaxes=( 1.3.10.0.2.8.3 DESC 'IBM entry GUID' )
ldapsyntaxes=( 1.3.6.1.1.4 DESC 'Attribute type description' )
ldapsyntaxes=( 1.3.6.1.1.4 DESC 'Binary' )
ldapsyntaxes=( 1.3.6.1.1.4 DESC 'Boolean' )
ldapsyntaxes=( 1.3.6.1.14 DESC 'Distinguished name' )
ldapsyntaxes=( 1.3.6.1.14 DESC 'Directory string' )
ldapsyntaxes=( 1.3.6.1.14 DESC 'DIT content rule description' )
ldapsyntaxes=( 1.3.6.1.14 DESC 'DIT structure rule description' )
ldapsyntaxes=( 1.3.6.1.14 DESC 'generalized time' )
ldapsyntaxes=( 1.3.6.1.14 DESC 'IA5 string' )
ldapsyntaxes=( 1.3.6.1.14 DESC 'integer' )
ldapsyntaxes=( 1.3.6.1.14 DESC 'Matching rule description' )
ldapsyntaxes=( 1.3.6.1.14 DESC 'matching rule use description' )
ldapsyntaxes=( 1.3.6.1.14 DESC 'Name form description' )
ldapsyntaxes=( 1.3.6.1.14 DESC 'Object class description' )
ldapsyntaxes=( 1.3.6.1.14 DESC 'Object identifier' )
ldapsyntaxes=( 1.3.6.1.14 DESC 'object identifier' )
ldapsyntaxes=( 1.3.6.1.14 DESC 'Telephone number' )
ldapsyntaxes=( 1.3.6.1.14 DESC 'UTC time' )
ldapsyntaxes=( 1.3.6.1.14 DESC 'LDAP syntax description' )
ldapsyntaxes=( 1.3.6.1.14 DESC 'Substring assertion' )

matchingrules=( 1.3.6.1.1.4.1.1466.109.114.1 NAME ( 'caseIgnoreIA5Match' )
MATCHINGRULESDESC 'Version of the LDAP Server implementation' EQUALITY caseExactMatch SINGLE-VALUE NO-USER-MODIFICATION SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 USAGE dSAOperation )

attributeTypes=( 0.9.2342.19200300.100.1.1 NAME ( 'uid' )
DESC 'User shortname or userid'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 USAGE userApplications )
attributeTypes=( 0.9.2342.19200300.100.1.23 NAME ( 'lastmodifiedtime' ) SINGLE-VALUE
SYNTAX 1.3.6.1.4.1.1466.115.121.1.24 USAGE userApplications )
attributeTypes=( 0.9.2342.19200300.100.1.24 NAME ( 'lastmodifiedby' ) SINGLE-VALUE
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 USAGE userApplications )
attributeTypes=( 1.2.840.113556.1.4.77 NAME ( 'maxTicketAge' )
DESC 'Value defining the maximum lifetime of a user ticket'
SINGLE-VALUE SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 USAGE userApplications )
attributeTypes=( 1.2.840.113556.1.4.66 NAME ( 'userPrincipalName' )
DESC 'Primary security identity in the form <principal>@<realm>' EQUALITY caseExactMatch
SINGLE-VALUE SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 USAGE userApplications )
attributeTypes=( 1.3.6.1.1.4 NAME ( 'vendorName' )
DESC 'Name of the company that implemented the LDAP server' EQUALITY caseExactMatch
SINGLE-VALUE NO-USER-MODIFICATION SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 USAGE dSAOperation )
attributeTypes=( 1.3.6.1.1.5 NAME ( 'vendorVersion' )
DESC 'Version of the LDAP Server implementation' EQUALITY caseExactMatch SINGLE-VALUE
```
Initial LDAP server schema
NO-USER-MODIFICATION SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 USAGE dSAOperation )
attributetypes=( 1.3.6.1.4.1.1466.101.120.5 NAME ( 'namingContexts' ) DESC 'LDAP server naming contexts'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 USAGE dSAOperation )
attributetypes=( 1.3.6.1.4.1.1466.101.120.6 NAME ( 'altServer' ) DESC 'Alternate LDAP server'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE dSAOperation )
attributetypes=( 1.3.6.1.4.1.1466.101.120.7 NAME ( 'supportedExtension' )
DESC 'Extensions supported by this server' SYNTAX 1.3.6.1.4.1.1466.115.121.1.38
USAGE dSAOperation )
attributetypes=( 1.3.6.1.4.1.1466.101.120.13 NAME ( 'supportedControl' )
DESC 'Controls supported by this server' SYNTAX 1.3.6.1.4.1.1466.115.121.1.38
USAGE dSAOperation )
attributetypes=( 1.3.6.1.4.1.1466.101.120.14 NAME ( 'supportedSASLMechanisms' )
DESC 'SASL mechanisms supported by this server'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 USAGE dSAOperation )
attributetypes=( 1.3.6.1.4.1.1466.101.120.15 NAME ( 'supportedLDAPVersion' )
DESC 'LDAP protocol versions supported by this server'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 USAGE dSAOperation )
attributetypes=( 1.3.6.1.4.1.1466.101.120.16 NAME ( 'ldapSyntaxes' ) DESC 'LDAP syntaxes'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.54 USAGE directoryOperation )
attributetypes=( 1.3.18.0.2.4.155 NAME ( 'secretKey' )
DESC 'Attribute is always stored in encrypted form' SINGLE-VALUE SYNTAX 1.3.6.1.4.1.1466.115.121.1.5
USAGE userApplications )
attributetypes=( 1.3.18.0.2.4.185 NAME ( 'sysplex' ) DESC 'Identifies the name of an OS/390 sysplex'
SINGLE-VALUE SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 USAGE userApplications )
attributetypes=( 1.3.18.0.2.4.186 NAME ( 'profileType' )
DESC 'Identifies the name of a OS/390 Security Server profile' SINGLE-VALUE
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 USAGE userApplications )
attributetypes=( 1.3.18.0.2.4.187 NAME ( 'racfid' )
DESC 'Identifies the name of a OS/390 Security Server userid or groupid' SINGLE-VALUE
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE userApplications )
attributetypes=( 1.3.18.0.2.4.188 NAME ( 'racfAuthorizationDate' )
DESC 'Date is displayed in yy.ddd format' SINGLE-VALUE SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
USAGE userApplications )
attributetypes=( 1.3.18.0.2.4.189 NAME ( 'racfOwner' ) SINGLE-VALUE
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE userApplications )
attributetypes=( 1.3.18.0.2.4.190 NAME ( 'racfInstallationData' ) SINGLE-VALUE
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE userApplications )
attributetypes=( 1.3.18.0.2.4.191 NAME ( 'racfDatasetModel' ) SINGLE-VALUE
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE userApplications )
attributetypes=( 1.3.18.0.2.4.192 NAME ( 'racfSuperiorGroup' ) SINGLE-VALUE
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE userApplications )
attributetypes=( 1.3.18.0.2.4.193 NAME ( 'racfGroupNoTermUAC' ) SINGLE-VALUE
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE userApplications )
attributetypes=( 1.3.18.0.2.4.194 NAME ( 'racfSubGroupName' ) SINGLE-VALUE
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE userApplications )
attributetypes=( 1.3.18.0.2.4.195 NAME ( 'racfGroupUserids' ) SINGLE-VALUE
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE userApplications )
attributetypes=( 1.3.18.0.2.4.197 NAME ( 'racfAttributes' )
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE userApplications )
attributetypes=( 1.3.18.0.2.4.198 NAME ( 'racfPassword' ) SINGLE-VALUE
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE userApplications )
attributetypes=( 1.3.18.0.2.4.199 NAME ( 'racfPasswordInterval' ) SINGLE-VALUE
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE userApplications )
attributetypes=( 1.3.18.0.2.4.200 NAME ( 'racfPasswordChangeDate' ) SINGLE-VALUE
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE userApplications )
attributetypes=( 1.3.18.0.2.4.201 NAME ( 'racfProgrammerName' ) SINGLE-VALUE
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE userApplications )
attributetypes=( 1.3.18.0.2.4.202 NAME ( 'racfDefaultGroup' ) SINGLE-VALUE
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE userApplications )
attributetypes=( 1.3.18.0.2.4.203 NAME ( 'racfLastAccess' ) SINGLE-VALUE
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE userApplications )
attributetypes=( 1.3.18.0.2.4.204 NAME ( 'racfSecurityLevel' ) SINGLE-VALUE
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE userApplications )
attributetypes=( 1.3.18.0.2.4.205 NAME ( 'racfSecurityCategoryList' )
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE userApplications )
attributetypes=( 1.3.18.0.2.4.206 NAME ( 'racfRevokeDate' ) SINGLE-VALUE
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE userApplications )
attributetypes=( 1.3.18.0.2.4.207 NAME ( 'racfResumeDate' ) SINGLE-VALUE
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE userApplications )
attributetypes=( 1.3.18.0.2.4.208 NAME ( 'racfLogonDays' ) SINGLE-VALUE
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE userApplications )
attributetypes=( 1.3.18.0.2.4.209 NAME ( 'racfLogonTime' ) SINGLE-VALUE
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE userApplications )
attributetypes=( 1.3.18.0.2.4.210 NAME ( 'racfClassName' )
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE userApplications )
attributetypes=( 1.3.18.0.2.4.211 NAME ( 'racfConnectGroupName' ) SINGLE-VALUE
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE userApplications )
attributetypes=( 1.3.18.0.2.4.212 NAME ( 'racfConnectGroupAuthority' ) SINGLE-VALUE
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE userApplications )
attributetypes=( 1.3.18.0.2.4.213 NAME ( 'racfConnectGroupUACC' ) SINGLE-VALUE
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE userApplications )
attributetypes=( 1.3.18.0.2.4.214 NAME ( 'racfSecurityLabel' ) SINGLE-VALUE
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE userApplications )

722

z/OS V1R11.0 IBM Tivoli Directory Server Administration and Use for z/OS


Appendix A. Initial LDAP server schema
### Initial LDAP server schema

```plaintext
<table>
<thead>
<tr>
<th>Attribute Type</th>
<th>Description</th>
<th>Syntax</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.3.18.0.2.4.257</td>
<td><code>replicaUpdateTimeInterval</code></td>
<td>NAME ('replicaUpdateTimeInterval')</td>
<td>SINGLE-VALUE</td>
</tr>
<tr>
<td>1.3.18.0.2.4.258</td>
<td><code>replicaBindMethod</code></td>
<td>NAME ('replicaBindMethod')</td>
<td>DESC 'Specifies the replica bind method'</td>
</tr>
<tr>
<td>1.3.18.0.2.4.259</td>
<td><code>replicaPort</code></td>
<td>NAME ('replicaPort')</td>
<td>DESC 'Specifies the replica bind port'</td>
</tr>
<tr>
<td>1.3.18.0.2.4.260</td>
<td><code>replicaCredentials</code></td>
<td>NAME ('replicaCredentials')</td>
<td>DESC 'Specifies the replica bind credentials'</td>
</tr>
<tr>
<td>1.3.18.0.2.4.261</td>
<td><code>replicaBindDN</code></td>
<td>NAME ('replicaBindDN')</td>
<td>DESC 'Specifies the replica bind DN'</td>
</tr>
<tr>
<td>1.3.18.0.2.4.262</td>
<td><code>replicaHost</code></td>
<td>NAME ('replicaHost')</td>
<td>DESC 'Specifies the replica host name'</td>
</tr>
</tbody>
</table>
```

```
<table>
<thead>
<tr>
<th>Attribute Type</th>
<th>Description</th>
<th>Syntax</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.3.18.0.2.4.263</td>
<td><code>ownerPropagate</code></td>
<td>NAME ('ownerPropagate')</td>
<td>DESC 'Defines entry owner subtree propagation'</td>
</tr>
<tr>
<td>1.3.18.0.2.4.264</td>
<td><code>entryOwner</code></td>
<td>NAME ('entryOwner')</td>
<td>DESC 'Defines an entry owner'</td>
</tr>
<tr>
<td>1.3.18.0.2.4.265</td>
<td><code>aclSource</code></td>
<td>NAME ('aclSource')</td>
<td>DESC 'Source of the access list for an entry'</td>
</tr>
<tr>
<td>1.3.18.0.2.4.266</td>
<td><code>aclEntry</code></td>
<td>NAME ('aclEntry')</td>
<td>DESC 'Defines an access list entry'</td>
</tr>
<tr>
<td>1.3.18.0.2.4.267</td>
<td><code>aclObjectClass</code></td>
<td>NAME ('aclObjectClass')</td>
<td>DESC 'Source of the access list for an entry'</td>
</tr>
<tr>
<td>1.3.18.0.2.4.268</td>
<td><code>aclObjectOwner</code></td>
<td>NAME ('aclObjectOwner')</td>
<td>DESC 'Source of the access list subtree propagation'</td>
</tr>
<tr>
<td>1.3.18.0.2.4.269</td>
<td><code>aclSource</code></td>
<td>NAME ('aclSource')</td>
<td>DESC 'Source of the access list for an entry'</td>
</tr>
<tr>
<td>1.3.18.0.2.4.270</td>
<td><code>aclObjectEntry</code></td>
<td>NAME ('aclObjectEntry')</td>
<td>DESC 'Source of the access list subtree propagation'</td>
</tr>
<tr>
<td>1.3.18.0.2.4.271</td>
<td><code>aclPropagate</code></td>
<td>NAME ('aclPropagate')</td>
<td>DESC 'Defines access list subtree propagation'</td>
</tr>
<tr>
<td>1.3.18.0.2.4.272</td>
<td><code>aclObjectOwner</code></td>
<td>NAME ('aclObjectOwner')</td>
<td>DESC 'Source of the access list subtree propagation'</td>
</tr>
<tr>
<td>1.3.18.0.2.4.273</td>
<td><code>aclObjectEntry</code></td>
<td>NAME ('aclObjectEntry')</td>
<td>DESC 'Source of the access list subtree propagation'</td>
</tr>
</tbody>
</table>
```

```
<table>
<thead>
<tr>
<th>Attribute Type</th>
<th>Description</th>
<th>Syntax</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.3.18.0.2.4.274</td>
<td><code>racfDefaultConsoleName</code></td>
<td>NAME ('racfDefaultConsoleName')</td>
<td>SINGLE-VALUE</td>
</tr>
<tr>
<td>1.3.18.0.2.4.275</td>
<td><code>racfDCEPrincipal</code></td>
<td>NAME ('racfDCEPrincipal')</td>
<td>EQUALITY caseExactMatch</td>
</tr>
<tr>
<td>1.3.18.0.2.4.276</td>
<td><code>racfDCEHomeCell</code></td>
<td>NAME ('racfDCEHomeCell')</td>
<td>SINGLE-VALUE</td>
</tr>
<tr>
<td>1.3.18.0.2.4.277</td>
<td><code>racfDCEHomeCellUID</code></td>
<td>NAME ('racfDCEHomeCellUID')</td>
<td>SINGLE-VALUE</td>
</tr>
<tr>
<td>1.3.18.0.2.4.278</td>
<td><code>racfDCEAutoLogin</code></td>
<td>NAME ('racfDCEAutoLogin')</td>
<td>SINGLE-VALUE</td>
</tr>
<tr>
<td>1.3.18.0.2.4.279</td>
<td><code>racfDCEHomeCell</code></td>
<td>NAME ('racfDCEHomeCell')</td>
<td>SINGLE-VALUE</td>
</tr>
</tbody>
</table>
```

```
<table>
<thead>
<tr>
<th>Attribute Type</th>
<th>Description</th>
<th>Syntax</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.3.18.0.2.4.280</td>
<td><code>racfDCEInitialProgram</code></td>
<td>NAME ('racfDCEInitialProgram')</td>
<td>SINGLE-VALUE</td>
</tr>
<tr>
<td>1.3.18.0.2.4.281</td>
<td><code>racfDCEAutoLogin</code></td>
<td>NAME ('racfDCEAutoLogin')</td>
<td>SINGLE-VALUE</td>
</tr>
<tr>
<td>1.3.18.0.2.4.282</td>
<td><code>racfDCEAutoLogin</code></td>
<td>NAME ('racfDCEAutoLogin')</td>
<td>SINGLE-VALUE</td>
</tr>
<tr>
<td>1.3.18.0.2.4.283</td>
<td><code>racfDCEAutoLogin</code></td>
<td>NAME ('racfDCEAutoLogin')</td>
<td>SINGLE-VALUE</td>
</tr>
<tr>
<td>1.3.18.0.2.4.284</td>
<td><code>racfDCEAutoLogin</code></td>
<td>NAME ('racfDCEAutoLogin')</td>
<td>SINGLE-VALUE</td>
</tr>
<tr>
<td>1.3.18.0.2.4.285</td>
<td><code>racfDCEAutoLogin</code></td>
<td>NAME ('racfDCEAutoLogin')</td>
<td>SINGLE-VALUE</td>
</tr>
<tr>
<td>1.3.18.0.2.4.286</td>
<td><code>racfDCEAutoLogin</code></td>
<td>NAME ('racfDCEAutoLogin')</td>
<td>SINGLE-VALUE</td>
</tr>
<tr>
<td>1.3.18.0.2.4.287</td>
<td><code>racfDCEAutoLogin</code></td>
<td>NAME ('racfDCEAutoLogin')</td>
<td>SINGLE-VALUE</td>
</tr>
<tr>
<td>1.3.18.0.2.4.288</td>
<td><code>racfDCEAutoLogin</code></td>
<td>NAME ('racfDCEAutoLogin')</td>
<td>SINGLE-VALUE</td>
</tr>
<tr>
<td>1.3.18.0.2.4.289</td>
<td><code>racfDCEAutoLogin</code></td>
<td>NAME ('racfDCEAutoLogin')</td>
<td>SINGLE-VALUE</td>
</tr>
<tr>
<td>1.3.18.0.2.4.290</td>
<td><code>racfDCEAutoLogin</code></td>
<td>NAME ('racfDCEAutoLogin')</td>
<td>SINGLE-VALUE</td>
</tr>
<tr>
<td>1.3.18.0.2.4.291</td>
<td><code>racfDCEAutoLogin</code></td>
<td>NAME ('racfDCEAutoLogin')</td>
<td>SINGLE-VALUE</td>
</tr>
<tr>
<td>1.3.18.0.2.4.292</td>
<td><code>racfDCEAutoLogin</code></td>
<td>NAME ('racfDCEAutoLogin')</td>
<td>SINGLE-VALUE</td>
</tr>
<tr>
<td>1.3.18.0.2.4.293</td>
<td><code>racfDCEAutoLogin</code></td>
<td>NAME ('racfDCEAutoLogin')</td>
<td>SINGLE-VALUE</td>
</tr>
<tr>
<td>1.3.18.0.2.4.294</td>
<td><code>racfDCEAutoLogin</code></td>
<td>NAME ('racfDCEAutoLogin')</td>
<td>SINGLE-VALUE</td>
</tr>
<tr>
<td>1.3.18.0.2.4.295</td>
<td><code>racfDCEAutoLogin</code></td>
<td>NAME ('racfDCEAutoLogin')</td>
<td>SINGLE-VALUE</td>
</tr>
<tr>
<td>1.3.18.0.2.4.296</td>
<td><code>racfDCEAutoLogin</code></td>
<td>NAME ('racfDCEAutoLogin')</td>
<td>SINGLE-VALUE</td>
</tr>
<tr>
<td>1.3.18.0.2.4.297</td>
<td><code>racfDCEAutoLogin</code></td>
<td>NAME ('racfDCEAutoLogin')</td>
<td>SINGLE-VALUE</td>
</tr>
<tr>
<td>1.3.18.0.2.4.298</td>
<td><code>racfDCEAutoLogin</code></td>
<td>NAME ('racfDCEAutoLogin')</td>
<td>SINGLE-VALUE</td>
</tr>
<tr>
<td>1.3.18.0.2.4.299</td>
<td><code>racfDCEAutoLogin</code></td>
<td>NAME ('racfDCEAutoLogin')</td>
<td>SINGLE-VALUE</td>
</tr>
</tbody>
</table>
```
```
Appendix A. Initial LDAP server schema
Initial LDAP server schema

attributetypes=( 1.3.18.0.2.4.2239 NAME ( 'ibm-scheduleFriday' ) DESC 'DN of the entry defining the replication schedule for Friday' SINGLE-VALUE SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 USAGE userApplications )
attributetypes=( 1.3.18.0.2.4.2240 NAME ( 'ibm-replicationBatchStart' ) DESC 'Time to replicate accumulated changes in the form: <time> <change-id> <result code> <operation> <entry-dn>' EQUALITY caseIgnoreMatch SINGLE-VALUE NO-USER-MODIFICATION SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE directoryOperation )
attributetypes=( 1.3.18.0.2.4.2241 NAME ( 'ibm-replicationLastResultAdditional' ) DESC 'Provides any additional information returned by the consuming server in the message component of the LDAP result' EQUALITY caseExactIA5Match SINGLE-VALUE NO-USER-MODIFICATION SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE directoryOperation )
attributetypes=( 1.3.18.0.2.4.2242 NAME ( 'ibm-frontEndHost' ) DESC 'Indicates the last time the replication thread completed sending all of the pending entries.' SINGLE-VALUE NO-USER-MODIFICATION SYNTAX 1.3.6.1.4.1.1466.115.121.1.24 USAGE directoryOperation )
attributetypes=( 1.3.18.0.2.4.2328 NAME ( 'ibm-serverId' ) DESC 'Indicates last change id successfully replicated for a replication agreement' SINGLE-VALUE NO-USER-MODIFICATION SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 USAGE directoryOperation )
attributetypes=( 1.3.18.0.2.4.2329 NAME ( 'ibm-replicationServerIsMaster' ) DESC 'Indicates that a server assumes the role of a master for a given subtree' SINGLE-VALUE NO-USER-MODIFICATION SYNTAX 1.3.6.1.4.1.1466.115.121.1.17 USAGE userApplications )
attributetypes=( 1.3.18.0.2.4.2330 NAME ( 'ibm-replicationChangeLDIF' ) DESC 'Provides LDIF representation of the last failing operation' SINGLE-VALUE NO-USER-MODIFICATION SYNTAX 1.3.6.1.4.1.1466.115.121.1.5 USAGE directoryOperation )
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 caseExactIA5Match SINGLE-VALUE NO-USER-MODIFICATION SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE dSAOperation )
attributetypes=( 1.3.18.0.2.4.2332 NAME ( 'ibm replicationLastResultAdditional' ) DESC 'Provides any additional information returned by the consuming server in the message component of the LDAP result' EQUALITY caseExactIA5Match SINGLE-VALUE NO-USER-MODIFICATION SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE dSAOperation )
attributetypes=( 1.3.18.0.2.4.2333 NAME ( 'ibm-supportedReplicationModels' ) DESC 'Advertises in the Root DSE the OIDs of replication models supported by the server' EQUALITY caseExactIA5Match SINGLE-VALUE NO-USER-MODIFICATION SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE dSAOperation )
attributetypes=( 1.3.18.0.2.4.2334 NAME ( 'ibm-replicationLastChangeId' ) DESC 'Indicates last change id successfully replicated for a replication agreement' SINGLE-VALUE NO-USER-MODIFICATION SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 USAGE directoryOperation )
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.' SINGLE-VALUE NO-USER-MODIFICATION SYNTAX 1.3.6.1.4.1.1466.115.121.1.24 USAGE directoryOperation )
attributetypes=( 1.3.18.0.2.4.2336 NAME ( 'ibm-replicationLastResultState' ) 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 caseExactIA5Match SINGLE-VALUE NO-USER-MODIFICATION SYNTAX 1.3.6.1.4.1.1466.115.121.1.16 USAGE directoryOperation )
attributetypes=( 1.3.18.0.2.4.2337 NAME ( 'ibm-replicationPendingChanges' ) DESC 'Replication pending changes for this replication agreement' EQUALITY caseExactIA5Match SINGLE-VALUE NO-USER-MODIFICATION SYNTAX 1.3.6.1.4.1.1466.115.121.1.24 USAGE dSAOperation )
attributetypes=( 1.3.18.0.2.4.2338 NAME ( 'ibm-replicationLastActivationTime' ) DESC 'Indicates the last time the replication thread was activated' SINGLE-VALUE NO-USER-MODIFICATION SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 USAGE directoryOperation )
attributetypes=( 1.3.18.0.2.4.2339 NAME ( 'ibm-replicationNextTime' ) DESC 'Indicates next scheduled time for replication' SINGLE-VALUE NO-USER-MODIFICATION SYNTAX 1.3.6.1.4.1.1466.115.121.1.24 USAGE directoryOperation )
attributetypes=( 1.3.18.0.2.4.2340 NAME ( 'ibm-replicationLastResult' ) DESC 'Result of last attempted replication in the form: <time> <change-id> <result code> <operation> <entry-dn>' EQUALITY caseIgnoreMatch SINGLE-VALUE NO-USER-MODIFICATION SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 USAGE directoryOperation )
attributetypes=( 1.3.18.0.2.4.2341 NAME ( 'ibm-replicationBatchStart' ) DESC 'Time to replicate accumulated changes in the form of Thhmmss where hh is hours, mm is minutes and ss is seconds, using a 24 hour clock' EQUALITY caseExactIA5Match SINGLE-VALUE SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE userApplications )
attributetypes=( 1.3.18.0.2.4.2342 NAME ( 'ibm-replicaKeyfile' ) DESC 'Name of key database file on the supplying server with the certificate of the consuming server and the supplier' EQUALITY caseIgnoreMatch SINGLE-VALUE SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 USAGE userApplications )
attributetypes=( 1.3.18.0.2.4.2343 NAME ( 'ibm-replicaKeyLabel' ) DESC 'Label for certificate containing private key for supplying server' EQUALITY caseIgnoreMatch SINGLE-VALUE SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 USAGE userApplications )
attributetypes=( 1.3.18.0.2.4.2344 NAME ( 'ibm-replicationNextTime' ) DESC 'Indicates next scheduled time for replication' SINGLE-VALUE NO-USER-MODIFICATION SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 USAGE userApplications )
attributetypes=( 1.3.18.0.2.4.2345 NAME ( 'ibm-replicationTimesUTC' ) DESC 'Scheduled times are GMT if TRUE or local time zone if FALSE' SINGLE-VALUE NO-USER-MODIFICATION SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 USAGE userApplications )
attributetypes=( 1.3.18.0.2.4.2346 NAME ( 'ibm-scheduleTuesday' ) DESC 'DN of the entry defining the replication schedule for Tuesday' SINGLE-VALUE NO-USER-MODIFICATION SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 USAGE userApplications )

Appendix A. Initial LDAP server schema
Initial LDAP server schema

Capabilities will not be replicated under the agreement containing this attribute. EQUALITY caseIgnoreMatch

Syntax: 1.3.6.1.4.1.1466.115.121.1.15 USAGE userApplications

Attribute types:
- 1.3.18.0.2.4.2486 NAME ("ibm-slapdMaxPendingChangesDisplayed") DESC 'Maximum number of pending replication updates to be displayed for any given replication agreement on a supplier server.'
- 1.3.18.0.2.4.2494 NAME ("ibm-replDailyScheduleName")
  DESC 'Naming attribute and descriptive name for an ibm-replDailySchedule object.' EQUALITY caseIgnoreMatch
  SINGLE-VALUE Syntax: 1.3.6.1.4.1.1466.115.121.1.15 USAGE userApplications
- 1.3.18.0.2.4.2497 NAME ("ibm-replWeeklyScheduleName")
  DESC 'Naming attribute and descriptive name for an ibm-replWeeklySchedule object.' EQUALITY caseIgnoreMatch
  SINGLE-VALUE Syntax: 1.3.6.1.4.1.1466.115.121.1.15 USAGE userApplications
- 1.3.18.0.2.4.2498 NAME ("ibm-replIsQuiesced")
  DESC 'Indicates whether the replicated subtree containing this attribute is quiesced on this server.'
  SINGLE-VALUE NO-USER-MODIFICATION Syntax: 1.3.6.1.4.1.1466.115.121.1.17 USAGE dsOperation
- 1.3.18.0.2.4.2500 NAME ("ibm-slapdMigrationInfo")
  DESC 'Information used to control migration of a component.' EQUALITY caseIgnoreMatch
  Syntax: 1.3.6.1.4.1.1466.115.121.1.15 USAGE userApplications
- 1.3.18.0.2.4.3081 NAME ("racOmvsSharedMemoryMaximum")
  DESC 'Represents the SHMEMMAX(shared-memory-size) field of the RACF user OMVS segment' SINGLE-VALUE
  Syntax: 1.3.6.1.4.1.1466.115.121.1.16 USAGE userApplications
- 1.3.18.0.2.4.3080 NAME ("racOmvsMemoryLimit")
  DESC 'Represents the MEMLIMIT(non-shared-memory-size) field of the RACF user OMVS segment' SINGLE-VALUE
  Syntax: 1.3.6.1.4.1.1466.115.121.1.16 USAGE userApplications
- 1.3.18.0.2.4.3091 NAME ("racfPasswordEnvelope")
  DESC 'Represents the MEMLIMIT(non-shared-memory-size) field of the RACF user OMVS segment' SINGLE-VALUE
  Syntax: 1.3.6.1.4.1.1466.115.121.1.16 USAGE userApplications
- 1.3.18.0.2.4.3090 NAME ("racfOmvsMemoryLimit")
  DESC 'Represents the MEMLIMIT(non-shared-memory-size) field of the RACF user OMVS segment' SINGLE-VALUE
  Syntax: 1.3.6.1.4.1.1466.115.121.1.16 USAGE userApplications
- 1.3.18.0.2.4.3094 NAME ("firstChangeNumber")
  DESC 'Change number for the earliest entry in the server change log' EQUALITY integerMatch
  SINGLE-VALUE NO-USER-MODIFICATION Syntax: 1.3.6.1.4.1.1466.115.121.1.27 USAGE dsOperation
- 1.3.18.0.2.4.3095 NAME ("lastChangeNumber")
  DESC 'Change number for the latest entry in the server change log' EQUALITY integerMatch SINGLE-VALUE
  NO-USER-MODIFICATION Syntax: 1.3.6.1.4.1.1466.115.121.1.27 USAGE dsOperation
- 1.3.18.0.2.4.3097 NAME ("ldapServiceName")
  DESC 'LDAP service name for this server as hostRealm' SINGLE-VALUE NO-USER-MODIFICATION
  Syntax: 1.3.6.1.4.1.1466.115.121.1.15 USAGE dsOperation
- 1.3.18.0.2.4.3098 NAME ("ibmdirDirectoryVersion")
  DESC 'Version of this directory server' SINGLE-VALUE NO-USER-MODIFICATION
  Syntax: 1.3.6.1.4.1.1466.115.121.1.15 USAGE dsOperation
- 1.3.18.0.2.4.3128 NAME ("ibm-slapdLog")
  DESC 'Log path and file name. On Windows, forward slashes are allowed, and a leading slash not preceded by a drive letter is assumed to be rooted at the install directory (i.e.: /tmp/bulksend.errors = D:\Program Files\IBM\ldap\tmp\bulksend.errors)'.
  EQUALITY caseExactMatch SINGLE-VALUE Syntax: 1.3.6.1.4.1.1466.115.121.1.15 USAGE userApplications
- 1.3.18.0.2.4.3129 NAME ("ibm-slapdLogMaxArchives")
  DESC 'The maximum number of archived logs where 0 means no archive file will be kept and -1 means an unlimited number of archive files will be kept.'
  EQUALITY integerMatch SINGLE-VALUE Syntax: 1.3.6.1.4.1.1466.115.121.1.27 USAGE userApplications
- 1.3.18.0.2.4.3130 NAME ("ibm-slapdLogOptions")
  DESC 'Any log options that the log uses, for example, log level or mask.' EQUALITY caseIgnoreMatch
  SINGLE-VALUE Syntax: 1.3.6.1.4.1.1466.115.121.1.15 USAGE userApplications
- 1.3.18.0.2.4.3131 NAME ("ibm-slapdLogArchivePath")
  DESC 'Path for archived files. On Windows, forward slashes are allowed, and a leading slash not preceded by a drive letter is assumed to be rooted at the install directory (i.e.: /tmp = D:\Program Files\IBM\ldap\tmp)'.
  EQUALITY caseExactMatch SINGLE-VALUE Syntax: 1.3.6.1.4.1.1466.115.121.1.15 USAGE userApplications
- 1.3.18.0.2.4.3132 NAME ("ibm-slapdLogArchivePath")
  DESC 'Method used by a server to replicate its single thread, multiple threads and connections. The value is not dynamic.' SINGLE-VALUE
- 1.3.18.0.2.4.3133 NAME ("ibm-slapdLogMaxEntries")
  DESC 'Maximum number of bytes that an entry can contain and still be resent to a target server as a result of replication conflict resolution. This value is dynamic.' EQUALITY integerMatch SINGLE-VALUE
  Syntax: 1.3.6.1.4.1.1466.115.121.1.27 USAGE userApplications
- 1.3.18.0.2.4.3134 NAME ("ibm-slapdLogMaxErrors")
  DESC 'Method used by a server to replicate its single thread, multiple threads and connections. The value is not dynamic.' SINGLE-VALUE
  Syntax: 1.3.6.1.4.1.1466.115.121.1.27 USAGE userApplications
- 1.3.18.0.2.4.3135 NAME ("ibm-slapdLogMaxErrors")
  DESC 'Limit to allowed errors per replication agreement, 0=unlimited. The value is dynamic.' SINGLE-VALUE
  Syntax: 1.3.6.1.4.1.1466.115.121.1.27 USAGE userApplications
Appendix A. Initial LDAP server schema
### Initial LDAP server schema

<table>
<thead>
<tr>
<th>Attribute Type</th>
<th>Description</th>
<th>Syntax</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ibm-slapdLogCachePath</code></td>
<td>Log path for an event formatted log. 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., <code>/tmp</code> or <code>D:\Program Files\IBM\ldap\V6.1\tmp</code>).</td>
<td><code>1.3.6.1.4.1.1466.115.121.1.15</code></td>
<td><code>USAGE</code> userApplications</td>
</tr>
<tr>
<td><code>ibm-slapdAuditOperation</code></td>
<td>Specifies which in event format the users want the ITDS log records to be converted to the specified event format. For example, if the attribute is set to BIND, then audit records related only to the bind operation will be converted to specified event format.</td>
<td><code>1.3.6.1.4.1.1466.115.121.1.15</code></td>
<td><code>USAGE</code> userApplications</td>
</tr>
<tr>
<td><code>ibm-slapdLogMgmtFrequency</code></td>
<td>Specifies the time interval between two cycles of the log management activity.</td>
<td><code>1.3.6.1.4.1.1466.115.121.1.15</code></td>
<td><code>USAGE</code> userApplications</td>
</tr>
<tr>
<td><code>ibm-slapdLogMgmtStartTime</code></td>
<td>Specifies the start date and time for the log management activity. The format is YYYYMMDDHHMM.</td>
<td><code>1.3.6.1.4.1.1466.115.121.1.15</code></td>
<td><code>USAGE</code> userApplications</td>
</tr>
<tr>
<td><code>ibm-slapdLogMgmtEndTime</code></td>
<td>Specifies the end date and time for the log management activity.</td>
<td><code>1.3.6.1.4.1.1466.115.121.1.15</code></td>
<td><code>USAGE</code> userApplications</td>
</tr>
<tr>
<td><code>ibm-slapdLogMgmtStartTime</code></td>
<td>Specifies the start date and time for the log management activity.</td>
<td><code>1.3.6.1.4.1.1466.115.121.1.15</code></td>
<td><code>USAGE</code> userApplications</td>
</tr>
<tr>
<td><code>ibm-slapdLogMgmtEndTime</code></td>
<td>Specifies the end date and time for the log management activity.</td>
<td><code>1.3.6.1.4.1.1466.115.121.1.15</code></td>
<td><code>USAGE</code> userApplications</td>
</tr>
</tbody>
</table>

**Additional Attributes**

- `racfCfdefMaxValue` (Single-Value)
- `racfCfdefMixed` (Single-Value)
- `racfAccessControl` (Single-Value)
- `racfHavePassPhraseEnvelope` (Single-Value)
- `racfPassPhraseEnvelope` (Single-Value)
- `racfStdataTrace` (Single-Value)
- `racfStdataUser` (Single-Value)
- `racfUacc` (Single-Value)
- `racfCfdefMaxValue` (Single-Value)
- `racfCfdefMixed` (Single-Value)
- `racfAccessControl` (Single-Value)
### Initial LDAP server schema

<table>
<thead>
<tr>
<th>AttributeType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.3.6.1.4.1.1466.115.121.1.26</td>
<td>Usage userApplications</td>
</tr>
<tr>
<td>racfApplData</td>
<td>Text string associated with the resource profile</td>
</tr>
<tr>
<td>racfMemberList</td>
<td>Name of member that RACF is to add to the resource profile</td>
</tr>
<tr>
<td>racfLogOptionsSuccesses</td>
<td>Level of security checking when conversations are established with the protected LU</td>
</tr>
<tr>
<td>racfSessionLock</td>
<td>Mark the resource profile as locked</td>
</tr>
<tr>
<td>racfPasswordEncrypted</td>
<td>Encrypt the key value</td>
</tr>
<tr>
<td>racfSessionSessKey</td>
<td>Mask the key value</td>
</tr>
<tr>
<td>racfApplData</td>
<td>Represent the AUTOMATIC field in the RACF TAPEVOL class</td>
</tr>
<tr>
<td>racfStdataTrusted</td>
<td>Whether this started task runs with the RACF TRUSTED attribute</td>
</tr>
<tr>
<td>racfUpdateAccessCount</td>
<td>Number of matching qualifiers to use when loading generic resource profile names</td>
</tr>
<tr>
<td>racfAutomatic</td>
<td>Whether the program object needs a digital signature</td>
</tr>
<tr>
<td>racfSigverFailure</td>
<td>Conditions under which module load fails when digital signature verification fails</td>
</tr>
<tr>
<td>racfStdataGroup</td>
<td>Group name associated with this started task</td>
</tr>
<tr>
<td>racfTimeZone</td>
<td>Time zone in which a terminal resides</td>
</tr>
<tr>
<td>racfSessionInterval</td>
<td>Maximum number of days the session key is valid</td>
</tr>
<tr>
<td>racfSessionConvSec</td>
<td>Level of security checking when conversations are established with the protected LU</td>
</tr>
<tr>
<td>racfCdtinfoKeyQualifiers</td>
<td>Number of matching qualifiers to use when loading generic resource profile names</td>
</tr>
<tr>
<td>racfCdtinfoFirst</td>
<td>Character type restriction for the first character of the resource profile name</td>
</tr>
<tr>
<td>racfCdtinfoOther</td>
<td>Character type restriction for the characters after the first one in a resource profile name</td>
</tr>
<tr>
<td>racfStdataPrivileged</td>
<td>Whether this started task runs with the RACF PRIVILEGED attribute</td>
</tr>
<tr>
<td>racfStdataGroup</td>
<td>Group name associated with this started task</td>
</tr>
<tr>
<td>racfSessionSessKey</td>
<td>Mask the key value</td>
</tr>
<tr>
<td>racfStdataTrusted</td>
<td>Whether this started task runs with the RACF TRUSTED attribute</td>
</tr>
<tr>
<td>racfServiceGlobalAudit</td>
<td>Digital signature verification events to be audited</td>
</tr>
<tr>
<td>racfStdataPrivileged</td>
<td>Whether this started task runs with the RACF PRIVILEGED attribute</td>
</tr>
<tr>
<td>racfCdtinfoKeyQualifiers</td>
<td>Number of matching qualifiers to use when loading generic resource profile names</td>
</tr>
<tr>
<td>racfCdtinfoFirst</td>
<td>Character type restriction for the first character of the resource profile name</td>
</tr>
<tr>
<td>racfCdtinfoOther</td>
<td>Character type restriction for the characters after the first one in a resource profile name</td>
</tr>
<tr>
<td>racfStdataPrivileged</td>
<td>Whether this started task runs with the RACF PRIVILEGED attribute</td>
</tr>
<tr>
<td>racfStdataTrusted</td>
<td>Whether this started task runs with the RACF TRUSTED attribute</td>
</tr>
<tr>
<td>racfSessionSessKey</td>
<td>Mask the key value</td>
</tr>
<tr>
<td>racfStdataGroup</td>
<td>Group name associated with this started task</td>
</tr>
<tr>
<td>racfTimeZone</td>
<td>Time zone in which a terminal resides</td>
</tr>
<tr>
<td>racfSessionInterval</td>
<td>Maximum number of days the session key is valid</td>
</tr>
<tr>
<td>racfSessionConvSec</td>
<td>Level of security checking when conversations are established with the protected LU</td>
</tr>
</tbody>
</table>

---

Appendix A. Initial LDAP server schema
Initial LDAP server schema

| syntax 1.3.6.1.4.1.1466.115.121.1.26 usage userapplications |
| attributetypes=( 1.3.18.0.2.4.3459 NAME ( 'racfRacList' ) desc 'name of class for which racf shares in-storage generic and discrete profiles' equality caseignorematch |
| syntax 1.3.6.1.4.1.1466.115.121.1.26 usage userapplications |
| attributetypes=( 1.3.18.0.2.4.3460 NAME ( 'racfOptionsFailures' ) desc 'name of class for which racf audits failed access attempts to resources' equality caseignorematch |
| syntax 1.3.6.1.4.1.1466.115.121.1.26 usage userapplications |
| attributetypes=( 1.3.18.0.2.4.3461 NAME ( 'racfLastReferenceDate' ) desc 'date when the resource profile was last referenced' equality casemismatch single-value usage userapplications |
| syntax 1.3.6.1.4.1.1466.115.121.1.26 usage userapplications |
| attributetypes=( 1.3.18.0.2.4.3462 NAME ( 'racfLastChangeDate' ) desc 'date when the resource profile was last changed' equality caseignorematch single-value usage userapplications |
| syntax 1.3.6.1.4.1.1466.115.121.1.26 usage userapplications |
| attributetypes=( 1.3.18.0.2.4.3463 NAME ( 'racfLevel' ) desc 'level number assigned by the installation' equality caseignorematch single-value usage userapplications |
| syntax 1.3.6.1.4.1.1466.115.121.1.26 usage userapplications |
| attributetypes=( 1.3.18.0.2.4.3464 NAME ( 'racfLogOptionsAlways' ) desc 'name of class for which racf audits all access attempts to resources' equality caseignorematch |
| syntax 1.3.6.1.4.1.1466.115.121.1.26 usage userapplications |
| attributetypes=( 1.3.18.0.2.4.3465 NAME ( 'racfLogOptionsDefault' ) desc 'name of class for which racf auditing is controlled by the profile protecting the resource' equality caseignorematch syntax 1.3.6.1.4.1.1466.115.121.1.26 usage userapplications |
| attributetypes=( 1.3.18.0.2.4.3466 NAME ( 'racfKerberosPassword' ) desc 'value of the kerberos password for the realm' equality caseignorematch single-value usage userapplications |
| syntax 1.3.6.1.4.1.1466.115.121.1.26 usage userapplications |
| attributetypes=( 1.3.18.0.2.4.3467 NAME ( 'racfIctxPassword' ) desc 'default ticket lifetime for the local network authentication services' equality caseignorematch single-value syntax 1.3.6.1.4.1.1466.115.121.1.26 usage userapplications |
| syntax 1.3.6.1.4.1.1466.115.121.1.26 usage userapplications |
| attributetypes=( 1.3.18.0.2.4.3468 NAME ( 'racfIctxUseMap' ) desc 'whether the ICTX identity cache stores a valid identity mapping' equality caseignorematch single-value usage userapplications |
| syntax 1.3.6.1.4.1.1466.115.121.1.26 usage userapplications |
| attributetypes=( 1.3.18.0.2.4.3469 NAME ( 'racfGlobal' ) desc 'name of class for which racf performs global access checking' equality caseignorematch usage userapplications |
| syntax 1.3.6.1.4.1.1466.115.121.1.26 usage userapplications |
| attributetypes=( 1.3.18.0.2.4.3470 NAME ( 'racfImRegOptionsDefault' ) desc 'name of class for which racf auditing is controlled by the profile protecting the resource' equality caseignorematch syntax 1.3.6.1.4.1.1466.115.121.1.26 usage userapplications |
| syntax 1.3.6.1.4.1.1466.115.121.1.26 usage userapplications |
| attributetypes=( 1.3.18.0.2.4.3471 NAME ( 'racfImRegPassword' ) desc 'value of the kerberos password for the realm' equality caseignorematch single-value usage userapplications |
| syntax 1.3.6.1.4.1.1466.115.121.1.26 usage userapplications |
| attributetypes=( 1.3.18.0.2.4.3472 NAME ( 'racfImRegGlobal' ) desc 'name of class for which racf performs global access checking' equality caseignorematch usage userapplications |
| syntax 1.3.6.1.4.1.1466.115.121.1.26 usage userapplications |
| attributetypes=( 1.3.18.0.2.4.3473 NAME ( 'racfImRegOptionsDefault' ) desc 'name of class for which racf auditing is controlled by the profile protecting the resource' equality caseignorematch syntax 1.3.6.1.4.1.1466.115.121.1.26 usage userapplications |
| syntax 1.3.6.1.4.1.1466.115.121.1.26 usage userapplications |
| attributetypes=( 1.3.18.0.2.4.3474 NAME ( 'racfImRegOptionsGlobal' ) desc 'name of class for which racf performs global access checking' equality caseignorematch usage userapplications |
| syntax 1.3.6.1.4.1.1466.115.121.1.26 usage userapplications |
| attributetypes=( 1.3.18.0.2.4.3475 NAME ( 'racfImRegGlobalOptionsDefault' ) desc 'name of class for which racf auditing is controlled by the profile protecting the resource' equality caseignorematch syntax 1.3.6.1.4.1.1466.115.121.1.26 usage userapplications |
| syntax 1.3.6.1.4.1.1466.115.121.1.26 usage userapplications |
| attributetypes=( 1.3.18.0.2.4.3476 NAME ( 'racfImRegGlobalOptionsGlobal' ) desc 'name of class for which racf performs global access checking' equality caseignorematch usage userapplications |
| syntax 1.3.6.1.4.1.1466.115.121.1.26 usage userapplications |
| attributetypes=( 1.3.18.0.2.4.3477 NAME ( 'racFDIProviderName' ) desc 'list of job names which can access the dlf objects protected by this resource profile' equality caseignorematch single-value usage userapplications |
| syntax 1.3.6.1.4.1.1466.115.121.1.26 usage userapplications |
| attributetypes=( 1.3.18.0.2.4.3478 NAME ( 'racFDIProviderVersion' ) desc 'whether the dlf object can be retained after use' equality caseignorematch single-value usage userapplications |
| syntax 1.3.6.1.4.1.1466.115.121.1.26 usage userapplications |
| attributetypes=( 1.3.18.0.2.4.3479 NAME ( 'racFDIProviderDescription' ) desc 'distinguished name of the eim domain' equality caseexactmatch single-value usage userapplications |
| syntax 1.3.6.1.4.1.1466.115.121.1.26 usage userapplications |
| attributetypes=( 1.3.18.0.2.4.3480 NAME ( 'racFDIProviderEnvironment' ) desc 'name of the kerberos registry in the eim domain' equality caseexactmatch single-value usage userapplications |
| syntax 1.3.6.1.4.1.1466.115.121.1.26 usage userapplications |
| attributetypes=( 1.3.18.0.2.4.3481 NAME ( 'racFDIProviderInstall' ) desc 'name of the local racf registry in the eim domain' equality caseexactmatch single-value usage userapplications |
| syntax 1.3.6.1.4.1.1466.115.121.1.26 usage userapplications |
| attributetypes=( 1.3.18.0.2.4.3482 NAME ( 'racFDIProviderProfile' ) desc 'the FCLASS, FGENERIC, FROM, and FVOLUME specifications for copying the values from a profile' equality caseignorematch single-value syntax 1.3.6.1.4.1.1466.115.121.1.26 usage userapplications |
| syntax 1.3.6.1.4.1.1466.115.121.1.26 usage userapplications |
| attributetypes=( 1.3.18.0.2.4.3483 NAME ( 'racFDIProviderProfileType' ) desc 'data type of the field' equality caseignorematch syntax 1.3.6.1.4.1.1466.115.121.1.26 usage userapplications |
| syntax 1.3.6.1.4.1.1466.115.121.1.26 usage userapplications |
| attributetypes=( 1.3.18.0.2.4.3484 NAME ( 'racFDIProviderHelp' ) desc 'help text for the field' equality caseignorematch single-value syntax 1.3.6.1.4.1.1466.115.121.1.26 usage userapplications |
| syntax 1.3.6.1.4.1.1466.115.121.1.26 usage userapplications |
Appendix A. Initial LDAP server schema

Initial LDAP server schema

attributetypes=( 1.3.18.0.2.4.3487 NAME ( 'racfcdefMaxLength' )
DESC 'Maximum number of characters the field can contain' EQUALITY caseIgnoreMatch SINGLE-VALUE
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE userApplications )

attributetypes=( 1.3.18.0.2.4.3488 NAME ( 'racfcdefMaxSigLength' )
DESC 'Maximum length of resource and resource profile names when MAXLENX is not specified' EQUALITY caseIgnoreMatch SINGLE-VALUE
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE userApplications )

attributetypes=( 1.3.18.0.2.4.3490 NAME ( 'racfcdefMaxPosit' )
DESC 'POSIT number associated with this class' EQUALITY caseIgnoreMatch SINGLE-VALUE
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE userApplications )

attributetypes=( 1.3.18.0.2.4.3492 NAME ( 'racfcdefRacList' )
DESC 'Whether a RACLIST is required for resource profiles' EQUALITY caseIgnoreMatch SINGLE-VALUE
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE userApplications )

attributetypes=( 1.3.18.0.2.4.3494 NAME ( 'racfcdefDefaultUacc' )
DESC 'Whether SETROPTS GENLIST is allowed for the class' EQUALITY caseIgnoreMatch SINGLE-VALUE
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE userApplications )

attributetypes=( 1.3.18.0.2.4.3495 NAME ( 'racfcdefDefaultRc' )
DESC 'Whether SETROPTS GENERIC and GENCMD are allowed for the class' EQUALITY caseIgnoreMatch SINGLE-VALUE
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE userApplications )

attributetypes=( 1.3.18.0.2.4.3496 NAME ( 'racfcdefGenProcessing' )
DESC 'Whether mixed-case resource profile names are allowed for this class' EQUALITY caseIgnoreMatch SINGLE-VALUE
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE userApplications )

attributetypes=( 1.3.18.0.2.4.3497 NAME ( 'racfcdefMember' )
DESC 'Name of class grouped by the resources within this class' EQUALITY caseIgnoreMatch SINGLE-VALUE
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE userApplications )

attributetypes=( 1.3.18.0.2.4.3498 NAME ( 'racfcdefGroup' )
DESC 'Name of class that groups the resources within this class' EQUALITY caseIgnoreMatch SINGLE-VALUE
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE userApplications )

attributetypes=( 1.3.18.0.2.4.3499 NAME ( 'racfcdefDefaultRacList' )
DESC 'Whether a RACLIST is allowed for the class' EQUALITY caseIgnoreMatch SINGLE-VALUE
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE userApplications )

attributetypes=( 1.3.18.0.2.4.3500 NAME ( 'racfcdefCase' )
DESC 'Whether mixed-case resource profile names are allowed for this class' EQUALITY caseIgnoreMatch SINGLE-VALUE
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE userApplications )

attributetypes=( 1.3.18.0.2.4.3501 NAME ( 'racfcdefDefaultUacc' )
DESC 'Whether SETROPTS GENLIST is allowed for the class' EQUALITY caseIgnoreMatch SINGLE-VALUE
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE userApplications )

attributetypes=( 1.3.18.0.2.4.3502 NAME ( 'racfcdefDefaultRc' )
DESC 'Whether SETROPTS GENERIC and GENCMD are allowed for the class' EQUALITY caseIgnoreMatch SINGLE-VALUE
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE userApplications )

attributetypes=( 1.3.18.0.2.4.3503 NAME ( 'racfcdefGeneric' )
DESC 'Whether SETROPTS GENERIC and GENCMD are allowed for the class' EQUALITY caseIgnoreMatch SINGLE-VALUE
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE userApplications )

attributetypes=( 1.3.18.0.2.4.3504 NAME ( 'racfcdefPosit' )
DESC 'POSIT number associated with this class' EQUALITY caseIgnoreMatch SINGLE-VALUE
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE userApplications )

attributetypes=( 1.3.18.0.2.4.3505 NAME ( 'racfcdefMaxSigLength' )
DESC 'Maximum length of resource and resource profile names when MAXLENX is not specified' EQUALITY caseIgnoreMatch SINGLE-VALUE
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE userApplications )

attributetypes=( 1.3.18.0.2.4.3506 NAME ( 'racfcdefMaxPosit' )
DESC 'POSIT number associated with this class' EQUALITY caseIgnoreMatch SINGLE-VALUE
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE userApplications )

attributetypes=( 1.3.18.0.2.4.3507 NAME ( 'racfcdefRacList' )
DESC 'Whether a RACLIST is required for resource profiles' EQUALITY caseIgnoreMatch SINGLE-VALUE
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE userApplications )

attributetypes=( 1.3.18.0.2.4.3508 NAME ( 'ibm-replicationWaitOnDepency' )
DESC 'Indicates whether the server will await the completion of the replication of dependencies prior to sending a replication update to a consumer.' SINGLE-VALUE SYNTAX 1.3.6.1.4.1.1466.115.121.1.7 USAGE userApplications )

attributetypes=( 1.3.18.0.2.4.3509 NAME ( 'ibm-slapdRep1Version' )
DESC 'This attribute defines the current version of the advanced replication feature.' EQUALITY caseIgnoreMatch ORDERING caseIgnoreOrderingMatch SUBSTR caseIgnoreStringMatch caseIgnoreSubstringMatch SINGLE-VALUE
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7 USAGE directoryOperation )

attributetypes=( 1.3.18.0.2.4.3511 NAME ( 'racfcfIsSymUsage' )
DESC 'Allowable usage of an asymmetric ICSF key' EQUALITY caseIgnoreMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE userApplications )

attributetypes=( 1.3.18.0.2.4.3513 NAME ( 'racfcfIsSymExportable' )
DESC 'How symmetric keys covered by this profile can be exported' EQUALITY caseIgnoreMatch SINGLE-VALUE
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE userApplications )

attributetypes=( 1.3.18.0.2.4.3514 NAME ( 'racfcfIsSymExportCerts' )
DESC 'Digital certificate labels to use to export symmetric keys covered by this profile' EQUALITY caseExactMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 USAGE userApplications )

attributetypes=( 1.3.18.0.2.4.3515 NAME ( 'racfcfIsSymExportKeys' )
DESC 'Key token labels for public keys to use to export symmetric keys covered by this profile'

attributetypes=( 1.2.5.4.0 NAME ( 'objectClass' )
SYNTAX 1.3.6.1.4.1.1466.115.121.1.38 USAGE userApplications )
Initial LDAP server schema

DESC 'True name for an alias entry' SINGLE-VALUE SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
USAGE userApplications
attributeTypes=( 2.5.4.3 NAME ( 'cn' 'commonName' ) SUP name USAGE userApplications )
attributeTypes=( 2.5.4.6 NAME ( 'cn' 'countryName' ) DESC 'A two-letter ISO 3166 country code' SUP name
SINGLE-VALUE USAGE userApplications )
attributeTypes=( 2.5.4.7 NAME ( 'l' 'localityName' ) DESC 'The name of a locality, such as a city, county or other geographic region' SUP name
USAGE userApplications )
attributeTypes=( 2.5.4.8 NAME ( 'st' 'stateOrProvinceName' ) DESC 'The full name of a state or province' SUP name USAGE userApplications )
attributeTypes=( 2.5.4.10 NAME ( 'o' 'organizationName' ) DESC 'The name of an organization' SUP name USAGE userApplications )
attributeTypes=( 2.5.18.1 NAME ( 'name' ) SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 USAGE userApplications )
attributeTypes=( 2.5.18.2 NAME ( 'dn' 'distinguishedName' ) SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 USAGE userApplications )
attributeTypes=( 2.5.4.11 NAME ( 'ou' 'organizationalUnitName' ) DESC ' Defines the distinguished name of an entry that was modified' EQUALITY distinguishedNameMatch SINGLE-VALUE NO-USER-MODIFICATION SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 USAGE userApplications )
attributeTypes=( 2.5.4.41 NAME ( 'name' ) SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 USAGE userApplications )
attributeTypes=( 2.5.4.49 NAME ( 'dn' 'distinguishedName' ) SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 USAGE userApplications )
attributeTypes=( 2.5.4.50 NAME ( 'uniqueMember' ) DESC 'Defines a member of a set' SUP dn
USAGE userApplications )
attributeTypes=( 2.5.4.32 NAME ( 'owner' ) DESC 'Identifies another entry that may contain information related this entry' SUP dn
USAGE userApplications )
attributeTypes=( 2.5.4.35 NAME ( 'userPassword' ) DESC 'Defines the user password' SYNTAX 1.3.6.1.4.1.1466.115.121.1.40 USAGE userApplications )
attributeTypes=( 2.5.4.41 NAME ( 'name' ) SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 USAGE userApplications )
attributeTypes=( 2.5.4.49 NAME ( 'dn' 'distinguishedName' ) SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 USAGE userApplications )
attributeTypes=( 2.5.4.50 NAME ( 'uniqueMember' ) DESC 'Defines a member of a set' SUP dn
USAGE userApplications )
attributeTypes=( 2.5.4.32 NAME ( 'owner' ) DESC 'Identifies another entry that may contain information related this entry' SUP dn
USAGE userApplications )
attributeTypes=( 2.5.4.35 NAME ( 'userPassword' ) DESC 'Defines the user password' SYNTAX 1.3.6.1.4.1.1466.115.121.1.40 USAGE userApplications )
attributeTypes=( 2.5.4.41 NAME ( 'name' ) SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 USAGE userApplications )
attributeTypes=( 2.5.4.49 NAME ( 'dn' 'distinguishedName' ) SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 USAGE userApplications )
attributeTypes=( 2.5.4.50 NAME ( 'uniqueMember' ) DESC 'Defines a member of a set' SUP dn
USAGE userApplications )
attributeTypes=( 2.5.4.32 NAME ( 'owner' ) DESC 'Identifies another entry that may contain information related this entry' SUP dn
USAGE userApplications )
attributeTypes=( 2.5.4.35 NAME ( 'userPassword' ) DESC 'Defines the user password' SYNTAX 1.3.6.1.4.1.1466.115.121.1.40 USAGE userApplications )
attributeTypes=( 2.5.4.41 NAME ( 'name' ) SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 USAGE userApplications )
attributeTypes=( 2.5.4.49 NAME ( 'dn' 'distinguishedName' ) SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 USAGE userApplications )
attributeTypes=( 2.5.4.50 NAME ( 'uniqueMember' ) DESC 'Defines a member of a set' SUP dn
USAGE userApplications )
attributeTypes=( 2.5.4.32 NAME ( 'owner' ) DESC 'Identifies another entry that may contain information related this entry' SUP dn
USAGE userApplications )
attributeTypes=( 2.5.4.35 NAME ( 'userPassword' ) DESC 'Defines the user password' SYNTAX 1.3.6.1.4.1.1466.115.121.1.40 USAGE userApplications )

Initial LDAP server schema

DESC 'Specifies the name of the new superior of the existing entry' EQUALITY distinguishedNameMatch
SINGLE-VALUE NO-USER-MODIFICATION SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 USAGE userApplications
attributeTypes( 1.3.6.1.4.1.1466.115.121.1.14 NAME 'changeTime' DESC 'Time last changed' SINGLE-VALUE
NO-USER-MODIFICATION SYNTAX 1.3.6.1.4.1.1466.115.121.1.24 USAGE userApplications
attributeTypes( 2.16.840.1.113556.1.4.656 NAME 'ref' DESC 'Specifies a URL associated with each member of a group' EQUALITY caseExactMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 USAGE userApplications )
attributeTypes( 0.9.2342.19200300.100.1.1 ACCESS-CLASS normal )
ibattributeTypes( 0.9.2342.19200300.100.1.23 ACCESS-CLASS system )
ibattributeTypes( 0.9.2342.19200300.100.1.24 ACCESS-CLASS system )
ibattributeTypes( 1.2.840.113556.1.4.77 ACCESS-CLASS normal )
ibattributeTypes( 1.2.840.113556.1.4.665 ACCESS-CLASS normal )
ibattributeTypes( 1.2.840.113556.1.4.667 ACCESS-CLASS normal )
ibattributeTypes( 1.3.6.1.4.1 ACCESS-CLASS normal )
ibattributeTypes( 1.3.6.1.1.5 ACCESS-CLASS normal )
ibattributeTypes( 1.3.6.1.4.1.1466.101.120.5 ACCESS-CLASS normal )
ibattributeTypes( 1.3.6.1.4.1.1466.101.120.6 ACCESS-CLASS normal )
ibattributeTypes( 1.3.6.1.4.1.1466.101.120.7 ACCESS-CLASS normal )
ibattributeTypes( 1.3.6.1.4.1.1466.101.120.13 ACCESS-CLASS normal )
ibattributeTypes( 1.3.6.1.4.1.1466.101.120.14 ACCESS-CLASS normal )
ibattributeTypes( 1.3.6.1.4.1.1466.101.120.15 ACCESS-CLASS normal )
ibattributeTypes( 1.3.6.1.4.1.1466.101.120.16 ACCESS-CLASS system )
ibattributeTypes( 1.3.1.8.0.2.4.155 ACCESS-CLASS critical )
ibattributeTypes( 1.3.1.8.0.2.4.185 ACCESS-CLASS sensitive )
ibattributeTypes( 1.3.1.8.0.2.4.186 ACCESS-CLASS critical )
ibattributeTypes( 1.3.1.8.0.2.4.187 ACCESS-CLASS sensitive )
ibattributeTypes( 1.3.1.8.0.2.4.188 ACCESS-CLASS sensitive )
ibattributeTypes( 1.3.1.8.0.2.4.189 ACCESS-CLASS sensitive )
ibattributeTypes( 1.3.1.8.0.2.4.190 ACCESS-CLASS sensitive )
ibattributeTypes( 1.3.1.8.0.2.4.191 ACCESS-CLASS sensitive )
ibattributeTypes( 1.3.1.8.0.2.4.192 ACCESS-CLASS sensitive )
ibattributeTypes( 1.3.1.8.0.2.4.193 ACCESS-CLASS sensitive )
ibattributeTypes( 1.3.1.8.0.2.4.194 ACCESS-CLASS sensitive )
ibattributeTypes( 1.3.1.8.0.2.4.195 ACCESS-CLASS sensitive )
ibattributeTypes( 1.3.1.8.0.2.4.197 ACCESS-CLASS sensitive )
ibattributeTypes( 1.3.1.8.0.2.4.198 ACCESS-CLASS critical )
ibattributeTypes( 1.3.1.8.0.2.4.199 ACCESS-CLASS sensitive )
ibattributeTypes( 1.3.1.8.0.2.4.200 ACCESS-CLASS sensitive )
ibattributeTypes( 1.3.1.8.0.2.4.201 ACCESS-CLASS sensitive )
ibattributeTypes( 1.3.1.8.0.2.4.202 ACCESS-CLASS sensitive )
ibattributeTypes( 1.3.1.8.0.2.4.203 ACCESS-CLASS sensitive )
ibattributeTypes( 1.3.1.8.0.2.4.204 ACCESS-CLASS critical )
ibattributeTypes( 1.3.1.8.0.2.4.205 ACCESS-CLASS critical )
ibattributeTypes( 1.3.1.8.0.2.4.206 ACCESS-CLASS sensitive )
ibattributeTypes( 1.3.1.8.0.2.4.207 ACCESS-CLASS sensitive )
ibattributeTypes( 1.3.1.8.0.2.4.208 ACCESS-CLASS sensitive )
ibattributeTypes( 1.3.1.8.0.2.4.209 ACCESS-CLASS sensitive )
ibattributeTypes( 1.3.1.8.0.2.4.210 ACCESS-CLASS sensitive )
ibattributeTypes( 1.3.1.8.0.2.4.211 ACCESS-CLASS sensitive )
ibattributeTypes( 1.3.1.8.0.2.4.212 ACCESS-CLASS sensitive )
ibattributeTypes( 1.3.1.8.0.2.4.213 ACCESS-CLASS sensitive )
ibattributeTypes( 1.3.1.8.0.2.4.214 ACCESS-CLASS sensitive )
ibattributeTypes( 1.3.1.8.0.2.4.215 ACCESS-CLASS sensitive )
ibattributeTypes( 1.3.1.8.0.2.4.216 ACCESS-CLASS sensitive )
ibattributeTypes( 1.3.1.8.0.2.4.217 ACCESS-CLASS sensitive )
ibattributeTypes( 1.3.1.8.0.2.4.218 ACCESS-CLASS sensitive )
ibattributeTypes( 1.3.1.8.0.2.4.219 ACCESS-CLASS sensitive )
ibattributeTypes( 1.3.1.8.0.2.4.220 ACCESS-CLASS sensitive )
ibattributeTypes( 1.3.1.8.0.2.4.221 ACCESS-CLASS sensitive )
ibattributeTypes( 1.3.1.8.0.2.4.222 ACCESS-CLASS sensitive )
ibattributeTypes( 1.3.1.8.0.2.4.223 ACCESS-CLASS sensitive )
ibattributeTypes( 1.3.1.8.0.2.4.224 ACCESS-CLASS sensitive )
ibattributeTypes( 1.3.1.8.0.2.4.225 ACCESS-CLASS sensitive )
ibattributeTypes( 1.3.1.8.0.2.4.226 ACCESS-CLASS sensitive )
ibattributeTypes( 1.3.1.8.0.2.4.227 ACCESS-CLASS sensitive )
ibattributeTypes( 1.3.1.8.0.2.4.228 ACCESS-CLASS sensitive )
ibattributeTypes( 1.3.1.8.0.2.4.229 ACCESS-CLASS sensitive )
ibattributeTypes( 1.3.1.8.0.2.4.230 ACCESS-CLASS sensitive )
ibattributeTypes( 1.3.1.8.0.2.4.231 ACCESS-CLASS sensitive )
ibattributeTypes( 1.3.1.8.0.2.4.232 ACCESS-CLASS sensitive )
ibattributeTypes( 1.3.1.8.0.2.4.233 ACCESS-CLASS sensitive )
ibattributeTypes( 1.3.1.8.0.2.4.234 ACCESS-CLASS sensitive )
ibattributeTypes( 1.3.1.8.0.2.4.235 ACCESS-CLASS sensitive )
ibattributeTypes( 1.3.1.8.0.2.4.236 ACCESS-CLASS sensitive )
ibattributeTypes( 1.3.1.8.0.2.4.237 ACCESS-CLASS sensitive )
ibattributeTypes( 1.3.1.8.0.2.4.238 ACCESS-CLASS sensitive )
ibattributeTypes( 1.3.1.8.0.2.4.239 ACCESS-CLASS sensitive )
Initial LDAP server schema

ibmattributetypes=( 1.3.18.0.2.4.240 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.241 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.242 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.243 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.244 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.245 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.246 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.247 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.248 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.249 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.250 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.251 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.252 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.253 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.254 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.255 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.256 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.257 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.258 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.259 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.260 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.261 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.262 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.263 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.264 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.265 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.266 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.267 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.268 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.269 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.270 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.271 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.272 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.273 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.274 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.275 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.276 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.277 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.278 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.279 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.280 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.281 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.282 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.283 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.285 ACCESS-CLASS restricted )
ibmattributetypes=( 1.3.18.0.2.4.286 ACCESS-CLASS restricted )
ibmattributetypes=( 1.3.18.0.2.4.287 ACCESS-CLASS system )
ibmattributetypes=( 1.3.18.0.2.4.288 ACCESS-CLASS restricted )
ibmattributetypes=( 1.3.18.0.2.4.289 ACCESS-CLASS restricted )
ibmattributetypes=( 1.3.18.0.2.4.290 ACCESS-CLASS system )
ibmattributetypes=( 1.3.18.0.2.4.298 ACCESS-CLASS normal )
ibmattributetypes=( 1.3.18.0.2.4.299 ACCESS-CLASS critical )
ibmattributetypes=( 1.3.18.0.2.4.300 ACCESS-CLASS critical )
ibmattributetypes=( 1.3.18.0.2.4.301 ACCESS-CLASS normal )
ibmattributetypes=( 1.3.18.0.2.4.302 ACCESS-CLASS normal )
ibmattributetypes=( 1.3.18.0.2.4.303 ACCESS-CLASS normal )
ibmattributetypes=( 1.3.18.0.2.4.304 ACCESS-CLASS normal )
ibmattributetypes=( 1.3.18.0.2.4.470 ACCESS-CLASS system )
ibmattributetypes=( 1.3.18.0.2.4.478 ACCESS-CLASS system )
ibmattributetypes=( 1.3.18.0.2.4.826 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.827 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.828 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.829 ACCESS-CLASS sensitive )
ibmatributetypes=( 1.3.18.0.2.4.830 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.831 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.1088 ACCESS-CLASS normal )
ibmattributetypes=( 1.3.18.0.2.4.1100 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.1108 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.1109 ACCESS-CLASS normal )
ibmattributetypes=( 1.3.18.0.2.4.1110 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.1114 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.1116 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.1117 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.1118 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.1119 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.1120 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.1121 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.1122 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.1123 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.1124 ACCESS-CLASS sensitive )
ibmattributetypes=( 1.3.18.0.2.4.1125 ACCESS-CLASS sensitive )
Appendix A. Initial LDAP server schema
Initial LDAP server schema
|
|
|
|

|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|

ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(
ibmattributetypes=(

738

1.3.18.0.2.4.3150
1.3.18.0.2.4.3151
1.3.18.0.2.4.3152
1.3.18.0.2.4.3153
1.3.18.0.2.4.3215
1.3.18.0.2.4.3216
1.3.18.0.2.4.3223
1.3.18.0.2.4.3239
1.3.18.0.2.4.3240
1.3.18.0.2.4.3241
1.3.18.0.2.4.3242
1.3.18.0.2.4.3243
1.3.18.0.2.4.3244
1.3.18.0.2.4.3245
1.3.18.0.2.4.3261
1.3.18.0.2.4.3263
1.3.18.0.2.4.3264
1.3.18.0.2.4.3265
1.3.18.0.2.4.3266
1.3.18.0.2.4.3267
1.3.18.0.2.4.3268
1.3.18.0.2.4.3269
1.3.18.0.2.4.3270
1.3.18.0.2.4.3288
1.3.18.0.2.4.3294
1.3.18.0.2.4.3295
1.3.18.0.2.4.3296
1.3.18.0.2.4.3320
1.3.18.0.2.4.3321
1.3.18.0.2.4.3330
1.3.18.0.2.4.3331
1.3.18.0.2.4.3332
1.3.18.0.2.4.3333
1.3.18.0.2.4.3334
1.3.18.0.2.4.3335
1.3.18.0.2.4.3342
1.3.18.0.2.4.3343
1.3.18.0.2.4.3344
1.3.18.0.2.4.3373
1.3.18.0.2.4.3417
1.3.18.0.2.4.3418
1.3.18.0.2.4.3419
1.3.18.0.2.4.3420
1.3.18.0.2.4.3421
1.3.18.0.2.4.3422
1.3.18.0.2.4.3423
1.3.18.0.2.4.3424
1.3.18.0.2.4.3425
1.3.18.0.2.4.3426
1.3.18.0.2.4.3427
1.3.18.0.2.4.3428
1.3.18.0.2.4.3429
1.3.18.0.2.4.3430
1.3.18.0.2.4.3431
1.3.18.0.2.4.3432
1.3.18.0.2.4.3433
1.3.18.0.2.4.3434
1.3.18.0.2.4.3435
1.3.18.0.2.4.3436
1.3.18.0.2.4.3437
1.3.18.0.2.4.3438
1.3.18.0.2.4.3439
1.3.18.0.2.4.3440
1.3.18.0.2.4.3441
1.3.18.0.2.4.3442
1.3.18.0.2.4.3443
1.3.18.0.2.4.3444
1.3.18.0.2.4.3445
1.3.18.0.2.4.3446
1.3.18.0.2.4.3447
1.3.18.0.2.4.3448
1.3.18.0.2.4.3449
1.3.18.0.2.4.3450
1.3.18.0.2.4.3451
1.3.18.0.2.4.3452
1.3.18.0.2.4.3453
1.3.18.0.2.4.3454
1.3.18.0.2.4.3455
1.3.18.0.2.4.3456
1.3.18.0.2.4.3457
1.3.18.0.2.4.3458
1.3.18.0.2.4.3459
1.3.18.0.2.4.3460

ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS
ACCESS-CLASS

normal )
normal )
normal )
normal )
sensitive )
sensitive )
normal )
sensitive )
sensitive )
sensitive )
critical )
sensitive )
sensitive )
critical )
critical )
critical )
critical )
critical )
critical )
critical )
critical )
critical )
critical )
normal )
normal )
normal )
normal )
critical )
critical )
critical )
critical )
critical )
critical )
critical )
critical )
sensitive )
critical )
sensitive )
critical )
sensitive )
sensitive )
sensitive )
sensitive )
sensitive )
sensitive )
sensitive )
sensitive )
sensitive )
sensitive )
sensitive )
sensitive )
sensitive )
sensitive )
sensitive )
sensitive )
sensitive )
sensitive )
sensitive )
sensitive )
sensitive )
sensitive )
sensitive )
sensitive )
critical )
critical )
sensitive )
sensitive )
sensitive )
sensitive )
sensitive )
sensitive )
sensitive )
sensitive )
sensitive )
sensitive )
sensitive )
sensitive )
sensitive )
sensitive )
sensitive )
sensitive )
sensitive )
sensitive )

z/OS V1R11.0 IBM Tivoli Directory Server Administration and Use for z/OS


Appendix A. Initial LDAP server schema
Initial LDAP server schema

```plaintext
ibnattributeTypes=( 2.16.840.1.113730.3.1.5 ACCESS-CLASS normal )
ibnattributeTypes=( 2.16.840.1.113730.3.1.6 ACCESS-CLASS normal )
ibnattributeTypes=( 2.16.840.1.113730.3.1.7 ACCESS-CLASS normal )
ibnattributeTypes=( 2.16.840.1.113730.3.1.8 ACCESS-CLASS sensitive )
ibnattributeTypes=( 2.16.840.1.113730.3.1.9 ACCESS-CLASS normal )
ibnattributeTypes=( 2.16.840.1.113730.3.1.10 ACCESS-CLASS normal )
ibnattributeTypes=( 2.16.840.1.113730.3.1.11 ACCESS-CLASS normal )
ibnattributeTypes=( 2.16.840.1.113730.3.1.12 ACCESS-CLASS normal )
ibnattributeTypes=( 2.16.840.1.113730.3.1.13 ACCESS-CLASS normal )
ibnattributeTypes=( 2.16.840.1.113730.3.1.14 ACCESS-CLASS normal )
objectclasses=( 1.3.18.0.2.6.28 NAME ( 'container' ) DESC 'An object that can contain other objects' )
objectclasses=( 1.3.18.0.2.6.29 NAME ( 'container' ) DESC 'An object that can contain other objects' )
objectclasses=( 1.3.18.0.2.6.30 NAME ( 'container' ) DESC 'An object that can contain other objects' )
objectclasses=( 1.3.18.0.2.6.31 NAME ( 'container' ) DESC 'An object that can contain other objects' )
objectclasses=( 1.3.18.0.2.6.32 NAME ( 'container' ) DESC 'An object that can contain other objects' )
objectclasses=( 1.3.18.0.2.6.33 NAME ( 'container' ) DESC 'An object that can contain other objects' )
objectclasses=( 1.3.18.0.2.6.34 NAME ( 'container' ) DESC 'An object that can contain other objects' )
objectclasses=( 1.3.18.0.2.6.35 NAME ( 'container' ) DESC 'An object that can contain other objects' )
objectclasses=( 1.3.18.0.2.6.36 NAME ( 'container' ) DESC 'An object that can contain other objects' )
objectclasses=( 1.3.18.0.2.6.37 NAME ( 'container' ) DESC 'An object that can contain other objects' )
objectclasses=( 1.3.18.0.2.6.38 NAME ( 'container' ) DESC 'An object that can contain other objects' )
objectclasses=( 1.3.18.0.2.6.39 NAME ( 'container' ) DESC 'An object that can contain other objects' )
objectclasses=( 1.3.18.0.2.6.40 NAME ( 'container' ) DESC 'An object that can contain other objects' )
objectclasses=( 1.3.18.0.2.6.41 NAME ( 'container' ) DESC 'An object that can contain other objects' )
objectclasses=( 1.3.18.0.2.6.42 NAME ( 'container' ) DESC 'An object that can contain other objects' )
objectclasses=( 1.3.18.0.2.6.43 NAME ( 'container' ) DESC 'An object that can contain other objects' )
objectclasses=( 1.3.18.0.2.6.44 NAME ( 'container' ) DESC 'An object that can contain other objects' )
objectclasses=( 1.3.18.0.2.6.45 NAME ( 'container' ) DESC 'An object that can contain other objects' )
objectclasses=( 1.3.18.0.2.6.46 NAME ( 'container' ) DESC 'An object that can contain other objects' )
objectclasses=( 1.3.18.0.2.6.47 NAME ( 'container' ) DESC 'An object that can contain other objects' )
objectclasses=( 1.3.18.0.2.6.48 NAME ( 'container' ) DESC 'An object that can contain other objects' )
objectclasses=( 1.3.18.0.2.6.49 NAME ( 'container' ) DESC 'An object that can contain other objects' )
objectclasses=( 1.3.18.0.2.6.50 NAME ( 'container' ) DESC 'An object that can contain other objects' )
objectclasses=( 1.3.18.0.2.6.51 NAME ( 'container' ) DESC 'An object that can contain other objects' )
objectclasses=( 1.3.18.0.2.6.52 NAME ( 'container' ) DESC 'An object that can contain other objects' )
objectclasses=( 1.3.18.0.2.6.53 NAME ( 'container' ) DESC 'An object that can contain other objects' )
objectclasses=( 1.3.18.0.2.6.54 NAME ( 'container' ) DESC 'An object that can contain other objects' )
objectclasses=( 1.3.18.0.2.6.55 NAME ( 'racfbase' ) DESC 'Represents the base of the Directory Information Tree that publishes information stored by the OS/390 Security Server RACF service' )
objectclasses=( 1.3.18.0.2.6.56 NAME ( 'racfProfileType' ) DESC 'Represents a RACFUSER Profile entry' )
objectclasses=( 1.3.18.0.2.6.57 NAME ( 'racfBaseCommon' ) DESC 'Represents the OS/390 DCE information in a RACF USER profile' )
objectclasses=( 1.3.18.0.2.6.58 NAME ( 'racfUser' ) DESC 'Represents a RACFUSER Profile entry' )
objectclasses=( 1.3.18.0.2.6.59 NAME ( 'racfGroup' ) DESC 'Represents a RACF GROUP Profile entry' )
objectclasses=( 1.3.18.0.2.6.60 NAME ( 'SAFDpSegment' ) DESC 'Represents the SAF DFP portions of a RACF USER or GROUP profile' )
objectclasses=( 1.3.18.0.2.6.61 NAME ( 'racfGroupOmvsSegment' ) DESC 'Represents the OS/390 OMVS User information portion of a RACF GROUP profile' )
objectclasses=( 1.3.18.0.2.6.62 NAME ( 'racfGroupOvmSegment' ) DESC 'Represents the OS/390 OVM User information portion of a RACF GROUP profile' )
objectclasses=( 1.3.18.0.2.6.63 NAME ( 'racfUserOmvsSegment' ) DESC 'Represents the OS/390 OMVS User information portion of a RACF USER profile' )
objectclasses=( 1.3.18.0.2.6.64 NAME ( 'racfUserOvmSegment' ) DESC 'Represents the OS/390 OVM User information portion of a RACF USER profile' )
objectclasses=( 1.3.18.0.2.6.65 NAME ( 'SAFTsSegment' ) DESC 'Represents the OS/390 TSO information in a RACF USER profile' )
objectclasses=( 1.3.18.0.2.6.66 NAME ( 'racfCicsSegment' ) DESC 'Represents the OS/390 CICS information in a RACF USER profile' )
objectclasses=( 1.3.18.0.2.6.67 NAME ( 'racfPrimaryLanguage' ) DESC 'Represents the OS/390 language information in a RACF USER profile' )
objectclasses=( 1.3.18.0.2.6.68 NAME ( 'racfSecondaryLanguage' ) DESC 'Represents the OS/390 language information in a RACF USER profile' )
objectclasses=( 1.3.18.0.2.6.69 NAME ( 'racfWorkAttrSegment' ) DESC 'Represents the OS/390 work attributes information in a RACF USER profile' )
objectclasses=( 1.3.18.0.2.6.70 NAME ( 'racfDefaultPolicySegment' ) DESC 'Represents the default RACF policy information in a RACF USER profile' )
objectclasses=( 1.3.18.0.2.6.71 NAME ( 'racfDCESegment' ) DESC 'Represents the OS/390 DCE information in a RACF USER profile' )
```

740 z/OS V1R11.0 IBM Tivoli Directory Server Administration and Use for z/OS
Appendix A. Initial LDAP server schema

```
replicaPort $ replicaBindMethod $ replicaUsesSSL $ replicaUpdateTimeInterval )
objectclasses=( 1.3.18.0.2.6.74 NAME ( 'aliasObject' ) DESC 'Defines an alias for a directory entry' AUXILIARY SUP ( top ) MUST ( aliasObjectName ) )
objectclasses=( 1.3.18.0.2.6.75 NAME ( 'accessGroup' ) DESC 'Group used for access control' STUDENTAL SUP ( top ) MUST ( cn ) MAY ( member $ businessCategory $ seeAlso $ owner $ ou $ $ description ) )
objectclasses=( 1.3.18.0.2.6.76 NAME ( 'accessRole' ) DESC 'Role used for access control' STUDENTAL SUP ( top ) MUST ( cn ) MAY ( member $ businessCategory $ seeAlso $ owner $ ou $ $ description ) )
objectclasses=( 1.3.18.0.2.6.174 NAME ( 'ibmSubschema' ) AUXILIARY SUP ( subschema ) MAY ( ibmAttributetypes ) )
objectclasses=( 1.3.18.0.2.6.241 NAME ( 'ibm-securityIdentities' ) DESC 'Defines the security identities of a user' AUXILIARY SUP ( top ) MAY ( altSecurityIdentities $ userPrincipalName ) )
objectclasses=( 1.3.18.0.2.6.248 NAME ( 'racfLNotesSegment' ) DESC 'Represents the OS/390 LNOTES segment information in a RACF USER profile' AUXILIARY SUP ( top ) MAY ( racfLNotesShortName ) )
objectclasses=( 1.3.18.0.2.6.249 NAME ( 'krbAlias' ) DESC 'Kerberos aliases' AUXILIARY SUP ( top ) MAY ( krbAliasedObjectName $ krbGlobalAliases ) )
objectclasses=( 1.3.18.0.2.6.260 NAME ( 'ibm-changeLog' ) DESC 'IBM extension to changeLogEntry object class' STUDENTAL SUP ( top ) MAY ( ibm-changeLogName ) )
objectclasses=( 1.3.18.0.2.6.261 NAME ( 'ibm-nativeAuthentication' ) DESC 'Use native security manager for authentication' AUXILIARY SUP ( top ) MUST ( ibm-nativeId ) )
objectclasses=( 1.3.18.0.2.6.262 NAME ( 'ibm-extensionChangeLog' ) DESC 'Represents a single changeLogEntry object class' STUDENTAL SUP ( top ) MAY ( description ) )
objectclasses=( 1.3.18.0.2.6.264 NAME ( 'accessGroup' ) DESC 'Group used for access control' STUDENTAL SUP ( top ) MUST ( cn ) MAY ( member $ businessCategory $ seeAlso $ owner $ ou $ $ description ) )
objectclasses=( 1.3.18.0.2.6.266 NAME ( 'accessRole' ) DESC 'Role used for access control' STUDENTAL SUP ( top ) MUST ( cn ) MAY ( member $ businessCategory $ seeAlso $ owner $ ou $ $ description ) )
objectclasses=( 1.3.18.0.2.6.267 NAME ( 'ibm-nativeAuthentication' ) DESC 'Use native security manager for authentication' AUXILIARY SUP ( top ) MUST ( ibm-nativeId ) )
objectclasses=( 1.3.18.0.2.6.270 NAME ( 'ibm-replicaSubentry' ) DESC 'Represents replication of a given subtree from a server to the consumer identified in this object' STUDENTAL SUP ( top ) MAY ( member ) )
objectclasses=( 1.3.18.0.2.6.271 NAME ( 'ibm-replicaGroup' ) DESC 'Represents a collection of servers participating in replication' STUDENTAL SUP ( top ) MUST ( ibm-replicaGroup ) MAY ( description ) )
objectclasses=( 1.3.18.0.2.6.272 NAME ( 'ibm-replicaSubentry' ) DESC 'Represents replication of a given subtree from a server to the consumer identified in this object' STUDENTAL SUP ( top ) MAY ( member ) )
objectclasses=( 1.3.18.0.2.6.273 NAME ( 'ibm-replicaGroup' ) DESC 'Represents a collection of servers participating in replication' STUDENTAL SUP ( top ) MUST ( ibm-replicaGroup ) MAY ( description ) )
objectclasses=( 1.3.18.0.2.6.278 NAME ( 'accessRole' ) DESC 'Role used for access control' STUDENTAL SUP ( top ) MAY ( member $ businessCategory $ seeAlso $ owner $ ou $ $ description ) )
```

Initial LDAP server schema
Initial LDAP server schema

MAY ( ibm-slapdMasterPW )
objectclasses=( 1.3.18.0.2.6.496 NAME ( 'ibm-slapdReplication' )
DESC 'Contains the default bind credentials and master server referral URL. This is used when the server contains one or more replication contexts that are replicated to it by other servers. This server may be acting as one of several masters or as a read only replica. If the MasterDN is specified without the Master PW attribute, kerberos authentication is used.' STRUCTURAL SUP ( top ) MUST ( cn )
MAY ( ibm-slapdMasterDN $ ibm-slapdMasterPW $ ibm-slapdMasterReferral $ ibm-slapdNoReplConflictResolution )

MAY ( ibm-slapdReplicationCredentials )
objectclasses=( 1.3.18.0.2.6.521 NAME ( 'ibm-replCredName' )
ABSTRACT SUP ( top )
MAY ( cn $ description $ ibm-replCredName )

MAY ( ibm-replicaGateway )
objectclasses=( 1.3.18.0.2.6.588 NAME ( 'ibm-slapdLogConfig' )
DESC 'Log management configuration.' AUXILIARY SUP ( top )

MAY ( ibm-slapdReplicationConfiguration )
objectclasses=( 1.3.18.0.2.6.596 NAME ( 'ibm-slapdLogReplConflictMaxEntrySize' )
MAY ( description $ ibm-replOnHold $ ibm-slapdMaxPendingChangesDisplayed $ ibm-slapdReplConflictMaxEntrySize $ ibm-slapdReplRestrictedAccess )

MAY ( ibm-replicationFilter )
objectclasses=( 1.3.18.0.2.6.607 NAME ( 'ibm-replicationFilter' )
DESC 'An auxiliary class attached to an ibm-replicaSubentry to indicate the associated server is acting as a gateway server.' AUXILIARY SUP ( top )
MAY ( cn $ ibm-replFilterAttr )

MAY ( racfResource )
objectclasses=( 1.3.18.0.2.6.627 NAME ( 'racfResource' )
DESC 'Provides naming attribute for a discrete or generic RACF resource profile in a class.' STRUCTURAL SUP ( top )
MAY ( profileName )

MAY ( ibm-tdszTop )
objectclasses=( 1.3.18.0.2.6.636 NAME ( 'ibm-tdszTop' )
DESC 'Global configuration settings for IBM Directory Server on z/OS.' STRUCTURAL SUP ( top )
MAY ( cn $ ibm-tdszServerId $ ibm-tdszMaxPendingChangesDisplayed )

MAY ( alias )
objectclasses=( 2.5.6.1 NAME ( 'alias' )
DESC 'Defines an alias for a directory entry' STRUCTURAL SUP ( top )
MAY ( aliasedObjectName )

MAY ( groupOfNames )
objectclasses=( 2.5.6.9 NAME ( 'groupOfNames' )
DESC 'Defines entries for a group of names' STRUCTURAL SUP ( top )
MAY ( aliasedObjectName )

MAY ( groupOfUniqueNames )
objectclasses=( 2.5.6.17 NAME ( 'groupOfUniqueNames' )
DESC 'Defines entries for a group of unique names' STRUCTURAL SUP ( top )
MAY ( uniqueMember )

MAY ( extensibleObject )
objectclasses=( 2.16.840.1.113730.3.2.1 NAME ( 'extensibleObject' )
DESC 'Permits the entry to hold any attribute type defined in the schema' AUXILIARY SUP ( top )
MAY ( objectClasses $ attributeTypes $ matchingRules $ matchingRuleUse $ ldapSyntaxes )

MAY ( referral )
objectclasses=( 2.16.840.1.113730.3.2.2 NAME ( 'referral' )
DESC 'Represents a pointer to another server' STRUCTURAL SUP ( top )
MAY ( referral )

MAY ( memberURL )
objectclasses=( 2.16.840.1.113730.3.2.33 NAME ( 'memberURL' )
DESC 'Represents a group of URLs' STRUCTURAL SUP ( top )
MAY ( memberURL )

MAY ( changeLogEntry )
objectclasses=( 2.16.840.1.113730.3.2.3 NAME ( 'changeLogEntry' )
DESC 'Used to represent changes made to a directory server' STRUCTURAL SUP ( top )
MAY ( modifiersName $ changes $ newRdn $ deleteOldRdn $ newSuperior )

MAY ( referral )
objectclasses=( 2.16.840.1.113730.3.2.6 NAME ( 'referral' )
DESC 'Represents a pointer to another server' STRUCTURAL SUP ( top )
MAY ( referral )

MAY ( memberURL )
objectclasses=( 2.16.840.1.113730.3.2.33 NAME ( 'memberURL' )
DESC 'Represents a group of URLs' STRUCTURAL SUP ( top )
MAY ( memberURL )
Appendix B. SPUFI files

This appendix shows the following SPUFI files:

- The DSTDBMDB SPUFI file
- The TDBMMGRT SPUFI file

Note: The same SPUFI files are used to generate both a TDBM and a GDBM (when DB2-based) database.

The DSTDBMDB SPUFI file

---**********************************************************************
--* Licensed Materials - Property of IBM
--* 5694-A01
--* Copyright IBM Corp. 2006, 2008
--**********************************************************************
--
-- Use the following statements to create your Directory Server DB2
-- database, tablespaces and indexes in SPUFI.
--
-- You will need to make DB2 decisions, in terms of buffer pool
-- size selection for tablespaces and column size selection, all of
-- which will be directly related to the data that will be stored in
-- the database. See the instructions below for more information.
--
-- ***************************************
-- Database Name Information
-- ***************************************
-- Change -DB_NAME- to the name of the LDAP database name you want to
-- create.
-- Example: GLDDB
--
-- ***************************************
-- DataBase Owner Information
-- ***************************************
-- Change -DB_USERID- to the MVS database owner id. This ID will be the
-- highlevel qualifier for the tables. Be sure this value is updated to
-- match what is defined for dbUserid in the LDAP server configuration
-- file.
-- Example: GLDSRV
--
-- ***************************************
-- LDAP Server User ID Information
-- ***************************************
-- Change -LDAP_USERID- to the LDAP server user id. This ID will be the
-- highlevel qualifier for the tables. Be sure this value is updated to
-- match what is defined for ldapuserid in the LDAP server configuration
-- file.
-- Example: GLDSRV
--
-- ***************************************
-- Database Plan Name Information
-- ***************************************
-- Change -DB_PLAN- to the DB2 CLI plan name.
-- Example: DSNACLI
--
-- ***************************************
-- Tablespace Information
-- ***************************************
--
-- NOTE: Refer to the DB2 manuals for a complete listing of valid
-- buffer pool names.
-- *******************************************************************
-- Change -ENTRYTS- to the LDAP entry tablespace name you want to
-- create.
--
-- Change -ENTRYTS_BP0- to the buffer pool name for the LDAP entry
-- tablespace. The size of the buffer pool can be determined with
-- the formula:
--
-- result = 62 bytes + <dn column trunc size (from below)> +
-- <maximum full size of a DN (from below)> +
-- <size of entry data (which includes creator's DN and
-- modifiers DN)>
--
-- There is also a concept of a "spill over" table, where if the
-- entry data does not fit into the row size, it will be broken up
-- in order to fit into a row. Entry data may be spread across
-- multiple rows if needed. So in the above formula, the <size of
-- entry data> does not need to be the maximum size of the data,
-- maybe the median size of the data would be a better choice. See
-- the long entry tablespace description below.
--
-- The suggested buffer pool name is BP0, which represents a buffer
-- pool size of 4K.
--
-- Change -LENTRYTS- to the LDAP long entry tablespace name you
-- want to create.
--
-- Change -LENTRYTS_BP0- to the buffer pool name for the LDAP long
-- entry tablespace. The long entry table space will hold "spill
-- over" rows for entry data that does not fit into the entry table
-- tablespace. To minimize the number of spill over rows, choose a
-- large buffer pool size.
--
-- The suggested buffer pool name is BP0, which represents a buffer
-- pool size of 4K.
--
-- Change -LATTRTS- to the LDAP long attribute tablespace name you
-- want to create.
--
-- Change -LATTRTS_BP0- to the buffer pool name for the LDAP long
-- attribute tablespace. The long attribute table space will hold
-- "spill over" rows for attribute data that does not fit into the
-- entry table tablespace. To minimize the number of spill over
-- rows, choose a large buffer pool size.
--
-- The suggested buffer pool name is BP0, which represents a buffer
-- pool size of 4K.
--
-- Change -SEARCHTS- to the LDAP search tablespace name you want to
-- create.
--
-- Change -SEARCHTS_BP0- to the buffer pool name for the LDAP search
-- tablespace. The size of the buffer pool can be determined with
-- the simple formula:
--
-- result = 16 bytes + <search column trunc size (from below)> +
-- <maximum size of attribute value you would like to
-- search for>
--
-- The result value is the maximum number of bytes a row in the
-- search table containing an attribute value will occupy. Choose
-- a buffer pool size which will accommodate this size.
--
-- The suggested buffer pool name is BP0, which represents a buffer
-- pool size of 4K.
--
-- Change -MISCTS- to the LDAP miscellaneous tablespace name you
-- want to create.
-- Change -MISCTS_BP0- to the buffer pool name for the LDAP
-- miscellaneous tablespace.
--
-- The suggested buffer pool name is BP0, which represents a buffer
-- pool size of 4K.
--
-- Change -DESCCTS- to the LDAP descendants tablespace name you want
-- to create.
--
-- Change -DESCCTS_BP0- to the buffer pool name for the LDAP
-- descendants tablespace.
--
-- The suggested buffer pool name is BP0, which represents a buffer
-- pool size of 4K.
--
-- Change -REPTS- to the LDAP replica tablespace name you want to
-- create.
--
-- Change -REPTS_BP0- to the buffer pool name for the LDAP replica
-- tablespace.
--
-- The suggested buffer pool name is BP0, which represents a buffer
-- pool size of 4K.
--
-- *********************************
-- Column Size Selection Information
-- *********************************
-- All searchable attributes of a given entry will be stored in two
-- forms. The first will be a truncated version, which will be used as
-- part of a DB2 index. The second version will be the entire attribute
-- value, potentially truncated by the buffer pool size you choose. The
-- reason two versions are stored is so that LDAP/DB2 can use indexes to
-- increase search performance. The reason we do not index the entire
-- searchable attribute value is because the cost (in terms of DASD)
-- associated with having indexes on a large column where there is a
-- large amount of data.
--
-- The choice of the search column trunc size should take into account
-- system limits you may have (as described in the above), and should
-- account for the typical size of the attribute values that are stored
-- in LDAP. For example, if most of your data is only 20 bytes long,
-- choosing 20 for this trunc size would be wise.
--
-- Change -SEARCH_TRUNC_SIZE- to the search column trunc size you
-- determine best fits your attribute data.
--
-- The suggested size is 32.
-- The minimum size is 8
--
-- Another search performance enhancement is related to the DN
-- attribute. The DN attribute value is stored separately from the
-- entry data to allow a fast path lookup. It is also stored in two
-- versions as well. The reasons are similar to those mentioned above
-- for the attribute column. Since the DN data is stored in it's own
-- column, you need to define the maximum DN attribute value size here.
-- You also need to choose a dn column trunc size that best fits your
-- data.
--
-- Change -ENTRY_DN_TRUNC_SIZE- to the dn trunc size you determine best
-- fits your dn data.
--
-- The suggested size for a TDBM backend is 64.
-- The suggested size for a DB2 based GDBM backend is 32.
-- The minimum size is 8
--
-- Note: If you are going to use DB2 utilities to copy an existing TDBM
-- or GBM database to the DB2 database created by this SPUFI script,
-- then you must use the dn trunc size of the DB2 database you are
-- copying. Otherwise the DB2 copy utility will produce unreliable
-- results.
--
-- Change -ENTRY_DN_SIZE- to the maximum size of a DN. This value
-- includes the null terminator, so the actual maximum length of a DN
-- will be one less than this value.
--
-- The suggested size is 512.
--
-- Note: The -ENTRY_DN_SIZE- value must be less than or equal to
-- the maximum size of the change DN column (CHNGDN) in the
-- replication table (DIR_REPENTRY). -ENTRY_DN_SIZE- will be
-- used to define both column sizes.
--
-- *************************
-- Storage Group Information
-- *************************
-- Change -SYSDEFLT- to the storage group you want to contain the
-- LDAP DB2 tablespaces. Use SYSDEFLT to choose the default storage
-- group.
-- NOTE: The values provided below for PRIQTY and SECQTY probably need
-- to be modified depending on the projected size of the
-- Directory information to be stored.
--
-- *********************************************************************
-- Use the following statements if you need to delete your LDAP Server
-- DB2 database and tablespaces in SPUFI.
--
-- NOTE: You need to remove the '--' from each line before you can run
-- these statements.
--
-- Change -DB_NAME- to the LDAP database name you want to delete.
--
-- Change the following names to the LDAP tablespace names you want to
-- delete:
--   -ENTRYTS-
--   -ENTRYTS-
--   -LATTRTS-
--   -SEARCHTS-
--   -MISCTS-
--   -DESCRIPTS-
--   -REPTS-
--*********************************************************************
--DROP TABLESPACE -DB_NAME--ENTRYTS--;
--DROP TABLESPACE -DB_NAME--ENTRYTS--;
--DROP TABLESPACE -DB_NAME--ENTRYTS--;
--DROP TABLESPACE -DB_NAME--ENTRYTS--;
--DROP TABLESPACE -DB_NAME--ENTRYTS--;
--DROP TABLESPACE -DB_NAME--ENTRYTS--;
--DROP TABLESPACE -DB_NAME--ENTRYTS--;
--DROP TABLESPACE -DB_NAME--ENTRYTS--;
--DROP TABLESPACE -DB_NAME--ENTRYTS--;
--DROP DATABASE -DB_NAME--;
--COMMIT;
--
-- ************************
-- Create the LDAP database
-- ************************
CREATE DATABASE -DB_NAME-- STOGROUP -SYSDEFLT- CCSID EBCDIC;
--
-- Create the LDAP entry tablespace
--
CREATE TABLESPACE -ENTRYTS- IN -DB_NAME-
   USING STOGROUP -SYSDEFLT-
   PRIQTY 14400
   SECQTY 7200;
BUFFERPOOL -ENTRYTS_BP0-;

-- ****************************
-- Create the LDAP long entry tablespace
-- ****************************
CREATE TABLESPACE -ENTRYTS- IN -DB_NAME-
    USING STOGROUP -SYSDEFLT-
    PRIQTY 14400
    SECQTY 7200
    BUFFERPOOL -ENTRYTS_BP0-;

-- **************************
-- Create the LDAP long attr tablespace
-- **************************
CREATE TABLESPACE -LATTRTS- IN -DB_NAME-
    USING STOGROUP -SYSDEFLT-
    PRIQTY 14400
    SECQTY 7200
    BUFFERPOOL -LATTRTS_BP0-;

-- **************************
-- Create the LDAP search tablespace
-- **************************
CREATE TABLESPACE -SEARCHTS- IN -DB_NAME-
    USING STOGROUP -SYSDEFLT-
    PRIQTY 14400
    SECQTY 7200
    BUFFERPOOL -SEARCHTS_BP0-;

-- **************************
-- Create the LDAP misc tablespace
-- **************************
CREATE TABLESPACE -MISCTS- IN -DB_NAME-
    SEGSIZE 4
    USING STOGROUP -SYSDEFLT-
    PRIQTY 14400
    SECQTY 7200
    LOCKSIZE ROW
    BUFFERPOOL -MISCTS_BP0-;

-- **************************
-- Create the LDAP descendants tablespace
-- **************************
CREATE TABLESPACE -DESCTS- IN -DB_NAME-
    USING STOGROUP -SYSDEFLT-
    PRIQTY 14400
    SECQTY 7200
    BUFFERPOOL -DESCTS_BP0-;

-- **************************
-- Create the LDAP replica tablespace
-- **************************
CREATE TABLESPACE -REPTS- IN -DB_NAME-
    USING STOGROUP -SYSDEFLT-
    PRIQTY 14400
    SECQTY 7200
    BUFFERPOOL -REPTS_BP0-;

-- *****************
-- Create the DB2 tables
-- *****************
-- **************************
-- Create the DIR_ENTRY table
-- **************************
CREATE TABLE -DB_USERID-.DIR_ENTRY (  
    EID DECIMAL(15, 0) NOT NULL,
### DSTDBMDB SPUI

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Nullable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>EID</strong></td>
<td>DECIMAL(15,0)</td>
<td>NOT NULL</td>
<td>Unique identifier for the entry.</td>
</tr>
<tr>
<td><strong>SEQ</strong></td>
<td>INTEGER</td>
<td>NOT NULL</td>
<td>Sequence number for the entry.</td>
</tr>
<tr>
<td><strong>ENTRYDATA</strong></td>
<td>LONG VARCHAR</td>
<td></td>
<td>Data associated with the entry.</td>
</tr>
</tbody>
</table>

**Create the DIR_LONGENTRY table**

```sql
CREATE TABLE -DB_USERID-.DIR_LONGENTRY
(
    EID DECIMAL(15,0) NOT NULL,
    SEQ INTEGER NOT NULL,
    ENTRYDATA LONG VARCHAR FOR BIT DATA,
    PRIMARY KEY( EID, SEQ )
) IN -DB_NAME-.-ENTRYTS-;
```

**Create the DIR_LONGATTR table**

```sql
CREATE TABLE -DB_USERID-.DIR_LONGATTR
(
    EID DECIMAL(15,0) NOT NULL,
    ATTR_ID INTEGER NOT NULL,
    VALUENUM INTEGER NOT NULL,
    SEQ INTEGER NOT NULL,
    ATTRDATA LONG VARCHAR FOR BIT DATA,
    PRIMARY KEY( EID, ATTR_ID, VALUENUM, SEQ )
) IN -DB_NAME-.-LATTRTS-;
```

**Create the DIR_MISC table**

```sql
CREATE TABLE -DB_USERID-.DIR_MISC
(
    NEXT_EID DECIMAL(15,0),
    NEXT_ATTR_ID INTEGER,
    DB_VERSION CHAR(10),
    SCHEMA_TIMESTAMP TIMESTAMP,
    PARTITIONED_EID CHAR(1)
) IN -DB_NAME-.-MISCTS-;
```

**Create the DIR_CACHE table**

```sql
CREATE TABLE -DB_USERID-.DIR_CACHE
(
    CACHE_NAME CHAR(25) NOT NULL,
    MODIFY_TIMESTAMP TIMESTAMP NOT NULL,
    PRIMARY KEY( CACHE_NAME, MODIFY_TIMESTAMP )
) IN -DB_NAME-.-MISCTS-;
```

**Create the DIR_ATTRID table**

```sql
CREATE TABLE -DB_USERID-.DIR_ATTRID
(
    ATTR_ID INTEGER,
    ATTR_NOID VARCHAR(200) NOT NULL,
    PRIMARY KEY( ATTR_NOID )
) IN -DB_NAME-.-MISCTS-;
```
CREATE TABLE -DB_USERID-.DIR_DESC (DEID DECIMAL(15,0) NOT NULL, AEID DECIMAL(15,0) NOT NULL, PRIMARY KEY (DEID, AEID)) IN -DB_NAME-.DESCTS;

CREATE TABLE -DB_USERID-.DIR_SEARCH (EID DECIMAL(15,0) NOT NULL, ATTR_ID INTEGER NOT NULL, VALUE CHAR(-SEARCH_TRUNC_SIZE-) FOR BIT DATA, LVALUE LONG VARCHAR FOR BIT DATA) IN -DB_NAME-.SEARCHTS;

CREATE TABLE -DB_USERID-.DIR_REPLICA (REPID DECIMAL(15,0) NOT NULL, CHNGID DECIMAL(15,0) NOT NULL, REPCAPS INTEGER NOT NULL, PRIMARY KEY (REPID)) IN -DB_NAME-.MISCTS;

CREATE TABLE -DB_USERID-.DIR_REPENTRY (CHNGID DECIMAL(15,0) NOT NULL, CHNGFLAGS INTEGER NOT NULL, CHNGSIZE INTEGER NOT NULL, CHNGDN VARCHAR(-ENTRY_DN_SIZE-) FOR BIT DATA, CHNGDATA LONG VARCHAR FOR BIT DATA, PRIMARY KEY (CHNGID)) IN -DB_NAME-.REPTS;

CREATE TABLE -DB_USERID-.DIR_LONGREPENTRY (CHNGID DECIMAL(15,0) NOT NULL, CHNGSEQ INTEGER NOT NULL, CHNGDATA LONG VARCHAR FOR BIT DATA, PRIMARY KEY (CHNGID, CHNGSEQ)) IN -DB_NAME-.REPTS;

CREATE TABLE -DB_USERID-.DIR_EID (PARTID DECIMAL(15,0) NOT NULL, NEXT_EID DECIMAL(15,0) NOT NULL, MODIFY_TIMESTAMP TIMESTAMP NOT NULL, PRIMARY KEY (PARTID)) IN -DB_NAME-.MISCTS;

CREATE TABLE -DB_USERID-.DIR_REPLSTATUS (AGREEMENTEID DECIMAL(15,0) NOT NULL, CHANGEID INTEGER NOT NULL, PRIMARY KEY (AGREEMENTEID, CHANGEID)) IN -DB_NAME-.REPTS;
CREATE TABLE -DB_USERID-.DIR_REPLCHANGE ( 
  CHANGEID INTEGER NOT NULL,
  CONTEXTEID DECIMAL(15, 0) NOT NULL,
  CHANGESIZE INTEGER NOT NULL,
  CHANGEDATA LONG VARCHAR FOR BIT DATA,
  PRIMARY KEY( CHANGEID, CONTEXTEID ) )
IN -DB_NAME-.-REPTS-;

CREATE TABLE -DB_USERID-.DIR_LONGREPLCHANGE ( 
  CHANGEID INTEGER NOT NULL,
  CONTEXTEID DECIMAL(15, 0) NOT NULL,
  CHANGESEQ INTEGER NOT NULL,
  CHANGEDATA LONG VARCHAR FOR BIT DATA,
  PRIMARY KEY( CHANGEID, CONTEXTEID, CHANGESEQ ) )
IN -DB_NAME-.-REPTS-;

CREATE TABLE -DB_USERID-.DIR_REPLERROR ( 
  ERRORID INTEGER NOT NULL,
  AGREEMENTEID DECIMAL(15, 0) NOT NULL,
  ERRORSIZE INTEGER NOT NULL,
  ERRORDATA LONG VARCHAR FOR BIT DATA,
  PRIMARY KEY( ERRORID ) )
IN -DB_NAME-.-REPTS-;

CREATE TABLE -DB_USERID-.DIR_LONGREPLERROR ( 
  ERRORID INTEGER NOT NULL,
  AGREEMENTEID DECIMAL(15, 0) NOT NULL,
  ERRORSEQ INTEGER NOT NULL,
  ERRORDATA LONG VARCHAR FOR BIT DATA,
  PRIMARY KEY( ERRORID, ERRORSEQ ) )
IN -DB_NAME-.-REPTS-;

-- Miscellaneous Information
-- All indexes have been defined DEFER YES, which means they need to be
-- recovered at some point. It is suggested to do the recovery after
-- the database has been populated for databases with large amounts of
-- data. Use of this option is strictly optional though.
-- To NOT use the DEFER YES option, simply remove DEFER YES globally.
--
CREATE UNIQUE INDEX -DB_USERID-.DIR_ENTRYX0
ON -DB_USERID-.DIR_ENTRY( EID )
USING STOGROUP -SYSDEFLT-
PRIQTY 720
SECQTY 720
CLUSTER
CREATE INDEX -DB_USERID-.DIR_ENTRYX1
ON -DB_USERID-.DIR_ENTRY( PEID, EID )
USING STOGROUP -SYSDEFLT-
PRIQTY 720
SECQTY 720
DEFER YES;

CREATE INDEX -DB_USERID-.DIR_ENTRYX2
ON -DB_USERID-.DIR_ENTRY( EID, DN_TRUNC )
USING STOGROUP -SYSDEFLT-
PRIQTY 7200
SECQTY 3600
DEFER YES;

CREATE INDEX -DB_USERID-.DIR_ENTRYX3
ON -DB_USERID-.DIR_ENTRY( DN_TRUNC, EID )
USING STOGROUP -SYSDEFLT-
PRIQTY 7200
SECQTY 3600
DEFER YES;

-- ********************************
-- Create the DIR_LONGENTRY indexes
-- ********************************
CREATE UNIQUE INDEX -DB_USERID-.DIR_LONGENTRYX1
ON -DB_USERID-.DIR_LONGENTRY( EID, SEQ )
USING STOGROUP -SYSDEFLT-
PRIQTY 720
SECQTY 720
CLUSTER
DEFER YES;

-- ********************************
-- Create the DIR_LONGATTR indexes
-- ********************************
CREATE UNIQUE INDEX -DB_USERID-.DIR_LONGATTRX1
ON -DB_USERID-.DIR_LONGATTR( EID, ATTR_ID, VALUENUM, SEQ )
USING STOGROUP -SYSDEFLT-
PRIQTY 720
SECQTY 720
CLUSTER
DEFER YES;

-- ********************************
-- Create the DIR_CACHE indexes
-- ********************************
CREATE UNIQUE INDEX -DB_USERID-.DIR_CACHEX1
ON -DB_USERID-.DIR_CACHE( CACHE_NAME, MODIFY_TIMESTAMP )
USING STOGROUP -SYSDEFLT-
CLUSTER
DEFER YES;

-- ********************************
-- Create the DIR_ATTRID indexes
-- ********************************
CREATE UNIQUE INDEX -DB_USERID-.DIR_ATTRIDX1
ON -DB_USERID-.DIR_ATTRID( ATTR_NOID )
USING STOGROUP -SYSDEFLT-
CLUSTER
DEFER YES;

-- ********************************
-- Create the DIR_DESC indexes
-- ********************************
CREATE UNIQUE INDEX -DB_USERID-.DIR_DESCX1
ON -DB_USERID-.DIR_DESC( DEID, AEID )
USING STOGROUP -SYSDEFLT-
PRIQTY 7200
SECQTY 3600
CLUSTER
DEFER YES;

CREATE INDEX -DB_USERID-.DIR_SEARCHX1
ON -DB_USERID-.DIR_SEARCH( ATTR_ID, VALUE, EID )
USING STOGROUP -SYSDEFLT-
PRIQTY 14400
SECQTY 7200
DETER YES;

CREATE INDEX -DB_USERID-.DIR_SEARCHX2
ON -DB_USERID-.DIR_SEARCH( EID, ATTR_ID )
USING STOGROUP -SYSDEFLT-
PRIQTY 7200
SECQTY 3600
CLUSTER
DETER YES;

CREATE UNIQUE INDEX -DB_USERID-.DIR_REPLICAX1
ON -DB_USERID-.DIR_REPLICA( REPID )
USING STOGROUP -SYSDEFLT-
CLUSTER
DETER YES;

CREATE UNIQUE INDEX -DB_USERID-.DIR_REPENTRYX1
ON -DB_USERID-.DIR_REPENTRY( CHNGID )
USING STOGROUP -SYSDEFLT-
CLUSTER
DETER YES;

CREATE UNIQUE INDEX -DB_USERID-.DIR_LONGREPX1
ON -DB_USERID-.DIR_LONGREPEX( CHNGID, CHNGSEQ )
USING STOGROUP -SYSDEFLT-
CLUSTER
DETER YES;

CREATE UNIQUE INDEX -DB_USERID-.DIR_EIDX1
ON -DB_USERID-.DIR_EID( PARTID )
USING STOGROUP -SYSDEFLT-
CLUSTER
DETER YES;

CREATE UNIQUE INDEX -DB_USERID-.DIR_REPLSTATUSX1
ON -DB_USERID-.DIR_REPLSTATUS( AGREEMENTEID )
USING STOGROUP -SYSDEFLT-
| CREATE UNIQUE INDEX -DB_USERID-.DIR_REPLCHANGEX1 ON -DB_USERID-.DIR_REPLCHANGE( CHANGEID, CONTEXTEID ) USING STOGROUP -SYSDEFLT- CLUSTER DEFER YES; |
| CREATE INDEX -DB_USERID-.DIR_REPLCHANGEX2 ON -DB_USERID-.DIR_REPLCHANGE( CONTEXTEID ) USING STOGROUP -SYSDEFLT- DEFER YES; |
| CREATE INDEX -DB_USERID-.DIR_REPLCHANGEX3 ON -DB_USERID-.DIR_REPLCHANGE( CHANGEID ) USING STOGROUP -SYSDEFLT- DEFER YES; |
| CREATE UNIQUE INDEX -DB_USERID-.DIR_LONGREPLCHANGEX1 ON -DB_USERID-.DIR_LONGREPLCHANGE( CHANGEID, CONTEXTEID, CHANGSEQ ) USING STOGROUP -SYSDEFLT- CLUSTER DEFER YES; |
| CREATE INDEX -DB_USERID-.DIR_LONGREPLCHANGEX2 ON -DB_USERID-.DIR_LONGREPLCHANGE( AGREEMENTEID ) USING STOGROUP -SYSDEFLT- DEFER YES; |
| CREATE UNIQUE INDEX -DB_USERID-.DIR_LONGREPLERRORX1 ON -DB_USERID-.DIR_LONGREPLError( ERRORID ) USING STOGROUP -SYSDEFLT- CLUSTER DEFER YES; |
| CREATE INDEX -DB_USERID-.DIR_LONGREPLERRORX2 ON -DB_USERID-.DIR_LONGREPLError( AGREEMENTEID ) USING STOGROUP -SYSDEFLT- DEFER YES; |
| CREATE UNIQUE INDEX -DB_USERID-.DIR_LONGREPLERRORX1 ON -DB_USERID-.DIR_LONGREPLError( ERRORID, ERRORSEQ ) USING STOGROUP -SYSDEFLT- CLUSTER DEFER YES; |
| CREATE INDEX -DB_USERID-.DIR_LONGREPLERRORX2 ON -DB_USERID-.DIR_LONGREPLError( AGREEMENTEID ) USING STOGROUP -SYSDEFLT- DEFER YES; |
| -- Commit all the above SQL statements |
| COMMIT; |

-- Use the following statement if you need to grant EXECUTE privilege
-- on the DB2 CLI plan to the user running the LDAP server.
--
-- NOTE: You need to remove the '--' from each line before you can run
-- these statements.
--
-- Change -DB_PLAN- to the DB2 CLI plan name.
-- Change -LDAP_USERID- to the user ID running the LDAP server.
-- *********************************************************************
--GRANT EXECUTE ON PLAN -DB_PLAN- TO -LDAP_USERID-;
--COMMIT;
-- *********************************************************************

-- Use the following statements if you need to grant SQL SELECT
-- privileges on DB2 catalog tables to the user running the LDAP server.
--
-- NOTE: You need to remove the '--' from each line before you can run
-- these statements.
--
-- Change -LDAP_USERID- to the user ID running the LDAP server.
-- *********************************************************************
--GRANT SELECT ON SYSIBM.SYSCOLUMNS TO -LDAP_USERID-;
--GRANT SELECT ON SYSIBM.SYSCOLDIST TO -LDAP_USERID-;
--GRANT SELECT ON SYSIBM.SYSTABLES TO -LDAP_USERID-;
--GRANT SELECT ON SYSIBM.SYSTABLEPART TO -LDAP_USERID-;
--GRANT SELECT ON SYSIBM.SYSKEYS TO -LDAP_USERID-;
--GRANT SELECT ON SYSIBM.SYSTABLESPACE TO -LDAP_USERID-;
--GRANT SELECT ON SYSIBM.SYSINDEXPART TO -LDAP_USERID-;
--COMMIT;
-- *********************************************************************

-- Use the following statement if you need to grant DBADM privilege
-- on the DB2 database to the user running the LDAP server.
--
-- NOTE: You need to remove the '--' from each line before you can run
-- these statements.
--
-- Change -DB_NAME- to the LDAP database name.
-- Change -LDAP_USERID- to the user ID running the LDAP server.
-- *********************************************************************
--GRANT DBADM ON DATABASE -DB_NAME- TO -LDAP_USERID-;
--COMMIT;
The TDBMMGRT SPUFI file

-- ******************************************************************
-- *
-- * Licensed Materials - Property of IBM
-- * 5694-A01
-- * Copyright IBM Corp. 2006, 2009
-- *
-- ******************************************************************
-- Use the SPUFI function to execute these SQL statements.
--
-- Database, table, tablespace and storage class values must be
-- retrieved from your z/OS LDAP Server DB2 configuration. Use
-- the following DB2 command to query the information:
--
-- 'DISPLAY DATABASE(GLDDB)'
--
-- NOTE: GLDDB is the default database name, your database name may
-- differ.
--
-- Database Name Information
-- ******************************************************************
-- Change any instance of -DB2_NAME- to the name of the LDAP
-- database name you want to migrate. Be sure this name matches what
-- was defined for the database name in the z/OS LDAP Server
-- configuration file.
--
-- The default configuration sets this value to GLDDB. But this
-- value should be verified.
--
-- Database Owner Information
-- ******************************************************************
-- Change any instance of -DB2_USERID- to the MVS database owner id
-- defined as the owner id for the z/OS LDAP Server DB2
-- configuration. This id is the highlevel qualifier for the tables.
--
-- The default configuration sets this value to GLDSRV. But this
-- value should be verified.
--
-- LDAP Server User ID Information
-- ******************************************************************
-- Change any instance of -LDAP_USERID- to the user ID for the LDAP
-- server to run under.
--
-- The default configuration sets this value to GLDSRV. But this
-- value should be verified.
--
-- Tablespace Information
-- ******************************************************************
-- Change any instance of -MISC_TABLESPACE- to the LDAP miscellaneous
-- tablespace as defined in the z/OS LDAP Server DB2 configuration.
--
-- The default configuration sets this value to MISCTS. But this
-- value should be verified.
--
-- Change any instance of -REPLICA_TABLESPACE- to the LDAP replica
-- tablespace as defined in the z/OS LDAP Server DB2 configuration.
--
-- The default configuration sets this value to REPTS. But this
-- value should be verified.
--
-- ******************************************************************
-- Storage Group Information
-- ******************************************************************
-- Change any instance of -STORAGEGROUP- to the storage group as
-- defined in the z/OS LDAP Server DB2 configuration.
--
-- The default configuration sets this value to SYSDEFLT. But this
-- value should be verified.
--
-- ******************************************************************

-- TABLE OF CONTENTS
-- ******************************************************************
-- SECTION TITLE
-- ******************************************************************
-- 01 SQL statements for migrating from z/OS Integrated
-- Security Services LDAP server to the latest version of
-- IBM Tivoli Directory Server for z/OS LDAP server.
-- 02 SQL for using the partitioned entry identifier assignment
-- algorithm. This is generally needed for users migrating
-- from IBM Tivoli Directory Server for z/OS V1R8 to IBM
-- 03 SQL for using advanced replication support.
-- This is generally needed for users migrating
-- from IBM Tivoli Directory Server for z/OS V1R10 to IBM
-- Tivoli Directory Server for z/OS V1R11.

-- *** SECTION 01: ***
-- *** SQL statements for migrating from z/OS Integrated ***
-- *** Security Services LDAP server to the latest version of ***
-- *** IBM Tivoli Directory Server for z/OS LDAP server. ***
-- ******************************************************************
-- The following SQL statements, created for the z/OS Integrated
-- Security Services LDAP server, will alter your DB2 database tables
-- such that they will be supported by the latest version of IBM
-- Tivoli Directory Server for z/OS LDAP server.
--
-- ***** SPECIAL NOTICE ***** ***** SPECIAL NOTICE *****
-- The DIR_CHANGE table used for replication must be empty before the
-- backend is started on the IBM Tivoli Directory Server for z/OS
-- LDAP server. The SQL statements in this file will empty the table
-- if it is not empty, but this will result in one or more replicas
-- that are no longer synchronized with this master server.
--
-- To verify that the DIR_CHANGE table is empty, from a separate
-- file, execute the following SQL statement using SPUFI:
--
-- SELECT * FROM -DB2_USERID-.DIR_CHANGE;
--
-- where -DB2_USERID- is changed to the MVS database owner id, as
-- previously described above.
--
-- After execution, if the row count returns 0, then all replication
-- changes have been made and the table is ready to be dropped.
-- Otherwise, refer to the z/OS Integrated Security Services LDAP
-- Server Administration and Use documentation to ensure all
-- replication changes have been made and then verify that the table
-- is empty.

-- If replication does not take place and the table does not become
-- empty, then all the replication tables will be emptied by the
-- SQL commands below. In this case replication updates to one or
-- more replicas will have been lost and it will be necessary to
-- synchronize these replicas with the master server. Refer to the
-- IBM Tivoli Directory Server Administration and Use for z/OS
-- publication for information about replica synchronization.

-- Empty the replication tables
DELETE FROM -DB2_USERID-.DIR_REGISTER;
DELETE FROM -DB2_USERID-.DIR_PROGRESS;
DELETE FROM -DB2_USERID-.DIR_CHANGE;
DELETE FROM -DB2_USERID-.DIR_LONGCHANGE;

-- Drop the replication tables

-- NOTE: these SQL statements are commented out. The IBM Tivoli
-- Directory Server for z/OS LDAP server will ignore these tables.
-- Keeping these tables allows you to roll-back DB2 data to the z/OS
-- Integrated Security Services LDAP server database. It also allows
-- sharing the database between the two servers. Remove the '--'
-- characters from the following SQL statements if you want to drop
-- the replication tables.

-- NOTE: New replica tables and indexes will be created below.

-- Alter the misc tablespace to set locksize row
ALTER TABLESPACE -DB2_NAME-.MISC_TABLESPACE
    LOCKSIZE ROW;

-- Alter the misc table to add the schema timestamp column

-- NOTE: This SQL statement is commented out. The IBM Tivoli
-- Directory Server for z/OS LDAP server will automatically alter the
-- DIR_MISC table to add the new schema timestamp column if it is not
-- already there. As an alternative, you can create the new column
-- here by removing the '--' characters from the following SQL
-- statement.

-- Alter the misc table to add the partitioned eid column

-- NOTE: This SQL statement is commented out. The IBM Tivoli
-- Directory Server for z/OS LDAP server will automatically alter the
-- DIR_MISC table to add the new partitioned eid column if it is not
-- already there. As an alternative, you can create the new column
-- here by removing the '--' characters from the following SQL
-- statement.
-- ALTER TABLE -DB2_USERID-.DIR_MISC
--   ADD PARTITIONED_EID CHAR(1)
--   WITH DEFAULT NULL;
--
-- Create the new replica table
--
CREATE TABLE -DB2_USERID-.DIR_REPLICA (
    REPID DECIMAL(15, 0) NOT NULL,
    CHNGID DECIMAL(15, 0) NOT NULL,
    REPCAPS INTEGER NOT NULL,
    PRIMARY KEY( REPID )
) IN -DB2_NAME-.MISC_TABLESPACE;
--
-- Create the new replica entry table
--
-- Note: The maximum size of the change DN column (CHNGDN) in the
-- replication table (DIR_REPENTRY) must be greater than
-- or equal to the maximum size of the DN column in the entry
-- table (DIR_ENTRY). The default maximum size of 512 is set
-- here.
--
CREATE TABLE -DB2_USERID-.DIR_REPENTRY (
    CHNGID DECIMAL(15, 0) NOT NULL,
    CHNGFLAGS INTEGER NOT NULL,
    CHNGSIZE INTEGER NOT NULL,
    CHNGDN VARCHAR(512) FOR BIT DATA,
    CHNGDATA LONG VARCHAR FOR BIT DATA,
    PRIMARY KEY( CHNGID )
) IN -DB2_NAME-.REPLICA_TABLESPACE;
--
-- Create the new long replica entry table
--
CREATE TABLE -DB2_USERID-.DIR_LONGREPENTRY (
    CHNGID DECIMAL(15, 0) NOT NULL,
    CHNGSEQ INTEGER NOT NULL,
    CHNGDATA LONG VARCHAR FOR BIT DATA,
    PRIMARY KEY( CHNGID, CHNGSEQ )
) IN -DB2_NAME-.REPLICA_TABLESPACE;
--
-- Create the entry identifier table
--
CREATE TABLE -DB2_USERID-.DIR_EID (
    PARTID DECIMAL(15, 0) NOT NULL,
    NEXT_EID DECIMAL(15, 0) NOT NULL,
    MODIFY_TIMESTAMP TIMESTAMP NOT NULL,
    PRIMARY KEY( PARTID )
) IN -DB2_NAME-.MISC_TABLESPACE;
--
-- Create the new replica index
--
CREATE UNIQUE INDEX -DB2_USERID-.DIR_REPLICAX1
    ON -DB2_USERID-.DIR_REPLICA( REPID )
    USING STOGROUP -STORAGEGROUP- CLUSTER
    DEFER YES;
--
-- Create the new replica entry index
--
CREATE UNIQUE INDEX -DB2_USERID-.DIR_REPENTRYX1
    ON -DB2_USERID-.DIR_REPENTRY( CHNGID )
USING STOGROUP -STORAGEGROUP-
CLUSTER
DEFER YES;

-- Create the new long replica entry index
CREATE UNIQUE INDEX -DB2_USERID-.DIR_LONGREPX1
ON -DB2_USERID-.DIR_LONGREPEXT( CHNGID, CHNGSEQ )
USING STOGROUP -STORAGEGROUP-
CLUSTER
DEFER YES;

-- Create the entry identifier index
CREATE UNIQUE INDEX -DB2_USERID-.DIR_EIDX1
ON -DB2_USERID-.DIR_EID( PARTID )
USING STOGROUP -STORAGEGROUP-
CLUSTER
DEFER YES;

-- Update the DB_VERSION value to 4.0 in the DIR_MISC table.
--
-- NOTE: This SQL statement is commented out. The DB_VERSION value
-- indicates the level of the TDBM database. Updating the
-- DB_VERSION value to 4.0 enables the backend to use the enhanced
-- sysplex and replication support in the IBM Tivoli Directory Server
-- for z/OS LDAP server. However, it will prevent the backend
-- database from being shared with a z/OS Integrated Security
-- Services LDAP server. If you do not plan to share the database,
-- you can update the DB_VERSION value here by removing the '--'
-- characters from the following SQL statement.
--
-- UPDATE -DB2_USERID-.DIR_MISC
-- SET DB_VERSION='4.0';
--
-- Commit the changes made in this section
--
COMMIT;

-- Grant SQL SELECT privileges on DB2 catalog tables to the user
-- running the LDAP server.
--
-- NOTE: You need to remove the '--' from each line before you can run
-- these statements.
--
-- Change -LDAP_USERID- to the user ID running the LDAP server.
--
--GRANT SELECT ON SYSCOLUMNS TO -LDAP_USERID-;
--GRANT SELECT ON SYSCOLDIST TO -LDAP_USERID-;
--GRANT SELECT ON SYSTABLES TO -LDAP_USERID-;
--GRANT SELECT ON SYSTABLEPART TO -LDAP_USERID-;
--GRANT SELECT ON SYSKEYS TO -LDAP_USERID-;
--GRANT SELECT ON SYSTABLESPACE TO -LDAP_USERID-;
--GRANT SELECT ON SYSINDEXPART TO -LDAP_USERID-;
--COMMIT;

-- *** SECTION 02: ***
-- *** ***
-- *** SQL for using the partitioned entry identifier assignment ***

Appendix B. SPUFI files  759
-- *** algorithm. This is generally needed for users migrating ***
-- *** from IBM Tivoli Directory Server for z/OS V1R8 to IBM ***
-- *** Tivoli Directory Server for z/OS V1R10. ***
-- ******************************************************************
-- The following SQL statements, created for the IBM Tivoli Directory
-- Server for z/OS V1R8, will alter your DB2 database tables such
-- that they will support the partitioned entry identifier assignment
-- algorithm.
--
-- You SHOULD NOT run the following SQL statements if your DB2
-- database tables were built to support IBM Tivoli Directory Server
-- for z/OS V1R10 or if your DB2 database tables already support the
-- partitioned entry identifier assignment algorithm.
--
-- If you are unsure whether you need to run the SQL statements in
-- this section, from a separate file, execute the following SQL
-- statement using SPUFI:
--
-- SELECT * FROM SYSCAT.SYSTABLES WHERE
-- CREATOR='-DB2_USERID-' AND NAME='DIR_EID';
--
-- where -DB2_USERID- is changed to the MVS database owner id, as
-- previously described above.
--
-- After execution, if the row count returns 0, then your DB2
-- database tables DO NOT support the partitioned entry identifier
-- assignment algorithm. If you want to use the partitioned entry
-- identifier assignment algorithm, or if you want to run IBM Tivoli
-- Directory Server for z/OS V1R10, you need to execute the
-- following SQL statements.
--
-- To ensure a successful update of your DB2 database tables, you
-- must comment out all SQL statements in other sections of this file
-- prior to running the following SQL statements.
--
-- ******************************************************************
-- Alter the misc tablespace to set locksize row
--
-- NOTE: This SQL statement is commented out. To set the misc
-- tablespace to locksize row, remove the '--' characters from the
-- following SQL statement.
--
-- ALTER TABLESPACE -DB2_NAME--.MISC_TABLESPACE-
-- LOCKSIZE ROW;
--
-- ******************************************************************
-- Alter the misc table to add the partitioned eid column
--
-- NOTE: This SQL statement is commented out. The IBM Tivoli
-- Directory Server for z/OS LDAP server will automatically alter the
-- DIR_MISC table to add the new partitioned eid column if it is not
-- already there. As an alternative, you can create the new column
-- here by removing the '--' characters from the following SQL
-- statement.
--
-- ALTER TABLE -DB2_USERID--.DIR_MISC
-- ADD PARTITIONED_EID CHAR(1)
-- WITH DEFAULT NULL;
--
-- ******************************************************************
-- Create the entry identifier table
--
-- NOTE: This SQL statement is commented out. To create the entry
-- identifier table, remove the '--' characters from the following
-- SQL statement.
--
-- CREATE TABLE -DB2_USERID--.DIR_EID {
-- PARTID DECIMAL(15, 0) NOT NULL,
-- NEXT_EID DECIMAL(15, 0) NOT NULL,
-- MODIFY_TIMESTAMP TIMESTAMP NOT NULL,
-- PRIMARY KEY( PARTID )
-- IN -DB2_NAME-.MISC_TABLESPACE;

-- ******************************************************************
-- Create the entry identifier index
--
-- NOTE: This SQL statement is commented out. To create the entry
-- identifier index, remove the '--' characters from the following
-- SQL statement.
--
-- CREATE UNIQUE INDEX -DB2_USERID-.DIR_EIDX1
-- ON -DB2_USERID-.DIR_EID( PARTID )
-- USING STOGROUP -STORAGEGROUP-
-- CLUSTER
-- DEFER YES;
--
-- ******************************************************
-- Commit the changes made in this section
--
-- NOTE: This SQL statement is commented out. Remove the '--'
-- characters from the following SQL statement to commit the changes
-- made in this section.
--
-- COMMIT;
--
-- ******************************************************
-- Grant SQL SELECT privileges on DB2 catalog tables to the user
-- running the LDAP server.
--
-- NOTE: You need to remove the '--' from each line before you can run
-- these statements.
--
-- Change -LDAP_USERID- to the user ID running the LDAP server.
--
-- GRANT SELECT ON SYSIBM.SYSCOLUMNS TO -LDAP_USERID-;
--GRANT SELECT ON SYSIBM.SYSCOLDIST TO -LDAP_USERID-;
--GRANT SELECT ON SYSIBM.SYSTABLES TO -LDAP_USERID-;
--GRANT SELECT ON SYSIBM.SYSTABLEPART TO -LDAP_USERID-;
--GRANT SELECT ON SYSIBM.SYSKEYS TO -LDAP_USERID-;
--GRANT SELECT ON SYSIBM.SYSTABLESPACE TO -LDAP_USERID-;
--GRANT SELECT ON SYSIBM.SYSINDEXPART TO -LDAP_USERID-;
--COMMIT;
--
--
-- ******************************************************
-- *** SECTION 03: ***
-- *** ***
-- *** SQL for using the advanced replication support. ***
-- *** This is generally needed for users migrating ***
-- *** from IBM Tivoli Directory Server for z/OS V1R10 to IBM ***
-- *** Tivoli Directory Server for z/OS V1R11. ***
-- The following SQL statements, created for the IBM Tivoli Directory
-- Server for z/OS V1R10, will alter your DB2 database tables such
-- that they will support advanced replication.
--
-- You SHOULD NOT run the following SQL statements if your DB2
-- database tables were built to support IBM Tivoli Directory Server
-- for z/OS V1R11 or if your DB2 database tables already support
-- advanced replication.
--
-- If you are unsure whether you need to run the SQL statements in
-- this section, from a separate file, execute the following SQL
-- statement using SPUFI:
--
-- SELECT * FROM SYSIBM.SYSTABLES WHERE
-- CREATOR='-DB2_USERID-' AND NAME='DIR_REPLSTATUS';
--
-- where -DB2_USERID- is changed to the MVS database owner id, as
-- previously described above.
--
-- After execution, if the row count returns 0, then your DB2
-- database tables DO NOT support advanced replication.
-- If you want to use advanced replication, you need to execute
-- the following SQL statements.
--
-- To ensure a successful update of your DB2 database tables, you
-- must comment out all SQL statements in other sections of this file
-- prior to running the following SQL statements.
--
-- *******************************************************************************
-- Create the DIR_REPLSTATUS table
--
-- NOTE: This SQL statement is commented out. Remove the '--'
-- characters from the following SQL statement to commit the changes
-- made in this section.
-- *******************************************************************************
--CREATE TABLE -DB2_USERID-.DIR_REPLSTATUS (    
-- AGREEMENTEID DECIMAL (15 , 0) NOT NULL,
-- CHANGEID INTEGER NOT NULL,
-- REPPLICACAPS CHAR (64) FOR BIT DATA,
-- PRIMARY KEY( AGREEMENTEID ) )
--IN -DB2_NAME-.MISC_TABLESPACE-;
-- *******************************************************************************
-- Create the DIR_REPLCHANGE table
--
-- NOTE: This SQL statement is commented out. Remove the '--'
-- characters from the following SQL statement to commit the changes
-- made in this section.
-- *******************************************************************************
--CREATE TABLE -DB2_USERID-.DIR_REPLCHANGE (    
-- CHANGEID INTEGER NOT NULL,
-- CONTEXTEID DECIMAL (15 , 0) NOT NULL,
-- CHANGESIZE INTEGER NOT NULL,
-- CHANGEDATA LONG VARCHAR FOR BIT DATA,
-- PRIMARY KEY( CHANGEID, CONTEXTEID ) )
--IN -DB2_NAME-.REPLICA_TABLESPACE-;
-- *******************************************************************************
-- Create the DIR_LONGREPLCHANGE table
--
-- NOTE: This SQL statement is commented out. Remove the '--'
-- characters from the following SQL statement to commit the changes
-- made in this section.
-- *******************************************************************************
--CREATE TABLE -DB2_USERID-.DIR_LONGREPLCHANGE (    
-- CHANGEID INTEGER NOT NULL,
-- CONTEXTEID DECIMAL (15 , 0) NOT NULL,
-- CHANGESIZE INTEGER NOT NULL,
-- CHANGEDATA LONG VARCHAR FOR BIT DATA,
-- PRIMARY KEY( CHANGEID, CONTEXTEID, CHANGESIZE ) )
--IN -DB2_NAME-.REPLICA_TABLESPACE-;
-- *******************************************************************************
-- Create the DIR_REPLEERROR table
--
-- NOTE: This SQL statement is commented out. Remove the '--'
-- characters from the following SQL statement to commit the changes
-- made in this section.
-- CREATE TABLE -DB2_USERID-.DIR_REPLERROR ( 
--   ERRORID INTEGER NOT NULL, 
--   AGREEMENTEID DECIMAL(15, 0) NOT NULL, 
--   ERRORSIZE INTEGER NOT NULL, 
--   ERRORDATA LONG VARCHAR FOR BIT DATA, 
--   PRIMARY KEY( ERRORID ) ) 
--IN -DB2_NAME-.REPLICA_TABLESPACE;

-- Create the DIR_LONGREPLERROR table
--
-- NOTE: This SQL statement is commented out. Remove the '--'
-- characters from the following SQL statement to commit the changes
-- made in this section.
--
-- CREATE TABLE -DB2_USERID-.DIR_LONGREPLERROR ( 
--   ERRORID INTEGER NOT NULL, 
--   AGREEMENTEID DECIMAL(15, 0) NOT NULL, 
--   ERRORSEQ INTEGER NOT NULL, 
--   ERRORDATA LONG VARCHAR FOR BIT DATA, 
--   PRIMARY KEY( ERRORID, ERRORSEQ ) ) 
--IN -DB2_NAME-.REPLICA_TABLESPACE;

-- Create the DIR_REPLSTATUS Index
--
-- NOTE: This SQL statement is commented out. Remove the '--'
-- characters from the following SQL statement to commit the changes
-- made in this section.
--
-- CREATE UNIQUE INDEX -DB2_USERID-.DIR_REPLSTATUSX1 
-- ON -DB2_USERID-.DIR_REPLSTATUS ( AGREEMENTEID ) 
-- USING STOGROUP -STORAGEGROUP- 
-- CLUSTER 
-- DEFER YES;

-- Create the DIR_REPLCHANGE Indexes
--
-- NOTE: This SQL statement is commented out. Remove the '--'
-- characters from the following SQL statements to commit the changes
-- made in this section.
--
-- CREATE UNIQUE INDEX -DB2_USERID-.DIR_REPLCHANGEX1 
-- ON -DB2_USERID-.DIR_REPLCHANGE( CHANGEID, CONTEXTID ) 
-- USING STOGROUP -STORAGEGROUP- 
-- CLUSTER 
-- DEFER YES;

--CREATE INDEX -DB2_USERID-.DIR_REPLCHANGEX2 
-- ON -DB2_USERID-.DIR_REPLCHANGE( CONTEXTID ) 
-- USING STOGROUP -STORAGEGROUP- 
-- DEFER YES;

--CREATE INDEX -DB2_USERID-.DIR_REPLCHANGEX3 
-- ON -DB2_USERID-.DIR_REPLCHANGE( CHANGEID ) 
-- USING STOGROUP -STORAGEGROUP- 
-- DEFER YES;

-- Create the DIR_LONGREPLCHANGE Index
--
-- NOTE: This SQL statement is commented out. Remove the '--'
-- characters from the following SQL statement to commit the changes
-- made in this section.
--
-- ******************************************************************
--CREATE UNIQUE INDEX -DB2_USERID-.DIR_LONGREPLCHANGEX1
-- ON -DB2_USERID-.DIR_LONGREPLCHANGE( CHANGEID, CONTEXTID,
-- CHANGESEQ )
-- USING STOGROUP -STORAGEGROUP-
-- CLUSTER
-- DEFER YES;

-- ******************************************************************
-- Create the DIR_REPLERROR Indexes
-- NOTE: This SQL statement is commented out. Remove the '--'
-- characters from the following SQL statements to commit the changes
-- made in this section.
-- ******************************************************************
--CREATE UNIQUE INDEX -DB2_USERID-.DIR_REPLERRORX1
-- ON -DB2_USERID-.DIR_REPLERROR( ERRORID )
-- USING STOGROUP -STORAGEGROUP-
-- CLUSTER
-- DEFER YES;

--CREATE INDEX -DB2_USERID-.DIR_REPLERRORX2
-- ON -DB2_USERID-.DIR_REPLERROR( AGREEMENTEID )
-- USING STOGROUP -STORAGEGROUP-
-- DEFER YES;

-- ******************************************************************
-- Create the DIR_LONGREPLERROR Indexes
-- NOTE: This SQL statement is commented out. Remove the '--'
-- characters from the following SQL statements to commit the changes
-- made in this section.
-- ******************************************************************
--CREATE UNIQUE INDEX -DB2_USERID-.DIR_LONGREPLERRORX1
-- ON -DB2_USERID-.DIR_LONGREPLERROR( ERRORID, ERRORSEQ )
-- USING STOGROUP -STORAGEGROUP-
-- CLUSTER
-- DEFER YES;

--CREATE INDEX -DB2_USERID-.DIR_LONGREPLERRORX2
-- ON -DB2_USERID-.DIR_LONGREPLERROR( AGREEMENTEID )
-- USING STOGROUP -STORAGEGROUP-
-- DEFER YES;

-- ******************************************************************
-- Commit the changes made in this section
-- NOTE: This SQL statement is commented out. Remove the '--'
-- characters from the following SQL statement to commit the changes
-- made in this section.
-- ******************************************************************
-- COMMIT;
Appendix C. Supported server controls

The sections that follow describe the supported server controls. For information about ASN.1 (Abstract Syntax Notation One) and BER (Basic Encoding Rules), go to the following Web site:

authenticateOnly

- **Name:** authenticateOnly
- **Description:** Used on an LDAP bind operation to indicate to the LDAP server that it should not attempt to find any group membership information for the client's bind DN.
- **Assigned object identifier:** 1.3.18.0.2.10.2
- **Target of control:** Server
- **Control criticality:** Critical at client's option
- **Values:** There is no value; the `controlValue` field is absent.
- **Detailed description:** This control is valid when sent on an LDAP client's bind request to the LDAP server. The presence of this control on the bind request overrides alternate DN look-ups, extended group searching, and default group membership gathering, and causes the LDAP server to only authenticate the client's bind DN and not gather group information at all. This control is intended for a client who does not care about group memberships and subsequent complete authorization checking using groups, but is using the bind only for authentication to the LDAP server and fast bind processing.

Do Not Replicate

- **Name:** Do Not Replicate
- **Description:** Used by a client to indicate that an add, delete, modify, or modify DN request is not to be replicated to a consumer or forwarding server in an advanced replication environment.
- **Assigned object identifier:** 1.3.18.0.2.10.23
- **Target of control:** Server
- **Control criticality:** Never
- **Values:** There is no value; the `controlValue` field is absent.
- **Detailed description:** This control is valid when sent on a client's add, delete, modify, or modify DN request. The presence of this control indicates that the supplier server in an advanced replication environment should not replicate the update to a consumer or forwarding server.

Note: The `ldapadd`, `ldapmodify`, `ldapmodrdn`, and `ldapdelete` utilities have a `-L` option to add this control to LDAP server requests. See IBM Tivoli Directory Server Client Programming for z/OS for more information about the `ldapadd`, `ldapmodify`, `ldapmodrdn`, and `ldapdelete` utilities.

IBMLdapProxyControl

- **Name:** IBMLdapProxyControl
- **Description:** Used to provide bind and connection information about extended operation requests that result in LDAP requests to another LDAP server. It is required on `GetDnForUserId` and `GetPrivileges` extended operation requests.
- **Assigned object identifier:** 1.3.18.0.2.10.6
- **Target of control:** EXOP backend of server
- **Control criticality:** Critical
- **Values:** The following ASN.1 (Abstract Syntax Notation One) syntax describes the BER (Basic Encoding Rules) encoding of the control value.
Detailed description: This control provides information about extended operation requests that result in the use of the LDAP client to make LDAP requests to another LDAP server. The EXOP backend uses the connection information and the bind information specified in the control to establish an LDAP connection. Then, using the established connection, it issues additional LDAP directory requests to the server.

If the ConnectInformation is not specified, the EXOP backend attempts to open a connection to its local host using a default port of 389. Otherwise, it uses the LDAP URL specified to open a connection. The specified URL must have the following form:

ldap[s]://host[:port]

where:
- host is a DNS-style host name
- port is an optional port number
- ldaps causes the EXOP backend to open a secure LDAP directory connection. The LDAP server must be set up to use SSL and it cannot use the sslKeyRingPWStashFile option. See Setting up for SSL/TLS for more information about SSL configuration.

If the BindInfo is not specified, the EXOP backend makes all of its LDAP requests anonymously. Otherwise, it uses the Bind DN and the AuthenticationChoice to bind to the LDAP server specified in the ConnectInformation. The EXOP backend does not support the SASL authentication choice which is described in the ASN.1.

If the control is specified more than once in a request, the server returns LDAP_PROTOCOL_ERROR.

IBMMODifyDNRealignDNAttributesControl

- Name: IBMMODifyDNRealignDNAttributesControl
- Description: Used by a client to request that a Modify DN operation be extended to realign attribute values for attributes with Distinguished Name syntax, and other specified attribute types known to contain distinguished names, with the new DN values established by the Modify DN operation for those DNs.
- Assigned object identifier: 1.3.18.0.2.10.11
- Target of control: Server
- Control criticality: Critical at client's option
- Values: There is no value; the controlValue field is absent.
• **Detailed description:** This control is valid when sent on a client's Modify DN request. Distinguished names which are renamed might be embedded in DN-syntax attributes throughout the directory contents. You might want to replace the embedded values with their renamed counterparts (realignment). The presence of this control on the Modify DN request causes the server to realign matching attribute values in all attribute types whose syntax is `Distinguished Name` (OID 1.3.6.1.4.1.1466.115.121.1.12), and in the attribute types of `aclEntry` and `entryOwner`, which are known to contain distinguished names. The server will evaluate whether the bound user has permission to modify the candidate attribute values, as determined by the appropriate access controls and the permissions granted by those access controls to the bound DN. If the permissions granted to the bound DN are sufficient to modify the candidate attribute values, those values will be realigned to match their respective new DN values. If any single access check fails, the entire Modify DN operation fails, and all changes to the directory associated with the current Modify DN operation are undone. The scope for realignment is the backend containing the base DN for the Modify DN request. DN references in other backends or other LDAP servers will not be updated.

### IBMModifyDNTimelimitControl

- **Name:** IBMModifyDNTimelimitControl
- **Description:** Used by a client to request that a Modify DN operation be abandoned if the specified time limit for that operation has been exceeded.
- **Assigned object identifier:** 1.3.18.0.2.10.10
- **Target of control:** Server
- **Control criticality:** Critical at client's option
- **Values:** The following ANSI.1 (Abstract Syntax Notation One) syntax describes the BER (Basic Encoding Rules) encoding of the control value.
  
  ```
  ControlValue ::= SEQUENCE {
      Time Limit INTEGER
  }
  ```

- **Detailed description:** This control is valid when sent on a client's Modify DN request. Modify DN operations might be long-running operations if they affect many entries in the directory (for example, if they rename an entry with a subtree containing many subordinate entries), therefore, you might want to limit the duration of the operation. The presence of this control on the Modify DN request causes the operation to be abandoned by the server if the number of seconds specified in the control value is exceeded. When the operation is abandoned, all changes to the directory associated with the Modify DN operation are undone. A time limit of zero will cause the control to be ignored. The last time limit value will be used if this control is specified more than once.

### IBMSchemaReplaceByValueControl

- **Name:** IBMSchemaReplaceByValueControl
- **Description:** Used on a schema modify request to tell the LDAP server that a replace operation will either replace all schema values or just matching values.
- **Assigned object identifier:** 1.3.18.0.2.10.20
- **Target of control:** Server
- **Control criticality:** Critical at client's option
- **Values:** The following ASN.1 (Abstract Syntax Notation One) statements describe the BER (Basic Encoding Rules) for encoding the control value using implicit tagging:

  ```
  ControlValue ::= SEQUENCE {
      replaceByValue BOOLEAN
  }
  ```

- **Detailed description:** This control is valid when sent on a client's modify request and has meaning only when performing a modify replace operation of an attribute in the LDAP server schema. If the control value is set to TRUE, then each replace value in the modify operation either replaces the existing value
IBMschemaReplaceByValueControl overrides the schemaReplaceByValue server configuration option for the current modify request. The last value will be used if this control is specified more than once.

**manageDsaIT**
- **Name:** manageDsaIT
- **Description:** Used on a request to suppress referral processing, thereby allowing the client to manipulate referral objects.
- **Assigned object identifier:** 2.16.840.1.113730.3.4.2
- **Target of control:** Server
- **Control criticality:** Critical
- **Values:** There is no value; the controlValue field is absent.
- **Detailed description:** This control is valid when sent on a client's search, compare, add, delete, modify, or modify DN request. The presence of the control indicates that the server should not return referrals or search continuation references to the client. This allows the client to read or modify referral objects. The LDAP server will not return a referral even if the requested object is not included in any suffix within the LDAP server and a global referral is defined using the referral option in the LDAP server configuration file.

**No Replication Conflict Resolution**
- **Name:** No Replication Conflict Resolution
- **Description:** Used on an update request from a supplier server to a consumer server in an advanced replication environment to indicate that the consumer server should not resolve any replication conflicts that might occur.
- **Assigned object identifier:** 1.3.18.0.2.10.27
- **Target of control:** Server
- **Control criticality:** Never critical
- **Values:** There is no value; the controlValue field is absent.
- **Detailed description:** This control is valid when sent by a supplier server to a consumer server on an add, delete, modify, or modify DN request in an advanced replication environment. The presence of the control indicates that the consumer server does not attempt to resolve any replication conflicts that might occur, such as rejecting an add request because it has an older createtimestamp value. In this scenario, the consumer server always accepts the replicated updates and attempts to apply them to the targeted backend.

**PersistentSearch**
- **Name:** PersistentSearch
- **Description:** Used on a search request to request not only the current contents of the directory that match the search request but also any entries that match the search specification in the future.
- **Assigned object identifier:** 2.16.840.1.113730.3.4.3
- **Target of control:** Server
- **Control criticality:** Critical at client's option
• **Values:** The following ASN.1 (Abstract Syntax Notation One) syntax describes the BER (Basic Encoding Rules) encoding of the control value.

```
ControlValue ::= SEQUENCE {
  changeTypes INTEGER,
  changesOnly BOOLEAN,
  returnECs BOOLEAN
}
```

```
EntryChangeNotification ::= SEQUENCE {
  changeType ENUMERATED {
    add (1),
    delete (2),
    modify (4),
    moddn (8) },
  previousDN LDAPDN OPTIONAL,
  changeNumber INTEGER OPTIONAL
}
```

Where,

- `changeTypes ::= A bit field that specifies one or more types of changes the client is interested in: 0x01 for add changes, 0x02 for delete changes, 0x04 for modify changes, and 0x08 for modify DN changes.`

- `changesOnly ::= A flag that, if TRUE, only changed entries that match the search are returned. If set to FALSE, existing entries matching the search are returned, in addition to changed entries that match the search.`

- `returnECs ::= A flag that, if TRUE, an entryChangeNotification control is included when returning a changed entry that matches the search. If set to FALSE, the control is not included.`

- `changeType ::= Indicates the type of change made to the entry.`

- `previousDN ::= For a moddn changeType, the DN of the entry before it was renamed.`

- `changeNumber ::= The changeNumber of the change log entry, if any, that was created for this change.`

• **Detailed description:** The control is valid when sent on a client’s search request. Support is provided in the z/OS client to create this control and parse the resultant entries. See `ldap_create_persistentsearch_control()` and `ldap_parse_entrychange_control()` API functions in the [IBM Tivoli Directory Server Client Programming for z/OS](http://www-01.ibm.com/support/docview.wss?uid=swg21283273) for more information.

A persistent search consists of two phases. The first phase is optional (it is done if changesOnly is FALSE), and consists of searching the directory for entries matching the search specification. The second phase consists of executing the search specification against any modifications that occur in the directory and, if found matching, then sending the search results to the waiting client.

Persistent search is supported in the TDBM, CDBM, LDBM, and GDBM backends. In addition, the schema entry (cn=schema) and the rootDSE (zero-length DN) support persistent searches. The `persistentSearch` configuration option can be used in the backend section of the configuration file to enable or disable persistent search for that backend. See Chapter 8, “Customizing the LDAP server configuration” for more information about the `persistentSearch` configuration option.

• **Server behavior:** The server behaves as described in the specification found at [http://www.mozilla.org/directory/ietf-docs/draft-smith-psearch-ldap-01.txt](http://www.mozilla.org/directory/ietf-docs/draft-smith-psearch-ldap-01.txt) with the following exceptions:

  1. An error is returned if an error occurs during processing of the persistent search request. Section 4.b of the specification indicates that SearchResultsDone message is not returned if a persistent search is requested. This is not recognized in the case of an error.

  2. If more than one `PersistentSearchControl` is received per search request, `LDAP_PROTOCOL_ERROR` is returned.

  3. If the requesting client is not bound as adminDN, `LDAP_UNWILLING_TO_PERFORM` is returned.

  4. If persistent search is requested and the dereference option was set to something other than `LDAP_DEREF_NEVER` or `LDAP_DEREF_FINDING`, `LDAP_PROTOCOL_ERROR` is returned. If `LDAP_DEREF_FINDING` is specified, alias dereferencing is performed when the persistent search
is issued to determine the real base entry. The dereferenced base entry is then used to determine if modified entries are within the scope of the persistent search request.

5. If a persistent search request is specified for a suffix that does not exist in the LDAP server configuration file, LDAP_NO_SUCH_OBJECT is returned.

6. If a persistent search request is specified for a suffix that is configured but for a search base that does not exist, no search results are returned until the object is added.

7. The search filter and scope are matched before a delete is done, all other operations are matched afterward. No search results are returned for entries moved out of the search filter or scope because of modification or rename.

8. For a persistent search of the root DSE, the search scope must be LDAP_SCOPE_SUBTREE. Backends that do not support persistent search or do not have persistent search enabled will be skipped if a null-based subtree search is used and the persistent search control is marked as critical, otherwise a typical search will be performed for those backends.

9. If a PersistentSearch control is included in a search request for a TDBM, CDBM, LDBM, or GDBM backend that has not enabled persistent search, the search request is rejected with LDAP_UNAVAILABLE_CRITICAL Extention (0x35) if the control is critical. If the control is not critical, a 'typical' search is performed (even if changesOnly is TRUE).

10. Change log entries trimmed by the LDAP server because of the changeLogMaxAge or changeLogMaxEntries configuration options are not returned to a persistent search of the change log directory.

11. If the manageDsaIT control is not specified with the PersistentSearch control and phase one of the search finds a referral, the referral is returned to the client. If the base of the search is equal to or below a referral, the referral is returned and the persistent search second phase does not occur. During the second phase of persistent search, referral entries are always processed such as typical entries, even if the manageDsaIT control is not specified on the persistent search.

12. Idle connection time out also affects persistent search connections. See the description of the idleConnectionTimeout configuration option in Chapter 8, "Customizing the LDAP server configuration" for more information.

13. sizeLimit and timeLimit parameters and configuration options are respected only during the first phase of persistent search, when existing entries are searched. An error is returned if either limit is exceeded and the persistent search ends. During the second phase, when changed entries are searched, sizeLimit and timeLimit are ignored.

14. Only the entry specified in a modify DN request (the target of the rename operation) can be returned during the second phase of the persistent search. Subentries or entries modified as part of the realignment process are not returned.

15. In a sysplex environment, only one persistent search request needs to be made to one of the LDAP servers in the sysplex. The sysplex support results in all the LDAP servers participating in the persistent search. The exception to this is if the target of the search is a TDBM backend where the server compatibility level is less than 4. In this case, an identical persistent search request must be issued to each LDAP server in the sysplex. See the description of the serverCompatLevel configuration option for more information about the server compatibility level.

16. Persistent search against a TDBM backend in a sysplex environment with large groups can result in performance problems. See large access groups considerations for more information.

17. Operational attributes are returned on persistent searches except the following: aclSource, hasSubordinates, ibm-allGroups, ibm-allMembers, ibm-entryChecksum, ibm-entryChecksumOp, ibm-replicationChangeDif, ibm-replicationFailedChangeCount, ibm-replicationFailedChanges, ibm-replicationIsQuiesced, ibm-
replicationLastActivationTime, ibm-replicationLastChangeId, ibm-replicationLastFinishTime,
ibm-replicationLastResult, ibm-replicationLastResultAdditional, ibm-replicationNextTime,
ibm-replicationPendingChangeCount, ibm-replicationPendingChanges, ibm-
replicationPerformance, ibm-replicationState, ibm-replicationThisServerIsMaster, and
ownerSource. The aclEntry, aclPropagate, entryOwner, and ownerPropagate attributes are
returned only if they are defined for the entry and are not inherited from a superior entry.

### Refresh Entry

- **Name:** Refresh Entry
- **Description:** Used by a consumer server in an advanced replication environment to notify a supplier
  server that a replication conflict has occurred during a modify request.
- **Assigned object identifier:** 1.3.18.0.2.10.24
- **Target of control:** Server
- **Control criticality:** Never critical
- **Values:** There is no value; the controlValue field is absent.
- **Detailed description:** This control is valid when sent on an LDAP modify response from a consumer to
  a supplier in an advanced replication environment when a replication conflict is detected in an entry on
  a consumer server. When the supplier server receives this control along with an LDAP_OTHER return
code from the consumer server, the supplier sends its copy of the modified entry to the consumer with
the intention of bringing the consumer server back in sync.

### ReplicateOperationalAttributes

- **Name:** replicateOperationalAttributes
- **Description:** Used to pass the values of operational attributes that are typically set by the server during
  an add, modify, or modify DN operation.
- **Assigned object identifier:** 1.3.18.0.2.10.19
- **Target of control:** Server
- **Control criticality:** Critical at client’s option
- **Values:** The values in this control identify the operational attributes and values to be set. The following
  ASN.1 (Abstract Syntax Notation One) syntax describes the BER (Basic Encoding Rules) encoding of
  the control value.

```plaintext
ControlValue ::= SEQUENCE OF SEQUENCE {
    operation ENUMERATED {
        add (0),
        delete (1),
        replace (2)
    },
    modification AttributeTypeAndValues
}

AttributeTypeAndValues ::= SEQUENCE {
    type OCTET STRING,
    vals SET OF OCTET STRING
}
```

where:
- operation ::= Indicates whether the operational attribute value should be added to the entry, should
  be deleted from the entry, or should replace the current value in the entry.
- type ::= Specifies the name of the operational attribute.
- vals ::= Specifies the values of the operational attribute.
- Detailed description: This control is intended to be used to pass values to the server for operational
  attributes that are typically set by the server, not by the client. For example, a master server might use
this control to pass the **modifiersName** and **modifyTimestamp** values on a replication request because the entry on the replica will have the same values as on the master.

**Server behavior:**

1. The control is only supported on an add, modify, or modify DN request on a basic replication peer or read-only replica server or an advanced replication consumer server. If the control is specified on another request and the control is critical, the server returns **LDAP_UNAVAILABLE_CRITICAL_EXTENSION**.

2. If using basic replication, the requester must be bound as the master server DN or peer server DN for the backend processing the request, as specified by the **masterServerDN** or **peerServerDN** configuration option in the backend section of the LDAP server configuration file. If using advanced replication, the requester must be bound as the DN specified as the **ibm-replicaCredentialsDN** attribute value in the replication agreement. If the requester is not bound in any of these manners and the control is critical, the server returns **LDAP_UNAVAILABLE_CRITICAL_EXTENSION**.

3. For an add request and for a modify DN request of a TDBM entry, the operation specified in the control value cannot be delete. If delete is specified, the server returns **LDAP_UNWILLING_TO_PERFORM**. delete is supported for a modify request and for a modify DN request of an LDBM or CDBM entry.

4. Each attribute type specified in the control must be defined in the LDAP server schema. If it is not, the server returns **LDAP_UNDEFINED_TYPE** if the control is critical, otherwise it ignores the attribute.

5. There is no ACL checking performed for the changes to the entry resulting from the control. The server does perform schema checking to assure the attributes are allowed in the entry.

6. If more than one **replicateOperationalAttributes** control is specified in a request, the server returns **LDAP_PROTOCOL_ERROR**.

**Replication Supplier ID Bind**

- **Name:** Replication Supplier ID Bind
- **Description:** Used by supplier gateway server when it binds to a consumer server in an advanced replication environment.
- **Assigned object identifier:** 1.3.18.0.2.10.18
- **Target of control:** Server
- **Control criticality:** Never critical
- **Values:** The following ANSI.1 (Abstract Syntax Notation One) syntax describes the BER (Basic Encoding Rules) encoding of the control value.

```plaintext
ControlValue ::= SEQUENCE {
    SupplierServerId OCTET STRING
}
```

where,

- **SupplierServerId** ::= A string containing the advanced replication supplier server id.

- **Detailed description:** This control is used by a gateway server to determine which servers to replicate to. Gateway servers only replicate updates that are received from other gateway servers to their own local servers (servers that reside in the same site as the gateway server, including peer and forwarding servers). When a gateway server binds to its consumer servers, this control is sent with its own server ID as the control value. When a gateway server receives such a control in a bind request, it knows that a gateway server is bound as a supplier and that only local servers should receive replicated updates.

- **Server behavior:** This control is only sent by a gateway server in an advanced replication environment when bound as the master server distinguished name specified in the replication agreement entry. If this control is sent by a user who does not have access, an **LDAP_UNWILLING_TO_PERFORM** error is returned.
**Server Administration**

- **Name:** Server Administration
- **Description:** Used by the LDAP administrator on an add, delete, modify, or modify DN operation under conditions where the operation is typically refused. This control forces the operation to occur.
- **Assigned object identifier:** 1.3.18.0.2.10.15
- **Target of control:** Server
- **Control criticality:** Critical at client's option
- **Values:** There is no value; the controlValue field is absent.
- **Detailed description:** In an advanced replication environment, this control allows an add, delete, modify, or modify DN operation sent by the LDAP administrator to be processed by a server which would typically refuse the operation, such as a quiesced forwarding server or a read-only replica server. The processed operation is then replicated like any other update. This control is used by the ldapdiff utility to enable updates to occur on consumer servers that are no longer synchronized with the supplier server.

**Note:** This control must be used with discretion because entry updates are allowed under unusual circumstances. Therefore, it is the user's responsibility to ensure the server being updated ends up in a state consistent with the other servers in an advanced replication environment. For example, in an advanced replication environment, the entry's modifyTimestamp attribute value, which is used as the base for conflict resolution, might be different on different servers if the entry gets updated individually on those servers with this control.

- **Server behavior:** This control can only be specified by a user bound as the LDAP administrator. If user is not bound as the LDAP administrator, the server returns an LDAP_UNWILLING_TO_PERFORM error. If the server is a supplier or consumer server and is quiesced in an advanced replication environment, the control must be specified in order to allow the update to occur.

**Note:** The ldapadd, ldapmodify, ldapmodrdn, and ldapdelete utilities have a -k option to add this control to LDAP server requests. See IBM Tivoli Directory Server Client Programming for z/OS for more information about the ldapadd, ldapmodify, ldapmodrdn, and ldapdelete utilities.
Appendix D. Supported extended operations

The sections that follow describe the supported extended operations. For information about ASN.1 (Abstract Syntax Notation One) and BER (Basic Encoding Rules), go to the following Web site:

Cascading control replication

- **Name:** Cascading control replication
- **Description:** Performs the requested action to the specified server and passes it along to all replicas of the given replication context. If any of these are forwarding replicas or gateway servers, they pass the extended operation along to their replicas. The operation cascades over the entire advanced replication topology. This extended operation should be targeted against a master server.
- **Assigned Object Identifier:** 1.3.18.0.2.12.15
- **Values:** The following ASN.1 syntax describes the BER encoding of the request value.

```
RequestValue ::= SEQUENCE {
  action INTEGER {
    quiesce (0),
    unquiesce (1),
    replicateNow (2),
    wait (3) },
  contextDN LDAPDN,
  timeout INTEGER
}
```

where,
- **action :=** An integer value indicating the operation to be performed on the specified server.
  - If set to quiesce, updates under the replication context contextDN are restricted to the LDAP administrator, if using the Server Administration control (OID 1.3.18.0.2.10.15), and any replication master DNs with authority under this context. Advanced replication continues for a quiesced context. If the server is restarted, all replication contexts are then unquiesced.
  - If set to unquiesce, updates under the replication context contextDN are allowed and normal operation resumes.
  - If set to replicateNow, all queued updates for each replication agreement under contextDN are immediately replicated, regardless of schedule. After queued replication updates have been replicated, each replication agreement follows its normal schedule. If there are any suspended replication agreements, they are skipped and any queued updates remain queued for those replication agreements. Unlike wait, this extended operation is propagated to the consumer server of each replication agreement without waiting for all queued updates to be applied.
  - If set to wait, all queued updates for each replication agreement under contextDN are immediately replicated, regardless of schedule. After queued replication updates have been replicated, each replication agreement follows its normal schedule. If there are any suspended replication agreements, they are skipped and any queued updates remain queued for those replication agreements. Unlike replicateNow, this extended operation is not propagated to the consumer server of each replication agreement until that agreement is finished replicating.
- **contextDN :=** A distinguished name (DN) containing the replication context that this operation affects.
- **timeout :=** An integer value indicating the number of seconds that the extended operation has to successfully complete. If not present, or 0, the operation has an indefinite amount of time to complete.
- **Detailed description:** The Cascading control replication extended operation returns when one of the following conditions occurs:
  - The request is complete on all servers.
  - A failure has occurred on one of the servers in the replication topology.
The timeout value is exceeded.

The **Cascading control replication** extended operation is allowed only when the bound user has update authority to all replication agreements in the specified contextDN or is authenticated as a master server for the specified contextDN.

- **Response object identifier**: 1.3.18.0.2.12.15
- **Response description**: This extended operation response is used to return error information when a problem is encountered with performing the **Cascading control replication** extended operation.
- **Response values**: The following describes the response value.

  ```plaintext
  ResponseValue ::= SEQUENCE {
    resultCode INTEGER,
    msg OCTET STRING,
    supplier OCTET STRING,
    consumer OCTET STRING,
    additionalResultCode[1] INTEGER OPTIONAL {
      LDAP_REPLICATION_SUCCESS (0),
      LDAP_REPLICATION_RETRYING (2) },
    agreementDN[2] LDAPDN OPTIONAL
  }
  ```

  where,
  - `resultCode` ::= An integer value indicating whether the extended operation is successful. Standard LDAP return codes are returned as values within this portion of the extended operation response.
  - `msg` ::= A string containing a reason code message. In most cases, it is an error message indicating why the extended operation failed. In some cases, it can be an informational or warning message.
  - `supplier` ::= A string containing the shortened hostname and advanced replication server ID of the supplier server targeted by the extended operation. The format is `shortenedHostName:serverID`. If the shortened hostname cannot be determined, the format is `server ID:serverID`.
  - `consumer` ::= A string containing the host:port of the consumer server that is reporting an error. This is only returned when the consumer has a problem performing the requested operation.
  - `additionalResultCode` ::= An integer value that is returned when the `resultCode` is set to `LDAP_TIMEOUT`.
  - `agreementDN` ::= A distinguished name (DN) containing the replication agreement that is in error.

- **Response detailed description**:

  The following table summarizes some different error scenarios and the **Cascading control replication response** returned for such scenarios.

<table>
<thead>
<tr>
<th>Error scenario</th>
<th>Cascading control replication response</th>
</tr>
</thead>
<tbody>
<tr>
<td>contextDN does not exist or is not a replication context</td>
<td>Returns an <strong>LDAP_NO_SUCH_OBJECT</strong> return code</td>
</tr>
<tr>
<td>Backend does not support advanced replication</td>
<td>Returns an <strong>LDAP_UNWILLING_TO_PERFORM</strong> return code</td>
</tr>
<tr>
<td>Not authorized to perform operation</td>
<td>Returns an <strong>LDAP_INSUFFICIENT_ACCESS</strong> return code</td>
</tr>
<tr>
<td>Operation did not complete within the specified time</td>
<td>Returns an <strong>LDAP_TIMEOUT</strong> return code</td>
</tr>
<tr>
<td>Syntax of DN specified is not correct</td>
<td>Returns an <strong>LDAP_INVALID_DN_SYNTAX</strong> return code</td>
</tr>
<tr>
<td>Value for the input option is not valid</td>
<td>Returns an <strong>LDAP_PROTOCOL_ERROR</strong> return code</td>
</tr>
<tr>
<td>LDAP server is unable to decode the request</td>
<td>Returns an <strong>LDAP_DECODING_ERROR</strong> return code</td>
</tr>
</tbody>
</table>

  When a **Cascading control replication response** returns an **LDAP_TIMEOUT** return code, the `additionalResultCode` field in the **Cascading control replication response** is set to either 0 or
LDAP_REPLICATION_RETRYING to indicate the replication update is being retried. The LDAP_REPLICATION_RETRYING error is only returned when action is set to wait.

**changeLogAddEntry**

- **Name:** changeLogAddEntryRequest
- **Description:** Causes the LDAP server to create a change log entry in the change log using information passed to the extended operation. All input values must be in UTF8.
- **Assigned Object Identifier:** 1.3.18.0.2.12.48
- **Values:** The following ASN.1 syntax describes the BER encoding of the request value.

```
RequestValue ::= SEQUENCE {
    version        INTEGER,
    applicationID  INTEGER,
    userid         OCTET STRING,
    group          OCTET STRING,
    class          OCTET STRING,
    resource       OCTET STRING,
    changeType     INTEGER {
        add (0),
        delete (1),
        modify (2),
        rename (3) },
    changeTime     OCTET STRING,
    initiator      OCTET STRING,
    changes        SEQUENCE OF changeAttributeList OPTIONAL
}
```

Where,

- **version:** Identifies which version of the interface is being used. Currently the only value supported is 2. If the interface is extended in the future then other values will be supported.
- **applicationID:** 1 for RACF. Other applications will have different identifiers. The identifier informs the LDAP server which (if any) translations of the data should be done.
- **userid:** A string containing the userid that is created, modified, deleted, or renamed. This string is used to form the value of the targetDN attribute in the change log entry.
- **group:** For the RACF application, a string containing the group that is created, modified, deleted, or renamed. The RACF application can specify a value for both userid and group to indicate that the change is to the connection of that user to that group. This string is used to form the value of the targetDN attribute in the change log entry.
- **class:** A string containing the class of the resource profile that is created, modified, deleted, or renamed. This string is used along with the resource string to form a resource profile DN as the value of the targetDN attribute in the change log entry.
- **resource:** A string containing the resource profile that is created, modified, deleted, or renamed. This string is used along with the class string to form a resource profile DN as the value of the targetDN attribute in the change log entry.
- **changeType:** An integer value indicating the type of change. This is used to form the value of the changeType attribute in the change log entry.
- **changeTime:** A string of decimal numbers, used to form the changeTime attribute in the change log entry. The format of the string is: yyyyymmddhhiissuuuuuuZ.

Where,

- **yyyy** is year, **mm** is month, **dd** is day, **hh** is hour, **ii** is minutes, **ss** is seconds, **uuuuuu** is micro seconds, **Z** is a character constant meaning that this time is based on Zulu time, also known as GMT.
- **initiator:** A string containing the userid that made the change. This string is used to form the value of the ibm-changeInitiatorsName attribute in the change log entry.

```
changeAttributeList ::= SEQUENCE {
    field attributeDescription,
    vals    SEQUENCE OF AttributeValue,
    action  ENUMERATED {
```

Appendix D. Supported extended operations 777
Where,

field ::= is the name of the attribute that has been changed. For RACF, this consists of the segment name followed by a period followed by the field name. LDAP maps the RACF segment and field name to an LDAP attribute name.

vals ::= is a ber representation (length and data) of the new attribute value.

action ::= describes what has happened to the attribute (value add, replace, or delete). To indicate that an entire attribute is deleted, specify an action of delete with no value in the vals field.

requestValue ::= is a flag that, if TRUE, indicates that the attribute value in the vals field is not present and should be requested from the application.

The changeAttributeList values are used to form the changes attribute in the change log entry. If changeAttributeList is not specified, a change log entry is created without a changes attribute. This acts as a notification to the user of the change log that it should read the entire entry out of the directory tree.

Detailed description: Class and resource cannot be specified with userid or group. Both class and resource must be specified if either one is specified. In this case, SDBM must be configured to support RACF resources, by specifying enableResources on in the SDBM section of the LDAP server configuration file.

Response object identifier: 1.3.18.0.2.12.49

Response description: This response is used to return error information if an invalid changeLogAddEntryRequest is passed to the LDAP server. If no errors are encountered, then an indication of success is returned to the caller. All output is in UTF8.

Response values: The following describes the response value.

```
changeLogAddEntryRequest ::= SEQUENCE {
  changeLogResultCode ENUMERATED {
    success (0),
    loggingFailed (1),
    invalidCredentials (2),
    remoteNotSupported (3),
    notConfigured (4),
    notActive (5),
    decodeFailed (6),
    valueOutOfRange (7),
    dnConvertFailed (8)
  },
  msg OCTET STRING
}
```

Response detailed description:

The following table summarizes some different error scenarios and the changeLogAddEntryRequest response returned for such scenarios.

<table>
<thead>
<tr>
<th>Error scenario</th>
<th>changeLogAddEntryRequest's response</th>
</tr>
</thead>
<tbody>
<tr>
<td>An internal error prevents the logging operation from completing</td>
<td>Returns a loggingFailed return code</td>
</tr>
<tr>
<td>The caller is not in supervisor state</td>
<td>Returns an InvalidCredentials return code</td>
</tr>
<tr>
<td>Change log is not configured</td>
<td>Returns a notConfigured return code</td>
</tr>
<tr>
<td>Change log is not active</td>
<td>Returns a notActive return code</td>
</tr>
<tr>
<td>LDAP server is unable to parse the request</td>
<td>Returns a decodeFailed return code</td>
</tr>
<tr>
<td>Value is outside the range of allowable values</td>
<td>Returns a valueOutOfRange return code</td>
</tr>
<tr>
<td>Error scenario</td>
<td>changeLogAddEntryRequest’s response</td>
</tr>
<tr>
<td>-------------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>LDAP server is unable to convert a RACF userid to an LDAP DN</td>
<td>Returns a dnConvertFailed return code</td>
</tr>
</tbody>
</table>

### Control replication

- **Name:** Control replication
- **Description:** Used to suspend replication, resume replication, or force immediate replication by a supplier server in an advanced replication environment. When a replication agreement is suspended, updates under the context are allowed but the agreement queues the updates to its replica server until advanced replication is resumed for the agreement. This extended operation should be targeted against a master server.
- **Assigned Object Identifier:** 1.3.18.0.2.12.16
- **Values:** The following ASN.1 syntax describes the BER encoding of the request value.

```plaintext
RequestValue ::= SEQUENCE {
  action INTEGER {
    suspend (0),
    resume (1),
    replicateNow (2)
  },
  scope INTEGER {
    singleAgreement (0),
    allAgreements (1)
  },
  entryDN LDAPDN
}
```

Where,

- **action:** An integer value indicating the operation to be performed on the supplier server. If set to `suspend`, the replication agreement queues the updates to its replica server until advanced replication is resumed for the agreement. If set to `resume`, advanced replication for the replication agreement continues. If set to `replicateNow` and the replication agreement is waiting for scheduled replication to occur, any outstanding updates are immediately replicated. `replicateNow` has no effect on a suspended replication agreement.
- **scope:** An integer value indicating the extent of the action that is to be performed. If set to `singleAgreement`, the request applies to a single replication agreement. If set to `allAgreements`, the request applies to all replication agreements within a replication context. This parameter indicates whether the `entryDN` is a replication agreement or context entry.
- **entryDN:** A distinguished name (DN) containing the replication context or agreement that this operation affects. If `scope` is set to `singleAgreement`, this specifies the distinguished name of the replication agreement that this extended operation is acting on. If `scope` is set to `allAgreements`, this specifies the distinguished name of the replication context and indicates that all agreements within the context are to be acted on.

- **Response object identifier:** 1.3.18.0.2.12.16
- **Response description:** This extended operation response is used to return error information when a problem is encountered with performing the Control replication extended operation.
- **Response values:** The following ASN.1 syntax describes the BER encoding of the response value.

```plaintext
ResponseValue ::= SEQUENCE {
  resultCode INTEGER,
  msg OCTET STRING,
  consumer OCTET STRING
}
```

Where,

- **resultCode:** An integer value indicating whether the extended operation is successful. Standard LDAP return codes are returned as values within this portion of the extended operation response.
msg ::= A string containing a reason code message. In most cases, it is an error message indicating why
the extended operation failed. In some cases, it can be an informational or warning message.
consumer ::= A string containing the host:port of the consumer server that is reporting an error. This is
returned when the consumer has certain problems performing the requested operation.

- **Response detailed description:**
The following table summarizes some different error scenarios and the Control replication response
returned for such scenarios.

<table>
<thead>
<tr>
<th>Error scenario</th>
<th>Control replication response</th>
</tr>
</thead>
<tbody>
<tr>
<td>entryDN does not exist or is not a replication context or agreement</td>
<td>Returns an LDAP_NO_SUCH_OBJECT return code</td>
</tr>
<tr>
<td>Backend does not support advanced replication</td>
<td>Returns an LDAP_UNWILLING_TO_PERFORM return code</td>
</tr>
<tr>
<td>Not authorized to perform operation</td>
<td>Returns an LDAP_INSUFFICIENT_ACCESS return code</td>
</tr>
<tr>
<td>Syntax of DN specified is not correct</td>
<td>Returns an LDAP_INVALID_DN_SYNTAX return code</td>
</tr>
<tr>
<td>Value for the input option is not valid</td>
<td>Returns an LDAP_PROTOCOL_ERROR return code</td>
</tr>
<tr>
<td>LDAP server is unable to decode the request</td>
<td>Returns an LDAP_DECODING_ERROR return code</td>
</tr>
</tbody>
</table>

### Control replication error log

- **Name:** Control replication error log
- **Description:** Used to display advanced replication errors in the error log and correct any advanced
  replication problems that occur.
- **Assigned Object Identifier:** 1.3.18.0.2.12.56
- **Values:** The following ASN.1 syntax describes the BER encoding of the request value.

```asn1
RequestValue ::= SEQUENCE {
    errorOption INTEGER {
        retry (1),
        display (2),
        delete (3)
    },
    failureId OCTET STRING,
    agreementDN LDAPDN
}
```

Where,

- **errorOption** ::= An integer value indicating the operation to be performed on the advanced replication
  error log. If set to retry, tries to reprocess one or all failed replication updates. If set to delete, deletes
  one or all failed replication updates. If set to show, shows the failed update specified by the failureId.
  The failureId cannot be set to 0 when errorOption is set to show.
- **failureId** ::= A string indicating the target of the operation. If set to 0, then all advanced replication
  errors in the error log are either retried or deleted based on the errorOption setting. Otherwise, this
  value specifies the failure ID in the advanced replication error log that is to be retried, displayed, or
  deleted based on the errorOption setting. The failure ID can be determined by searching the
  agreementDN for the ibm-replicationFailedChanges operational attribute. The failure ID must be in the
  range 1 - 4294967295.
- **agreementDN** ::= A distinguished name (DN) containing the replication agreement that this operation
  affects.

- **Response object identifier:** 1.3.18.0.2.12.57
- **Response description:** This extended operation response is used to return error information when a
  problem is encountered with performing the Control replication error log extended operation.
- **Response values:** The following ASN.1 syntax describes the BER encoding of the response value.
ResponseValue ::= SEQUENCE {
  resultCode INTEGER,
  countEffected OCTET STRING,
  msg OCTET STRING
}

Where,

- resultCode ::= An integer value indicating whether the extended operation is successful. Standard LDAP return codes are returned as values within this portion of the extended operation response.
- countEffected ::= A string containing the number of error log failures retried, deleted, or displayed.
- msg ::= A string containing a reason code message. In most cases, it is an error message indicating why the extended operation failed. In some cases, it can be an informational or warning message.

**Response detailed description:**

The following table summarizes some different error scenarios and the Control replication error log response returned for such scenarios.

<table>
<thead>
<tr>
<th>Error scenario</th>
<th>Control replication error log response</th>
</tr>
</thead>
<tbody>
<tr>
<td>agreementDN does not exist or is not a replication agreement</td>
<td>Returns an LDAP_NO_SUCH_OBJECT return code</td>
</tr>
<tr>
<td>failureId does not exist for any agreement in this backend</td>
<td>Returns an LDAP_NO_SUCH_OBJECT return code</td>
</tr>
<tr>
<td>Backend does not support advanced replication</td>
<td>Returns an LDAP_UNWILLING_TO_PERFORM return code</td>
</tr>
<tr>
<td>failureId is not logged to this agreementDN</td>
<td>Returns an LDAP_UNWILLING_TO_PERFORM return code</td>
</tr>
<tr>
<td>Not authorized to perform operation</td>
<td>Returns an LDAP_INSUFFICIENT_ACCESS return code</td>
</tr>
<tr>
<td>Syntax of DN specified is not correct</td>
<td>Returns an LDAP_INVALID_DN_SYNTAX return code</td>
</tr>
<tr>
<td>Value for the input option is not valid</td>
<td>Returns an LDAP_PROTOCOL_ERROR return code</td>
</tr>
<tr>
<td>LDAP server is unable to decode the request</td>
<td>Returns an LDAP_DECODING_ERROR return code</td>
</tr>
</tbody>
</table>

### Control replication queue

- **Name:** Control replication queue
- **Description:** Used to indicate which pending changes in the advanced replication queue for a replication agreement ought to be skipped (deleted) and not replicated to the consumer server.
- **Assigned Object Identifier:** 1.3.18.0.2.12.17
- **Values:** The following ASN.1 syntax describes the BER encoding of the request value.

```
requestValue ::= SEQUENCE {
  action INTEGER {
    skipAll (0),
    skipSingle (1)
  },
  agreementDN LDAPDN,
  changeId OCTET STRING OPTIONAL
}
```

Where,

- action ::= An integer value indicating the operation that is to be performed on the advanced replication queue. If set to skipAll, the server skips (deletes) all updates that have not yet been replicated from the replication agreement. If set to skipSingle, the server skips (deletes) the specified changeId. Only the next change to be replicated can be skipped in this manner. If the changeId that is specified is not the first one in the list of pending changes, the extended operation fails. This ensures that the operation only affects the entry that is preventing advanced replication from occurring.
agreementDN ::= A distinguished name (DN) containing the replication agreement that this operation affects.
changeID ::= A string that identifies the change ID of a pending operation in the replication agreement that is to be skipped (deleted). The changeId that is specified must be in the range 1 - 4294967295.
Change IDs can be determined by searching the agreementDN for the ibm-replicationPendingChanges operational attribute. The changeId is required when action is set to skipSingle and ignored when action is set to skipAll.

- Response object identifier: 1.3.18.0.2.12.17
- Response description: This extended operation response is used to return error information when a problem is encountered with performing the Control replication queue extended operation. If action is set to skipAll and there are no pending updates in the advanced replication queue, the extended operation is considered successful.
- Response values: The following ASN.1 syntax describes the BER encoding of the response value.

```
ResponseValue ::= SEQUENCE {
    resultCode INTEGER,
    msg OCTET STRING,
    changesSkipped INTEGER
}
```

Where,

resultCode ::= An integer value indicating whether the extended operation is successful. Standard LDAP return codes are returned as values within this portion of the extended operation response.
msg ::= A string containing a reason code message. In most cases, it is an error message indicating why the extended operation failed. In some cases, it can be an informational or warning message.
changesSkipped ::= An integer value indicating the number of pending updates in the advanced replication queue that have been skipped (deleted).

- Response detailed description:

The following table summarizes some different error scenarios and the Control replication queue response returned for such scenarios.

<table>
<thead>
<tr>
<th>Error scenario</th>
<th>Control replication queue response</th>
</tr>
</thead>
<tbody>
<tr>
<td>agreementDN does not exist or is not a replication agreement</td>
<td>Returns an LDAP_NO_SUCH_OBJECT return code</td>
</tr>
<tr>
<td>Backend does not support advanced replication</td>
<td>Returns an LDAP_UNWILLING_TO_PERFORM return code</td>
</tr>
<tr>
<td>The specified changeId is not the next change to be replicated</td>
<td>Returns an LDAP_UNWILLING_TO_PERFORM return code</td>
</tr>
<tr>
<td>Not authorized to perform the requested operation</td>
<td>Returns an LDAP_INSUFFICIENT_ACCESS return code</td>
</tr>
<tr>
<td>Syntax of DN specified is not correct</td>
<td>Returns an LDAP_INVALID_DN_SYNTAX return code</td>
</tr>
<tr>
<td>Value for the input option is not valid</td>
<td>Returns an LDAP_PROTOCOL_ERROR return code</td>
</tr>
<tr>
<td>LDAP server is unable to decode the request</td>
<td>Returns an LDAP_DECODING_ERROR return code</td>
</tr>
</tbody>
</table>

GetDnForUserid

- Name: GetDnForUserid
- Description: Causes the EXOP backend to open a connection to an LDAP server with Policy Director data to retrieve all of a user ID’s distinguished names. The extended operation is rejected if the EXOP backend is not configured.
- Assigned Object Identifier: 1.3.18.0.2.12.8
- Values: The following ASN.1 syntax describes the BER encoding of the request value.
RequestValue ::= SEQUENCE {
  Userid OCTET STRING,
  Searchbase [0] LDAPDN OPTIONAL,
  EntryTypes [1] SEQUENCE OF Objectclass-name OPTIONAL
}

Where,
Objectclass-name ::= OCTET STRING

- **Detailed description:** Given a user ID and the required IBMLdapProxyControl, the EXOP backend opens a connection to the target LDAP server specified in the IBMLdapProxyControl and retrieves all of the specified user ID’s distinguished names.

  The search base in the request value establishes the sub-tree to search for the user ID’s distinguished names. If this is not specified, the EXOP backend performs a root DSE search to determine all of the naming contexts of the target LDAP server and proceed to search each naming context for the user ID’s distinguished names. In addition to the search base, the user can specify optional object classes to filter distinguished names from the result.

- **Response object identifier:** 1.3.18.0.2.12.10
- **Response description:** Returned by EXOP backend when it receives a GetDnForUserid extended operations request.
- **Response values:** The following ASN.1 syntax describes the BER encoding of the response value.

  ResponseValue ::= SEQUENCE OF Distinguished-name

  where,
  Distinguished-name LDAPDN

- **Response detailed description:** When the EXOP backend receives a GetDnForUserid extended operation request and the required IBMLdapProxyControl, it issues requests to the target LDAP server specified in the IBMLdapProxyControl to retrieve all of the specified user ID’s distinguished names. The user may further filter the results by specifying a search base and object class names in the request value.

  The following table summarizes some different error scenarios and the EXOP backend’s response returned for such scenarios.

<table>
<thead>
<tr>
<th>Error scenario</th>
<th>EXOP backend’s response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cannot find distinguished names</td>
<td>Returns an LDAP_NO_SUCH_OBJECT return code</td>
</tr>
<tr>
<td>Encounters an LDAP_NO_SUCH_OBJECT return code in its attempt to bind to the target LDAP server</td>
<td>Returns an LDAP_INAPPROPRIATE_AUTH return code</td>
</tr>
<tr>
<td>Encounters any other unsuccessful return codes in its attempt to make LDAP requests to the target LDAP server</td>
<td>Returns these return codes encountered and a detailed message describing the point of failure</td>
</tr>
</tbody>
</table>

**GetPrivileges**

- **Name:** GetPrivileges
- **Description:** Causes the EXOP backend to open a connection to an LDAP server with Policy Director data and retrieve all of a subject’s Policy Director privilege information. The extended operation is rejected if the EXOP backend is not configured.
- **Assigned Object Identifier:** 1.3.18.0.2.12.7
- **Values:** The following ASN.1 syntax describes the BER encoding of the request value.

  RequestValue ::= SEQUENCE {
    Subject LDAPDN,
    DomainName OCTET STRING OPTIONAL
  }
• **Detailed description:** Given the subject and the required `IBMLdapProxyControl`, the EXOP backend opens a connection to the target LDAP server specified in the `IBMLdapProxyControl` and retrieves all of the specified subject's Policy Director data. If no domain name is specified, the EXOP backend assumes that the subject exists in the DEFAULT domain.

• **Response object identifier:** 1.3.18.0.2.12.9

• **Response description:** Returned by EXOP backend when it receives a GetPrivileges extended operations request.

• **Response values:** The following ASN.1 syntax describes the BER encoding of the response value.

```asn1
ResponseValue ::= SEQUENCE {
  DomainName           OCTET STRING,
  SecLoginType         OCTET STRING,
  PrincipalName        OCTET STRING,
  SecPwValid           BOOLEAN,
  SecAcctValid         BOOLEAN,
  UserUUID             SEQUENCE {
    Username           LDAPDN,
    UserUUID           OCTET STRING
  }
  GroupInfo            SEQUENCE {
    NumberOfGroups     INTEGER,
    GroupUUIDs         SEQUENCE OF SEQUENCE {
      Groupname         LDAPDN,
      GroupUUID         OCTET STRING
    }
  }
}
```

• **Response detailed description:** When the EXOP backend receives a GetPrivileges extended operation request and the required `IBMLdapProxyControl`, it issues requests to the target LDAP server specified in the `IBMLdapProxyControl` to retrieve all of the subject's Policy Director data. See Policy Director Authorization Services for z/OS and OS/390 Customization and Use for descriptions of the fields returned in the response value.

The following table summarizes some different error scenarios and the EXOP backend's response returned for such scenarios.

<table>
<thead>
<tr>
<th>Error scenario</th>
<th>EXOP backend's response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cannot find any of the fields in the response value</td>
<td>Returns an LDAP_NO_SUCH_OBJECT return code</td>
</tr>
<tr>
<td>Retrieves more than one UserUUID or more than one GroupUUID per Groupname</td>
<td>Returns an LDAP_OTHER return code</td>
</tr>
<tr>
<td>Encounters an LDAP_NO_SUCH_OBJECT return code in its attempt to bind to the target LDAP server</td>
<td>Returns an LDAP_INAPPROPRIATE_AUTH return code</td>
</tr>
<tr>
<td>Encounters any other unsuccessful return codes in its attempt to make LDAP requests to the target LDAP server</td>
<td>Returns these return codes encountered and a detailed message describing the point of failure</td>
</tr>
</tbody>
</table>

---

**Quiesce or unquiesce context**

• **Name:** Quiesce or unquiesce context

• **Description:** Used to change an advanced replication context to or from a quiesced state. In a quiesced state, the entire subtree starting from the context does not accept client updates except from the LDAP administrator, if using the Server Administration control (OID 1.3.18.0.2.10.15), and any replication master DNs with authority under this context. Advanced replication continues for a quiesced context. If the server is restarted, all replication contexts are then unquiesced.

• **Assigned Object Identifier:** 1.3.18.0.2.12.19

• **Values:** The following ASN.1 syntax describes the BER encoding of the request value.
RequestValue ::= SEQUENCE {
  quiesce BOOLEAN,
  contextDN LDAPDN
}

Where,
  quiesce ::= A boolean indicating whether to quiesce (TRUE) or unquiesce (FALSE) an advanced replication context.
  contextDN ::= A distinguished name (DN) containing the replication context that this operation affects.

- Response object identifier: 1.3.18.0.2.12.19
- Response description: This extended operation response is used to return error information when a problem is encountered with performing the Quiesce or unquiesce context extended operation.
- Response values: The following ASN.1 syntax describes the BER encoding of the response value.

ResponseValue ::= SEQUENCE {
  resultCode INTEGER,
  msg OCTET STRING
}

Where,
  resultCode ::= An integer value indicating whether the extended operation is successful. Standard LDAP return codes are returned as values within this portion of the extended operation response.
  msg ::= A string containing a reason code message. In most cases, it is an error message indicating why the extended operation failed. In some cases, it can be an informational or warning message.

- Response detailed description:
  The following table summarizes some different error scenarios and the Quiesce or unquiesce context response returned for such scenarios.

<table>
<thead>
<tr>
<th>Error scenario</th>
<th>Quiesce or unquiesce context response</th>
</tr>
</thead>
<tbody>
<tr>
<td>contextDN does not exist or is not a replication context</td>
<td>Returns an LDAP_NO_SUCH_OBJECT return code</td>
</tr>
<tr>
<td>Backend does not support advanced replication</td>
<td>Returns an LDAP_UNWILLING_TO_PERFORM return code</td>
</tr>
<tr>
<td>Not authorized to perform operation</td>
<td>Returns an LDAP_INSUFFICIENT_ACCESS return code</td>
</tr>
<tr>
<td>Syntax of DN specified is not correct</td>
<td>Returns an LDAP_INVALID_DN_SYNTAX return code</td>
</tr>
<tr>
<td>Unable to find the request data</td>
<td>Returns an LDAP_PROTOCOL_ERROR return code</td>
</tr>
<tr>
<td>LDAP server is unable to decode the request</td>
<td>Returns an LDAP_DECODING_ERROR return code</td>
</tr>
</tbody>
</table>

Replication topology

- Name: Replication topology
- Description: Used to synchronize replication topology related entries across the replication topology. A replication topology entry is an entry that contains an objectclass whose name begins with ibm-replica, such as ibm-replicationContext, ibm-replicaGroup, ibm-replicaSubentry, ibm-replicationAgreement, and ibm-replicationCredentialsSimple. This extended operation is cascaded through all forwarding and gateway servers if agreementDN is not specified. If agreementDN is specified, then the extended operation only synchronizes the replication topology entries in the consumer server defined by the replication agreement.

Note: Replication topology entries containing credentials are also replicated if they are located under the replication context. These entries should be located instead under the cn=localhost suffix.
- Assigned Object Identifier: 1.3.18.0.2.12.54
- Values: The following ASN.1 syntax describes the BER encoding of the request value.
RequestValue ::= SEQUENCE {
  contextDN      LDAPDN,
  timeout        INTEGER,
  agreementDN    LDAPDN OPTIONAL
}

Where,
  contextDN ::= A distinguished name (DN) containing the replication context on the supplier server that
               the advanced replication topology related entries are synchronized.
  timeout ::= An integer value indicating the number of seconds that the extended operation has to
               successfully complete. If not present, or 0, the operation has an indefinite amount of time to complete.
  agreementDN ::= A distinguished name (DN) containing the replication agreement used to connect to a
                  consumer server. If a value is specified, only the consumer server defined in the agreementDN is
                  synchronized. The extended operation is not cascaded to any consumers of that server.

• Response object identifier: 1.3.18.0.2.12.55

• Response description: This extended operation response is used to return error information when a
  problem is encountered with performing the Replication topology extended operation.

• Response values: The following ASN.1 syntax describes the BER encoding of the response value.
ResponseValue ::= SEQUENCE {
  resultCode INTEGER,
  msg         OCTET STRING,
  supplier    OCTET STRING,
  consumer    OCTET STRING
}

Where,
  resultCode ::= An integer value indicating whether the extended operation is successful. Standard
                 LDAP return codes are returned as values within this portion of the extended operation response.
  msg ::= A string containing a reason code message. In most cases, it is an error message indicating why
          the extended operation failed. In some cases, it can be an informational or warning message.
  supplier ::= A string containing the shortened hostname and advanced replication server ID of the
               supplier server targeted by the extended operation. The format is ‘shortenedHostName:serverID’. If the
               shortened hostname cannot be determined, the format is ‘serverID:serverID’.
  consumer ::= A string containing the host:port of the consumer server that is reporting an error. This is
               returned when the consumer has certain problems performing the requested operation.

• Response detailed description:
  The following table summarizes some different error scenarios and the Replication topology response
  returned for such scenarios.

<table>
<thead>
<tr>
<th>Error scenario</th>
<th>Replication topology response</th>
</tr>
</thead>
<tbody>
<tr>
<td>contextDN does not exist or is not a</td>
<td>Returns an LDAP_NO_SUCH_OBJECT return code</td>
</tr>
<tr>
<td>replication context</td>
<td></td>
</tr>
<tr>
<td>agreementDN does not exist or is not a</td>
<td>Returns an LDAP_NO_SUCH_OBJECT return code</td>
</tr>
<tr>
<td>replication agreement</td>
<td></td>
</tr>
<tr>
<td>Backend does not support advanced replication</td>
<td>Returns an LDAP_UNWILLING_TO_PERFORM return code</td>
</tr>
<tr>
<td>agreementDN is not under a replication</td>
<td>Returns an LDAP_UNWILLING_TO_PERFORM return code</td>
</tr>
<tr>
<td>context</td>
<td></td>
</tr>
<tr>
<td>Not authorized to perform operation</td>
<td>Returns an LDAP_INSUFFICIENT_ACCESS return code</td>
</tr>
<tr>
<td>Operation did not complete with specified</td>
<td>Returns an LDAP_TIMEOUT return code</td>
</tr>
<tr>
<td>time</td>
<td></td>
</tr>
<tr>
<td>Syntax of DN specified is not correct</td>
<td>Returns an LDAP_INVALID_DN_SYNTAX return code</td>
</tr>
<tr>
<td>Value for the input option is not valid</td>
<td>Returns an LDAP_PROTOCOL_ERROR return code</td>
</tr>
</tbody>
</table>
Error scenario | Replication topology response
--- | ---
LDAP server is unable to decode the request | Returns an LDAP_DECODING_ERROR return code

### Start TLS

- **Name:** Start TLS Extended Request
- **Description:** Causes a non-secure connection to change to a secure connection.
- **Assigned Object Identifier:** 1.3.6.1.4.1.1466.20037
- **Values:** None.
- **Detailed description:** The client may send the Start TLS extended request at any time after establishing an LDAP directory association, except in the following cases:
  - If a secure connection is already established, or
  - During a multi-stage SASL negotiation, or
  - If there are any outstanding LDAP directory operations on the connection.

  The LDAP server responds with an indication of whether the change to a secure connection is allowed. If accepted, the client is expected to immediately begin the secure protocol handshake.

  The secure connection might be ended and a non-secure connection resumed by having the client cause a TLS closure alert to be sent to the server. Communication after receiving the TLS closure alert is over a non-secure connection. The client is considered to be in an anonymous authentication state.

- **Response object identifier:** 1.3.6.1.4.1.1466.20037
- **Response description:** Upon receiving the Start TLS extended request, the server returns an extended response containing a response code indicating success or failure.
- **Response values:** For the successful response, no response value is returned. For an error response, a response message indicating the cause of the error is returned.
- **Response detailed description:**

  The following table summarizes some different error scenarios and the server’s response returned for such scenarios.

<table>
<thead>
<tr>
<th>Error scenario</th>
<th>Server response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server accepts connection and can handle the request</td>
<td>Returns an LDAP_SUCCESS return code</td>
</tr>
<tr>
<td>SSL/TLS is not configured</td>
<td>Returns an LDAP_UNAVAILABLE return code</td>
</tr>
<tr>
<td>A secure connection is already established</td>
<td>Returns an LDAP_OPERATIONS_ERROR return code</td>
</tr>
<tr>
<td>Secure connections are not supported by the server</td>
<td>Returns an LDAP_PROTOCOL_ERROR return code</td>
</tr>
<tr>
<td>There are outstanding operations on the connection</td>
<td>Returns an LDAP_OPERATIONS_ERROR return code</td>
</tr>
<tr>
<td>A multi-stage SASL negotiation is in progress</td>
<td>Returns an LDAP_OPERATIONS_ERROR return code</td>
</tr>
</tbody>
</table>

### unloadRequest

- **Name:** unloadRequest
- **Description:** Causes the LDAP server to unload entries from a directory into a file in LDIF format. This extended operation is used by the ds2ldif utility. See [ds2ldif utility](#) for more information about ds2ldif.
- **Assigned Object Identifier:** 1.3.18.0.2.12.62
- **Values:** The following ASN.1 syntax describes the BER encoding of the request value:

  ```
  RequestValue ::= SEQUENCE {
    tagOption BOOLEAN,
    outputFileName OCTET STRING,
    [0] subtreeDN LDAPDN OPTIONAL,
  }
  ```

Appendix D. Supported extended operations 787
backendName ::= OCTET STRING OPTIONAL,

genealogicalOrder ::= BOOLEAN OPTIONAL,

filterDN ::= LDAPDN OPTIONAL,

noControlValues ::= BOOLEAN OPTIONAL,

unloadLocalhost ::= BOOLEAN OPTIONAL

Where,

\[ \text{tagOption ::=} \text{A boolean indicating whether the encryption tag of userPassword attribute values should be displayed in the clear when directory data is unloaded. This value corresponds to the } -t \text{ option of the ds2ldif utility.} \]

\[ \text{outputFileName ::=} \text{A string containing the fully qualified z/OS UNIX System Services file name or dataset name that the server writes the unloaded directory entries in LDIF format. This value corresponds to the } -o \text{ option of the ds2ldif utility.} \]

\[ \text{subtreeDN ::=} \text{A distinguished name (DN) containing the name of the top entry to unload. This entry and all of the entries below it in the directory hierarchy are unloaded. The value must be the DN of an entry in an LDBM, TDBM, or CDBM backend or of the LDAP server schema entry, cn=schema. This value corresponds to the } -s \text{ option of the ds2ldif utility.} \]

\[ \text{backendName ::=} \text{A string containing the name of an LDBM, TDBM, or CDBM backend. All the entries in the backend are unloaded. The value is either the optional fourth parameter of the database option in the LDAP server configuration file or the name generated for the backend by the LDAP server if the parameter is not specified in the configuration file. This value corresponds to the } -n \text{ option of the ds2ldif utility.} \]

\[ \text{genealogicalOrder ::=} \text{A boolean value that specifies how the entries are unloaded. If set to TRUE, entries in each subtree are unloaded together, doing a depth-first traversal of the directory. If set to FALSE or if not specified, there is no guaranteed order of unloaded entries, other than parent entries are always unloaded before child entries. This value corresponds to the } -g \text{ option of the ds2ldif utility.} \]

\[ \text{filterDN ::=} \text{A distinguished name (DN) of a replication filter entry that contains ibm-replicationFilterAttr attribute values. These values are filters used to skip entire entries or attributes within entries while unloading the directory. See [Partial replication] for more information about replication filter entries. Advanced replication must be enabled in the LDAP server with the CDBM backend configured and useAdvancedReplication on specified in the CDBM backend section of the server configuration file. This value corresponds to the } -q \text{ option of the ds2ldif utility.} \]

\[ \text{noControlValues ::=} \text{A boolean indicating whether the replicateOperationalAttributes control value is unloaded for each entry. If set to TRUE, the replicateOperationalAttributes control is not unloaded for each entry. If set to FALSE or if not specified and the server compatibility level is 5 or greater, the replicateOperationalAttributes control value is written to the output LDIF file. This value corresponds to the } -j \text{ option of the ds2ldif utility.} \]

\[ \text{unloadLocalhost ::=} \text{A boolean indicating whether the cn=localhost subtree is unloaded if it exists on the targeted LDAP server. This boolean can only be set to TRUE when subtreeDN is not specified. This value corresponds to the } -l \text{ option of the ds2ldif utility.} \]

**Detailed description:** The unloadRequest extended operation is rejected if the requester is not bound as the LDAP administrator, as specified by the adminDN option in the LDAP server configuration file. subtreeDN and backendName cannot both be specified in the request. If neither is specified and there is only one LDBM, TDBM, or CDBM backend configured, then that backend is unloaded; otherwise, an error is returned.

The aclEntry, aclPropagate, entryOwner, and ownerPropagate attributes are included in an unloaded entry if these attributes have been explicitly set for that entry. The unloaded entry also includes the ibm-entryuuid attribute. The replicateOperationalAttributes control value is written to the output LDIF file if noControlValues is set to FALSE.
The `ds2ldif` utility uses this extended operation to unload the selected directory data if it encounters a problem starting the backend that is to be unloaded or if the `-r` command-line option is specified. For additional details about the operation of the `ds2ldif` utility, see `ds2ldif utility`.

- **Response object identifier:** `1.3.18.0.2.12.63`
- **Response description:** The response indicates the number of entries that were successfully unloaded by the LDAP server as a result of the `unloadRequest` extended operation.
- **Response values:** The following ASN.1 syntax describes the BER encoding of the response value.
  ```
  ResponseValue ::= SEQUENCE {
    entryCount INTEGER
  }
  ```
- **Response detailed description:**
  The following table summarizes some different error scenarios and the `unloadRequest` response returned for such scenarios:

<table>
<thead>
<tr>
<th>Error scenario</th>
<th><code>unloadRequest</code>'s response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not bound as the LDAP administrator DN</td>
<td>Returns an <code>LDAP_OTHER</code> return code</td>
</tr>
<tr>
<td>Multiple LDBM, TDBM, or CDBM backends are active in the server but a subtreeDN or backendName is not specified</td>
<td>Returns an <code>LDAP_OTHER</code> return code</td>
</tr>
<tr>
<td>A non-existent subtreeDN, filterDN, or backendName is specified</td>
<td>Returns an <code>LDAP_OTHER</code> return code</td>
</tr>
<tr>
<td>Error opening or writing the output LDIF file</td>
<td>Returns an <code>LDAP_OTHER</code> return code</td>
</tr>
<tr>
<td>Both subtreeDN and backendName are specified</td>
<td>Returns an <code>LDAP_PROTOCOL_ERROR</code> return code</td>
</tr>
<tr>
<td>A zero-length value is specified for subtreeDN, filterDN, or backendName</td>
<td>Returns an <code>LDAP_PROTOCOL_ERROR</code> return code</td>
</tr>
<tr>
<td>LDAP server is unable to parse the request</td>
<td>Returns an <code>LDAP_PROTOCOL_ERROR</code> return code</td>
</tr>
<tr>
<td>A subtreeDN is specified when <code>unloadLocalHost</code> is set to TRUE</td>
<td>Returns an <code>LDAP_PROTOCOL_ERROR</code> return code</td>
</tr>
<tr>
<td><code>filterDN</code> specified when CDBM backend is not configured or <code>useAdvancedReplication off</code> is specified in the CDBM backend section of the server configuration file</td>
<td>Returns an <code>LDAP_UNWILLING_TO_PERFORM</code> return code</td>
</tr>
</tbody>
</table>
Appendix E. SMF records

SMF Record Type 83, subtype 3 records

When auditing is enabled for LDAP events, SMF record type 83, subtype 3 records are recorded in the SMF dataset. Each logged LDAP event has a unique event code with a corresponding event code qualifier. The event qualifier indicates if the value that indicates if the event succeeded or failed. For more information about SMF records and the relocates that are common to all SMF Type 83 subtype 2 and above records, see z/OS Security Server RACF Macros and Interfaces.

Table 80 shows LDAP event codes and event code qualifiers:

<table>
<thead>
<tr>
<th>Event</th>
<th>Operation</th>
<th>Event qualifier</th>
<th>Description</th>
<th>Relocate type sections</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Add</td>
<td>0</td>
<td>Successful LDAP operation</td>
<td>Common relocates (100-112) 201-208</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>Failed LDAP operation</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Bind</td>
<td>0</td>
<td>Successful LDAP operation</td>
<td>Common relocates (100-112) 201-207 209</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>Failed LDAP operation</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Compare</td>
<td>0</td>
<td>Successful LDAP operation</td>
<td>Common relocates (100-112) 201-207 210</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>Failed LDAP operation</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Connect</td>
<td>0</td>
<td>Successful LDAP operation</td>
<td>Common relocates (100-112) 201-207</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>Failed LDAP operation</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Delete</td>
<td>0</td>
<td>Successful LDAP operation</td>
<td>Common relocates (100-112) 201-207</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>Failed LDAP operation</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Disconnect</td>
<td>0</td>
<td>Successful LDAP operation</td>
<td>Common relocates (100-112) 201-207</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>Failed LDAP operation</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Extended</td>
<td>0</td>
<td>Successful LDAP operation</td>
<td>Common relocates (100-112) 201-207 221</td>
</tr>
<tr>
<td>Operation</td>
<td></td>
<td>1</td>
<td>Failed LDAP operation</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Modify</td>
<td>0</td>
<td>Successful LDAP operation</td>
<td>Common relocates (100-112) 201-207 211-213</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>Failed LDAP operation</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Modify DN</td>
<td>0</td>
<td>Successful LDAP operation</td>
<td>Common relocates (100-112) 201-207 214-216</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>Failed LDAP operation</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Search</td>
<td>0</td>
<td>Successful LDAP operation</td>
<td>Common relocates (100-112) 201-207 218-220</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>Failed LDAP operation</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Unbind</td>
<td>0</td>
<td>Successful LDAP operation</td>
<td>Common relocates (100-112) 201-207</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>Failed LDAP operation</td>
<td></td>
</tr>
</tbody>
</table>

Table 81 shows LDAP extended relocates:

<table>
<thead>
<tr>
<th>Relocate #</th>
<th>Field name</th>
<th>Type</th>
<th>Length</th>
<th>Audited by event code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>LDAP_RESERVED_01</td>
<td>CHAR</td>
<td>16</td>
<td>N/A</td>
<td>Reserved</td>
</tr>
<tr>
<td>Relocate #</td>
<td>Field name</td>
<td>Type</td>
<td>Length</td>
<td>Audited by event code</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>--------------------</td>
<td>-------</td>
<td>--------</td>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>101</td>
<td>LDAP_REQ_TIMESTP</td>
<td>CHAR</td>
<td>16</td>
<td>1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11</td>
<td>Elapsed time for request completion (in microseconds)</td>
</tr>
<tr>
<td>102</td>
<td>LDAP_SERVER_URL</td>
<td>CHAR</td>
<td>32</td>
<td>1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11</td>
<td>The auditing server’s URL for the connection that the client came in on (Listen URL)</td>
</tr>
<tr>
<td>103</td>
<td>LDAP_CONN_ID</td>
<td>CHAR</td>
<td>8</td>
<td>1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11</td>
<td>Internal Connection ID. Used to indicate operations performed on the same connection.</td>
</tr>
<tr>
<td>104</td>
<td>LDAP_MESSAGE_ID</td>
<td>CHAR</td>
<td>8</td>
<td>1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11</td>
<td>Message ID from BER. Used to connect events</td>
</tr>
<tr>
<td>105</td>
<td>LDAP_BIND_DN</td>
<td>CHAR</td>
<td>512</td>
<td>1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11</td>
<td>Bind DN for the connection</td>
</tr>
<tr>
<td>106</td>
<td>LDAP_CLIENT_SECL</td>
<td>CHAR</td>
<td>32</td>
<td>1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11</td>
<td>LDAP client security label</td>
</tr>
<tr>
<td>107</td>
<td>LDAP_SRC_IP_ADDR</td>
<td>CHAR</td>
<td>16</td>
<td>1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11</td>
<td>IP of the client request (PC if request came across PC interface; XCF for sysplex replica requests; SLAPI for SLAPI internal requests)</td>
</tr>
<tr>
<td>108</td>
<td>LDAP_AUDIT_VERSION</td>
<td>CHAR</td>
<td>2</td>
<td>1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11</td>
<td>Version of the LDAP audit support in use</td>
</tr>
<tr>
<td>109</td>
<td>LDAP_EVENT_CODE</td>
<td>CHAR</td>
<td>2</td>
<td>1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11</td>
<td>Event number 1 - 11</td>
</tr>
<tr>
<td>110</td>
<td>LDAP_RESERVED_04</td>
<td>CHAR</td>
<td>242</td>
<td>1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11</td>
<td>Reserved</td>
</tr>
<tr>
<td>111</td>
<td>LDAP_MAPCERT_OPT</td>
<td>CHAR</td>
<td>2</td>
<td>1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11</td>
<td>sslMapCertificate configuration option (see 793)</td>
</tr>
<tr>
<td>112</td>
<td>LDAP_MAPPED_SAFID</td>
<td>CHAR</td>
<td>8</td>
<td>1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11</td>
<td>SAF ID (if any) associated with the bind DN</td>
</tr>
<tr>
<td>201</td>
<td>LDAP_PROTOCOL_VER</td>
<td>CHAR</td>
<td>2</td>
<td>1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11</td>
<td>Request version 2 or 3</td>
</tr>
<tr>
<td>202</td>
<td>LDAP_RETURN_CODE</td>
<td>INT</td>
<td>10</td>
<td>1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11</td>
<td>Return code in hex (blank if no return code was provided)</td>
</tr>
<tr>
<td>203</td>
<td>LDAP_ERROR_MSG</td>
<td>CHAR</td>
<td>256</td>
<td>1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11</td>
<td>Reason code message number and text</td>
</tr>
<tr>
<td>204</td>
<td>LDAP_ENTRY_NM</td>
<td>CHAR</td>
<td>512</td>
<td>1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11</td>
<td>Target entry of the operation (Bind DN, DN to be deleted, DN to be added, and so on)</td>
</tr>
<tr>
<td>Relocate #</td>
<td>Field name</td>
<td>Type</td>
<td>Length</td>
<td>Event code</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>--------------------------</td>
<td>------</td>
<td>----------</td>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>205</td>
<td>LDAP_RELOC_REQ_TS</td>
<td>CHAR</td>
<td>32</td>
<td>1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11</td>
<td>Time the request was received (day month date hh:mm:ss year)</td>
</tr>
<tr>
<td>206</td>
<td>LDAP_ENTRY_SUFFIX</td>
<td>CHAR</td>
<td>64</td>
<td>1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11</td>
<td>Suffix of the DN</td>
</tr>
<tr>
<td>207</td>
<td>LDAP_CNTRLS_PRESNT</td>
<td>CHAR</td>
<td>512</td>
<td>1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11</td>
<td>Control OID - Criticality pairs for each control</td>
</tr>
<tr>
<td>208</td>
<td>LDAP_ADD_ATTR</td>
<td>CHAR</td>
<td>64</td>
<td>1</td>
<td>Name of attribute added Will have one of these for each unique attribute added (with a maximum of 20, note allowing 1 space in between each value)</td>
</tr>
<tr>
<td>209</td>
<td>LDAP_BIND_MECH</td>
<td>CHAR</td>
<td>16</td>
<td>2</td>
<td>Simple, external, GSSAPI, DIGEST-MD5, CRAM-MD5</td>
</tr>
<tr>
<td>210</td>
<td>LDAP_COMPARE_ATTR</td>
<td>CHAR</td>
<td>64</td>
<td>3</td>
<td>Attribute being compared</td>
</tr>
<tr>
<td>211</td>
<td>LDAP_MOD_ATTR_DEL</td>
<td>CHAR</td>
<td>64 (max)</td>
<td>8</td>
<td>Name of an attribute that was deleted. Will have one of these for each unique attribute (with a maximum of 14, note allowing 1 space in between each value)</td>
</tr>
<tr>
<td>212</td>
<td>LDAP_MOD_ATTR_ADD</td>
<td>CHAR</td>
<td>64 (max)</td>
<td>8</td>
<td>Name of an attribute that was added. Will have one of these for each unique attribute (with a maximum of 14, allowing 1 space in between each value)</td>
</tr>
<tr>
<td>213</td>
<td>LDAP_MOD_ATTR_REP</td>
<td>CHAR</td>
<td>64 (max)</td>
<td>8</td>
<td>Name of an attribute to that was replaced. Will have one of these for each unique attribute (with a maximum of 14, note allowing 1 space in between each value)</td>
</tr>
<tr>
<td>214</td>
<td>LDAP_MDN_NEW_RDN</td>
<td>CHAR</td>
<td>32</td>
<td>9</td>
<td>New relative distinguished name for target entry</td>
</tr>
<tr>
<td>215</td>
<td>LDAP_MDN_DOLD_RDN</td>
<td>CHAR</td>
<td>1</td>
<td>9</td>
<td>T (true) or F (false), should the old RDN be deleted</td>
</tr>
<tr>
<td>216</td>
<td>LDAP_MDN_NEW_SUP</td>
<td>CHAR</td>
<td>512</td>
<td>9</td>
<td>New parent of target entry</td>
</tr>
<tr>
<td>218</td>
<td>LDAP_SEARCH_FILTER</td>
<td>CHAR</td>
<td>256</td>
<td>10</td>
<td>Search filter attribute</td>
</tr>
<tr>
<td>219</td>
<td>LDAP_SEARCH_SCOPE</td>
<td>CHAR</td>
<td>12</td>
<td>10</td>
<td>Subtree search, base search,</td>
</tr>
<tr>
<td>220</td>
<td>LDAP_SEARCH_ATTRS</td>
<td>CHAR</td>
<td>64</td>
<td>10</td>
<td>Attribute requested to be returned on the search. One of these for each</td>
</tr>
<tr>
<td>221</td>
<td>LDAP_XOP_REQ_NAME</td>
<td>CHAR</td>
<td>64</td>
<td>11</td>
<td>OID for extended operation</td>
</tr>
</tbody>
</table>

Note: The LDAP_MAPCERT_OPT value represents the value of the `sslMapCertificate` option in the LDAP server configuration file:

- 00 not an external bind request or `sslMapCertificate off ignore/fail` specified
- 01 `sslMapCertificate check ignore`
- 02 `sslMapCertificate add ignore`
03 sslMapCertificate replace ignore
11 sslMapCertificate check fail
12 sslMapCertificate add fail
13 sslMapCertificate replace fail

**RACF SMF unload utility output**

A general description of the format of the tabular records created by the SMF unload utility can be found in the [z/OS Security Server RACF Macros and Interfaces](https://www.ibm.com) in the topic “The format of the unloaded SMF type83 data”.

The data following the header in the tabular record varies according to the LDAP event that was recorded. The LDAP event types are:

<table>
<thead>
<tr>
<th>Event code from SMF type 83 subtype 3 records</th>
<th>Tabular output Event type strings</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 *ADD</td>
<td></td>
</tr>
<tr>
<td>2 *BIND</td>
<td></td>
</tr>
<tr>
<td>3 *COMPARE</td>
<td></td>
</tr>
<tr>
<td>4 *CONN</td>
<td></td>
</tr>
<tr>
<td>5 *DELETE</td>
<td></td>
</tr>
<tr>
<td>6 *DISCONNECT</td>
<td></td>
</tr>
<tr>
<td>7 *EXOP</td>
<td></td>
</tr>
<tr>
<td>8 *MODIFY</td>
<td></td>
</tr>
<tr>
<td>9 *MODDN</td>
<td></td>
</tr>
<tr>
<td>10 *SEARCH</td>
<td></td>
</tr>
<tr>
<td>11 *UNBIND</td>
<td></td>
</tr>
</tbody>
</table>

Event qualifiers are the same for all LDAP event codes:

<table>
<thead>
<tr>
<th>Event qualifier</th>
<th>Event qualifier number</th>
<th>Event description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUCCESS</td>
<td>0</td>
<td>Audit of successful LDAP operation</td>
</tr>
<tr>
<td>FAILURE</td>
<td>1</td>
<td>Audit of failed LDAP operation</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Field name</th>
<th>Type</th>
<th>Length</th>
<th>Position</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAP_RESERVED_01</td>
<td>CHAR</td>
<td>16</td>
<td>3000</td>
<td>3015</td>
</tr>
<tr>
<td>LDAP_REQ_TIMESTP</td>
<td>CHAR</td>
<td>16</td>
<td>3018</td>
<td>3033</td>
</tr>
<tr>
<td>LDAP_SERVER_URL</td>
<td>CHAR</td>
<td>32</td>
<td>3036</td>
<td>3067</td>
</tr>
<tr>
<td>LDAP_CONN_ID</td>
<td>CHAR</td>
<td>8</td>
<td>3070</td>
<td>3077</td>
</tr>
</tbody>
</table>
Table 84. Event specific fields for LDAP add event (Event code 1) (continued)

<table>
<thead>
<tr>
<th>Field name</th>
<th>Type</th>
<th>Length</th>
<th>Position</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAP_MESSAGE_ID</td>
<td>CHAR</td>
<td>8</td>
<td>3080</td>
<td>Message ID from BER. Used to connect events.</td>
</tr>
<tr>
<td>LDAP_BIND_DN</td>
<td>CHAR</td>
<td>512</td>
<td>3090</td>
<td>Bind DN for the connection</td>
</tr>
<tr>
<td>LDAP_CLIENT_SECL</td>
<td>CHAR</td>
<td>32</td>
<td>3604</td>
<td>LDAP client security label</td>
</tr>
<tr>
<td>LDAP_SRC_IP_ADDR</td>
<td>CHAR</td>
<td>16</td>
<td>3638</td>
<td>IP of the client request (PC if request came across PC interface; XCF for sysplex replica requests; SLAPI for SLAPI internal requests)</td>
</tr>
<tr>
<td>LDAP_AUDIT_VERSION</td>
<td>CHAR</td>
<td>2</td>
<td>3656</td>
<td>Version of the LDAP audit support in use</td>
</tr>
<tr>
<td>LDAP_EVENT_CODE</td>
<td>CHAR</td>
<td>2</td>
<td>3660</td>
<td>Event number 1 (add)</td>
</tr>
<tr>
<td>LDAP_MAPCERT_OPT</td>
<td>CHAR</td>
<td>2</td>
<td>3664</td>
<td>Value is always 00</td>
</tr>
<tr>
<td>LDAP_MAPPED_SAFID</td>
<td>CHAR</td>
<td>8</td>
<td>3668</td>
<td>SAF ID (if any) associated with the bind DN</td>
</tr>
<tr>
<td>LDAP_RESERVED_04</td>
<td>CHAR</td>
<td>242</td>
<td>3678</td>
<td>Reserved</td>
</tr>
<tr>
<td>LDAP_PROTOCOL_VER</td>
<td>CHAR</td>
<td>2</td>
<td>3922</td>
<td>Request version 2 or 3</td>
</tr>
<tr>
<td>LDAP_RETURN_CODE</td>
<td>INT</td>
<td>10</td>
<td>3926</td>
<td>Return code in hex (blank if no return code was provided)</td>
</tr>
<tr>
<td>LDAP_ERROR_MSG</td>
<td>CHAR</td>
<td>256</td>
<td>3938</td>
<td>Reason code, message number and text</td>
</tr>
<tr>
<td>LDAP_ENTRY_NM</td>
<td>CHAR</td>
<td>512</td>
<td>4196</td>
<td>Target entry of the operation (Bind DN, DN to be deleted, DN to be added, and so on)</td>
</tr>
<tr>
<td>LDAP_RELOC_REQ_TS</td>
<td>CHAR</td>
<td>32</td>
<td>4710</td>
<td>Time the request was received (day month date hh:mm:ss year)</td>
</tr>
<tr>
<td>LDAP_ENTRY_SUFFIX</td>
<td>CHAR</td>
<td>64</td>
<td>4744</td>
<td>Suffix of the DN.</td>
</tr>
<tr>
<td>LDAP_CNTRLS_PRESNT</td>
<td>CHAR</td>
<td>512</td>
<td>4810</td>
<td>Control OID - Criticality pairs for each control (Note, no value)</td>
</tr>
<tr>
<td>LDAP_ADD_ATTR</td>
<td>CHAR</td>
<td>64 (maximum 1299)</td>
<td>5324</td>
<td>6622</td>
</tr>
</tbody>
</table>

Table 85. Event specific fields for LDAP bind event (Event code 2)

<table>
<thead>
<tr>
<th>Field name</th>
<th>Type</th>
<th>Length</th>
<th>Position</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAP_RESERVED_01</td>
<td>CHAR</td>
<td>16</td>
<td>3000</td>
<td>Reserved</td>
</tr>
<tr>
<td>LDAP_REQ_TIMESTP</td>
<td>CHAR</td>
<td>16</td>
<td>3018</td>
<td>Elapsed time for request completion (in microseconds)</td>
</tr>
<tr>
<td>LDAP_SERVER_URL</td>
<td>CHAR</td>
<td>32</td>
<td>3036</td>
<td>The auditing server’s URL for the connection that the client came in on (Listen URL)</td>
</tr>
</tbody>
</table>
Table 85. Event specific fields for LDAP bind event (Event code 2) (continued)

<table>
<thead>
<tr>
<th>Field name</th>
<th>Type</th>
<th>Length</th>
<th>Position</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAP_CONN_ID</td>
<td>CHAR</td>
<td>8</td>
<td>3070</td>
<td>Internal connection ID. Used to indicate operations performed on the same connection</td>
</tr>
<tr>
<td>LDAP_MESSAGE_ID</td>
<td>CHAR</td>
<td>8</td>
<td>3080</td>
<td>Message ID from BER. Used to connect events.</td>
</tr>
<tr>
<td>LDAP_BIND_DN</td>
<td>CHAR</td>
<td>512</td>
<td>3090</td>
<td>Bind DN for the connection</td>
</tr>
<tr>
<td>LDAP_CLIENT_SECL</td>
<td>CHAR</td>
<td>32</td>
<td>3604</td>
<td>LDAP client security label</td>
</tr>
<tr>
<td>LDAP_SRC_IP_ADDR</td>
<td>CHAR</td>
<td>16</td>
<td>3638</td>
<td>IP of the client request (PC if request came across PC interface; XCF for sysplex replica requests; SLAPI for SLAPI internal requests)</td>
</tr>
<tr>
<td>LDAP_AUDIT_VERSION</td>
<td>CHAR</td>
<td>2</td>
<td>3656</td>
<td>Version of the LDAP audit support in use</td>
</tr>
<tr>
<td>LDAP_EVENT_CODE</td>
<td>CHAR</td>
<td>2</td>
<td>3660</td>
<td>Event number 2 (bind)</td>
</tr>
<tr>
<td>LDAP_MAPCERT_OPT</td>
<td>CHAR</td>
<td>2</td>
<td>3664</td>
<td>Represents value of sslMapCertificate configuration option</td>
</tr>
<tr>
<td>LDAP_MAPPED_SAFID</td>
<td>CHAR</td>
<td>8</td>
<td>3668</td>
<td>SAF ID (if any) associated with the bind DN</td>
</tr>
<tr>
<td>LDAP_RESERVED_04</td>
<td>CHAR</td>
<td>242</td>
<td>3678</td>
<td>Reserved</td>
</tr>
<tr>
<td>LDAP_PROTOCOL_VER</td>
<td>CHAR</td>
<td>2</td>
<td>3922</td>
<td>Request version 2 or 3</td>
</tr>
<tr>
<td>LDAP_RETURN_CODE</td>
<td>INT</td>
<td>10</td>
<td>3926</td>
<td>Return Code in hex (blank if no return code was provided)</td>
</tr>
<tr>
<td>LDAP_ERROR_MSG</td>
<td>CHAR</td>
<td>256</td>
<td>3938</td>
<td>Reason Code, message number and text</td>
</tr>
<tr>
<td>LDAP_ENTRY_NM</td>
<td>CHAR</td>
<td>512</td>
<td>4196</td>
<td>Target entry of the operation (Bind DN, DN to be deleted, DN to be added, and so on)</td>
</tr>
<tr>
<td>LDAP_RELOC_REQ_TS</td>
<td>CHAR</td>
<td>32</td>
<td>4710</td>
<td>Time the request was received (day month date hh:mm:ss year)</td>
</tr>
<tr>
<td>LDAP_ENTRY_SUFFIX</td>
<td>CHAR</td>
<td>64</td>
<td>4744</td>
<td>Suffix of the DN.</td>
</tr>
<tr>
<td>LDAP_CNTRLS_PRESENT</td>
<td>CHAR</td>
<td>512</td>
<td>4810</td>
<td>Control OID - Criticality pairs for each control (Note, no value)</td>
</tr>
<tr>
<td>LDAP_BIND_MECH</td>
<td>CHAR</td>
<td>16</td>
<td>5324</td>
<td>Simple, external, GSSAPI, DIGEST-MDS, CRAM-MD5</td>
</tr>
</tbody>
</table>

Table 86. Event specific fields for LDAP compare event (Event code 3)

<table>
<thead>
<tr>
<th>Field name</th>
<th>Type</th>
<th>Length</th>
<th>Position</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAP_RESERVED_01</td>
<td>CHAR</td>
<td>16</td>
<td>3000</td>
<td>3015</td>
</tr>
<tr>
<td>LDAP_REQ_TIMESTP</td>
<td>CHAR</td>
<td>16</td>
<td>3018</td>
<td>3033</td>
</tr>
</tbody>
</table>
### Table 86. Event specific fields for LDAP compare event (Event code 3) (continued)

<table>
<thead>
<tr>
<th>Field name</th>
<th>Type</th>
<th>Length</th>
<th>Start</th>
<th>End</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAP_SERVER_URL</td>
<td>CHAR</td>
<td>32</td>
<td>3036</td>
<td>3067</td>
<td>The auditing server’s URL for the connection that the client came in on (Listen URL)</td>
</tr>
<tr>
<td>LDAP_CONN_ID</td>
<td>CHAR</td>
<td>8</td>
<td>3070</td>
<td>3077</td>
<td>Internal connection ID. Used to indicate operations performed on the same connection</td>
</tr>
<tr>
<td>LDAP_MESSAGE_ID</td>
<td>CHAR</td>
<td>8</td>
<td>3080</td>
<td>3087</td>
<td>Message ID from BER. Used to connect events</td>
</tr>
<tr>
<td>LDAP_BIND_DN</td>
<td>CHAR</td>
<td>512</td>
<td>3090</td>
<td>3601</td>
<td>Bind DN for the connection</td>
</tr>
<tr>
<td>LDAP_CLIENT_SECL</td>
<td>CHAR</td>
<td>32</td>
<td>3604</td>
<td>3635</td>
<td>LDAP client security label</td>
</tr>
<tr>
<td>LDAP_SRC_IP_ADDR</td>
<td>CHAR</td>
<td>16</td>
<td>3638</td>
<td>3653</td>
<td>IP of the client request (PC if request came across PC interface; XCF for sysplex replica requests; SLAPI for SLAPI internal requests)</td>
</tr>
<tr>
<td>LDAP_AUDIT_VERSION</td>
<td>CHAR</td>
<td>2</td>
<td>3656</td>
<td>3657</td>
<td>Version of the LDAP audit support in use</td>
</tr>
<tr>
<td>LDAP_EVENT_CODE</td>
<td>CHAR</td>
<td>2</td>
<td>3660</td>
<td>3661</td>
<td>Event number 3 (compare)</td>
</tr>
<tr>
<td>LDAP_MAPCERT_OPT</td>
<td>CHAR</td>
<td>2</td>
<td>3664</td>
<td>3665</td>
<td>Value is always 00</td>
</tr>
<tr>
<td>LDAP_MAPPED_SAFID</td>
<td>CHAR</td>
<td>8</td>
<td>3668</td>
<td>3675</td>
<td>SAF ID (if any) associated with the bind DN</td>
</tr>
<tr>
<td>LDAP_RESERVED_04</td>
<td>CHAR</td>
<td>242</td>
<td>3678</td>
<td>3919</td>
<td>Reserved</td>
</tr>
<tr>
<td>LDAP_PROTOCOL_VER</td>
<td>CHAR</td>
<td>2</td>
<td>3922</td>
<td>3923</td>
<td>Request version 2 or 3</td>
</tr>
<tr>
<td>LDAP_RETURN_CODE</td>
<td>INT</td>
<td>10</td>
<td>3926</td>
<td>3935</td>
<td>Return Code in hex (blank if no return code was provided)</td>
</tr>
<tr>
<td>LDAP_ERROR_MSG</td>
<td>CHAR</td>
<td>256</td>
<td>3938</td>
<td>4193</td>
<td>Reason Code, message number and text</td>
</tr>
<tr>
<td>LDAP_ENTRY_NM</td>
<td>CHAR</td>
<td>512</td>
<td>4196</td>
<td>4707</td>
<td>Target Entry of the operation (Bind DN, DN to be deleted, DN to be added, and so on)</td>
</tr>
<tr>
<td>LDAP_RELOC_REQ_TS</td>
<td>CHAR</td>
<td>32</td>
<td>4710</td>
<td>4741</td>
<td>Time the request was received (day month date hh:mm:ss year)</td>
</tr>
<tr>
<td>LDAP_ENTRY_SUFFIX</td>
<td>CHAR</td>
<td>64</td>
<td>4744</td>
<td>4807</td>
<td>Suffix of the DN.</td>
</tr>
<tr>
<td>LDAP_CNTRSLS_PRESNT</td>
<td>CHAR</td>
<td>512</td>
<td>4810</td>
<td>5321</td>
<td>Control OID - Criticality pairs for each control (Note, no value)</td>
</tr>
<tr>
<td>LDAPCOMPARE_ATTR</td>
<td>CHAR</td>
<td>64</td>
<td>5324</td>
<td>5387</td>
<td>Attribute being compared.</td>
</tr>
</tbody>
</table>

### Table 87. Event specific fields for LDAP connect, delete, disconnect, and unbind events (Event code 4, 5, 6, 11)

<table>
<thead>
<tr>
<th>Field name</th>
<th>Type</th>
<th>Length</th>
<th>Start</th>
<th>End</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAP_RESERVED_01</td>
<td>CHAR</td>
<td>16</td>
<td>3000</td>
<td>3015</td>
<td>Reserved</td>
</tr>
<tr>
<td>LDAP_REQ_TIMESTP</td>
<td>CHAR</td>
<td>16</td>
<td>3018</td>
<td>3033</td>
<td>Elapsed time for request completion (in microseconds)</td>
</tr>
</tbody>
</table>
Table 87. Event specific fields for LDAP connect, delete, disconnect, and unbind events (Event code 4, 5, 6, 11) (continued)

<table>
<thead>
<tr>
<th>Field name</th>
<th>Type</th>
<th>Length</th>
<th>Position</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAP_SERVER_URL</td>
<td>CHAR</td>
<td>32</td>
<td>3036</td>
<td>3067 The auditing server's URL for the connection that the client came in on (Listen URL)</td>
</tr>
<tr>
<td>LDAP_CONN_ID</td>
<td>CHAR</td>
<td>8</td>
<td>3070</td>
<td>3077 Internal connection ID. Used to indicate operations performed on the same connection</td>
</tr>
<tr>
<td>LDAP_MESSAGE_ID</td>
<td>CHAR</td>
<td>8</td>
<td>3080</td>
<td>3087 Message ID from BER. Used to connect events.</td>
</tr>
<tr>
<td>LDAP_BIND_DN</td>
<td>CHAR</td>
<td>512</td>
<td>3090</td>
<td>3601 Bind DN for the connection</td>
</tr>
<tr>
<td>LDAP_CLIENT_SECL</td>
<td>CHAR</td>
<td>32</td>
<td>3604</td>
<td>3635 LDAP client security label</td>
</tr>
<tr>
<td>LDAP_SRC_IP_ADDR</td>
<td>CHAR</td>
<td>16</td>
<td>3638</td>
<td>3653 IP of the client request (PC if request came across PC interface; XCF for sysplex replica requests; SLAPI for SLAPI internal requests)</td>
</tr>
<tr>
<td>LDAP_AUDIT_VERSION</td>
<td>CHAR</td>
<td>2</td>
<td>3656</td>
<td>3657 Version of the LDAP audit support in use</td>
</tr>
<tr>
<td>LDAP_EVENT_CODE</td>
<td>CHAR</td>
<td>2</td>
<td>3660</td>
<td>3661 Event number 4, 5, 6, 11 (connect, delete, disconnect, unbind)</td>
</tr>
<tr>
<td>LDAP_MAPCERT_OPT</td>
<td>CHAR</td>
<td>2</td>
<td>3664</td>
<td>3665 Value is always 00</td>
</tr>
<tr>
<td>LDAP_MAPPED_SAFID</td>
<td>CHAR</td>
<td>8</td>
<td>3668</td>
<td>3675 SAF ID (if any) associated with the bind DN</td>
</tr>
<tr>
<td>LDAP_RESERVED_04</td>
<td>CHAR</td>
<td>242</td>
<td>3678</td>
<td>3919 Reserved</td>
</tr>
<tr>
<td>LDAP_PROTOCOL_VER</td>
<td>CHAR</td>
<td>2</td>
<td>3922</td>
<td>3923 Request version 2 or 3</td>
</tr>
<tr>
<td>LDAP_RETURN_CODE</td>
<td>INT</td>
<td>10</td>
<td>3926</td>
<td>3935 Return Code in hex (blank if no return code was provided)</td>
</tr>
<tr>
<td>LDAP_ERROR_MSG</td>
<td>CHAR</td>
<td>256</td>
<td>3938</td>
<td>4193 Reason Code, message number and text</td>
</tr>
<tr>
<td>LDAP_ENTRY_NM</td>
<td>CHAR</td>
<td>512</td>
<td>4196</td>
<td>4707 Target Entry of the operation (Bind DN, DN to be deleted, DN to be added, and so on)</td>
</tr>
<tr>
<td>LDAP_RELOC_REQ_TS</td>
<td>CHAR</td>
<td>32</td>
<td>4710</td>
<td>4741 Time the request was received (day month date hh:mm:ss year)</td>
</tr>
<tr>
<td>LDAP_ENTRY_SUFFIX</td>
<td>CHAR</td>
<td>64</td>
<td>4744</td>
<td>4807 Suffix of the DN</td>
</tr>
<tr>
<td>LDAP_CNTRL_04</td>
<td>CHAR</td>
<td>512</td>
<td>4810</td>
<td>5321 Control OID - Criticality pairs for each control (Note, no value)</td>
</tr>
</tbody>
</table>

Table 88. Event specific fields for LDAP extended operations event (Event code 7)

<table>
<thead>
<tr>
<th>Field name</th>
<th>Type</th>
<th>Length</th>
<th>Position</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAP_RESERVED_01</td>
<td>CHAR</td>
<td>16</td>
<td>3000</td>
<td>3015 Reserved</td>
</tr>
<tr>
<td>LDAP_REQ_TIMESTEP</td>
<td>CHAR</td>
<td>16</td>
<td>3018</td>
<td>3033 Elapsed time for request completion (in microseconds)</td>
</tr>
</tbody>
</table>
### Table 88. Event specific fields for LDAP extended operations event (Event code 7) (continued)

<table>
<thead>
<tr>
<th>Field name</th>
<th>Type</th>
<th>Length</th>
<th>Position</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAP_SERVER_URL</td>
<td>CHAR</td>
<td>32</td>
<td>3036</td>
<td>3067</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>The auditing server’s URL for the connection that the client came in on</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(Listen URL)</td>
</tr>
<tr>
<td>LDAP_CONN_ID</td>
<td>CHAR</td>
<td>8</td>
<td>3070</td>
<td>3077</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Internal connection ID. Used to indicate operations performed on the same</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>connection</td>
</tr>
<tr>
<td>LDAP_MESSAGE_ID</td>
<td>CHAR</td>
<td>8</td>
<td>3080</td>
<td>3087</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Message ID from BER. Used to connect events.</td>
</tr>
<tr>
<td>LDAP_BIND_DN</td>
<td>CHAR</td>
<td>512</td>
<td>3090</td>
<td>3601</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Bind DN for the connection</td>
</tr>
<tr>
<td>LDAP_CLIENT_SECL</td>
<td>CHAR</td>
<td>32</td>
<td>3604</td>
<td>3635</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>LDAP client security label</td>
</tr>
<tr>
<td>LDAP_SRC_IP_ADDR</td>
<td>CHAR</td>
<td>16</td>
<td>3638</td>
<td>3653</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>IP of the client request (PC if request came across PC interface; XCF for</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>sysplex replica requests; SLAPI for SLAPI internal requests)</td>
</tr>
<tr>
<td>LDAP_AUDIT_VERSION</td>
<td>CHAR</td>
<td>2</td>
<td>3656</td>
<td>3657</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Version of the LDAP audit support in use</td>
</tr>
<tr>
<td>LDAP_EVENT_CODE</td>
<td>CHAR</td>
<td>2</td>
<td>3660</td>
<td>3661</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Event number 7 (extended operations)</td>
</tr>
<tr>
<td>LDAP_MAPCERT_OPT</td>
<td>CHAR</td>
<td>2</td>
<td>3664</td>
<td>3665</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Value is always 00</td>
</tr>
<tr>
<td>LDAP_MAPPED_SAFID</td>
<td>CHAR</td>
<td>8</td>
<td>3668</td>
<td>3675</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SAF ID (if any) associated with the bind DN</td>
</tr>
<tr>
<td>LDAP_RESERVED_04</td>
<td>CHAR</td>
<td>242</td>
<td>3678</td>
<td>3919</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>LDAP_PROTOCOL_VER</td>
<td>CHAR</td>
<td>2</td>
<td>3922</td>
<td>3923</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Request version 2 or 3</td>
</tr>
<tr>
<td>LDAP_RETURN_CODE</td>
<td>INT</td>
<td>10</td>
<td>3926</td>
<td>3935</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Return code in hex (blank if no return code was provided)</td>
</tr>
<tr>
<td>LDAP_ERROR_MSG</td>
<td>CHAR</td>
<td>256</td>
<td>3938</td>
<td>4193</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Reason code, message number and text</td>
</tr>
<tr>
<td>LDAP_ENTRY_NM</td>
<td>CHAR</td>
<td>512</td>
<td>4196</td>
<td>4707</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Target entry of the operation (Bind DN, DN to be deleted, DN to be added,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>and so on)</td>
</tr>
<tr>
<td>LDAP_RELOC_REQ_TS</td>
<td>CHAR</td>
<td>32</td>
<td>4710</td>
<td>4741</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Time the request was received</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(day month date hh:mm:ss year)</td>
</tr>
<tr>
<td>LDAP_ENTRY_SUFFIX</td>
<td>CHAR</td>
<td>64</td>
<td>4744</td>
<td>4807</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Suffix of the DN.</td>
</tr>
<tr>
<td>LDAP_CNTRLS_PRESNT</td>
<td>CHAR</td>
<td>512</td>
<td>4810</td>
<td>5321</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Control OID - Criticality pairs for each control (Note, no value)</td>
</tr>
<tr>
<td>LDAP_XOP_REQ_NAME</td>
<td>CHAR</td>
<td>64</td>
<td>5324</td>
<td>5387</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>OID for extended operations</td>
</tr>
</tbody>
</table>

### Table 89. Event specific fields for LDAP modify event (Event code 8)

<table>
<thead>
<tr>
<th>Field name</th>
<th>Type</th>
<th>Length</th>
<th>Position</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAP_RESERVED_01</td>
<td>CHAR</td>
<td>16</td>
<td>3000</td>
<td>3015</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>LDAP_REQ_TIMESTP</td>
<td>CHAR</td>
<td>16</td>
<td>3018</td>
<td>3033</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Elapsed time for request completion (in microseconds)</td>
</tr>
</tbody>
</table>
Table 89. Event specific fields for LDAP modify event (Event code 8) (continued)

<table>
<thead>
<tr>
<th>Field name</th>
<th>Type</th>
<th>Length</th>
<th>Position</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAP_SERVER_URL</td>
<td>CHAR</td>
<td>32</td>
<td>3036</td>
<td>3067</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>The auditing server’s URL for the connection that the client came in on (Listen URL)</td>
</tr>
<tr>
<td>LDAP_CONN_ID</td>
<td>CHAR</td>
<td>8</td>
<td>3070</td>
<td>3077</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Internal connection ID. Used to indicate operations performed on the same connection</td>
</tr>
<tr>
<td>LDAP_MESSAGE_ID</td>
<td>CHAR</td>
<td>8</td>
<td>3080</td>
<td>3087</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Message ID from BER. Used to connect events.</td>
</tr>
<tr>
<td>LDAP_BIND_DN</td>
<td>CHAR</td>
<td>512</td>
<td>3090</td>
<td>3601</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Bind DN for the connection</td>
</tr>
<tr>
<td>LDAP_CLIENT_SECL</td>
<td>CHAR</td>
<td>32</td>
<td>3604</td>
<td>3635</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>LDAP client security label</td>
</tr>
<tr>
<td>LDAP_SRC_IP_ADDR</td>
<td>CHAR</td>
<td>16</td>
<td>3638</td>
<td>3653</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>IP of the client request (PC if request came across PC interface; XCF for sysplex replica requests; SLAPI for SLAPI internal requests)</td>
</tr>
<tr>
<td>LDAP_AUDIT_ VERSION</td>
<td>CHAR</td>
<td>2</td>
<td>3656</td>
<td>3657</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Version of the LDAP audit support in use</td>
</tr>
<tr>
<td>LDAP_EVENT_CODE</td>
<td>CHAR</td>
<td>2</td>
<td>3660</td>
<td>3661</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Event number 8 (modify)</td>
</tr>
<tr>
<td>LDAP_MAPCERT_ OPT</td>
<td>CHAR</td>
<td>2</td>
<td>3664</td>
<td>3665</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Value is always 00</td>
</tr>
<tr>
<td>LDAP_MAPPED_ SAFID</td>
<td>CHAR</td>
<td>8</td>
<td>3668</td>
<td>3675</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SAF ID (if any) associated with the bind DN</td>
</tr>
<tr>
<td>LDAP_RESERVED_04</td>
<td>CHAR</td>
<td>242</td>
<td>3678</td>
<td>3919</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>LDAP_PROTOCOL_ VER</td>
<td>CHAR</td>
<td>2</td>
<td>3922</td>
<td>3923</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Request version 2 or 3</td>
</tr>
<tr>
<td>LDAP_RETURN_ CODE</td>
<td>INT</td>
<td>10</td>
<td>3926</td>
<td>3935</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Return code in hex (blank if no return code was provided)</td>
</tr>
<tr>
<td>LDAP_ERROR_MSG</td>
<td>CHAR</td>
<td>256</td>
<td>3938</td>
<td>4193</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Reason code, message number and text</td>
</tr>
<tr>
<td>LDAP_ENTRY_NM</td>
<td>CHAR</td>
<td>512</td>
<td>4196</td>
<td>4707</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Target entry of the operation (Bind DN, DN to be deleted, DN to be added, and so on)</td>
</tr>
<tr>
<td>LDAP_RELOC_REQ_TS</td>
<td>CHAR</td>
<td>32</td>
<td>4710</td>
<td>4741</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Time the request was received (day month date hh:mm:ss year)</td>
</tr>
<tr>
<td>LDAP_ENTRY_SUFFIX</td>
<td>CHAR</td>
<td>64</td>
<td>4744</td>
<td>4807</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Suffix of the DN.</td>
</tr>
<tr>
<td>LDAP_CNTRLS_ PRESNT</td>
<td>CHAR</td>
<td>512</td>
<td>4810</td>
<td>5321</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Control OID - Criticality pairs for each control (Note, no value)</td>
</tr>
<tr>
<td>LDAP_MOD_ATTR_DEL</td>
<td>CHAR</td>
<td>64</td>
<td>5324</td>
<td>6233</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Name of attribute that was deleted. Will have one of these for each unique attr (with a maximum of 14, note allowing 1 space in between each value).</td>
</tr>
<tr>
<td>LDAP_MOD_ATTR_ADD</td>
<td>CHAR</td>
<td>64</td>
<td>6236</td>
<td>7145</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Name of an attribute that was added. Will have one of these for each unique attr (with a maximum of 14, note allowing 1 space in between each value).</td>
</tr>
</tbody>
</table>
Table 89. Event specific fields for LDAP modify event (Event code 8) (continued)

<table>
<thead>
<tr>
<th>Field name</th>
<th>Type</th>
<th>Length</th>
<th>Position</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAP_MOD_ATTR_REP</td>
<td>CHAR</td>
<td>64 (maximum 909)</td>
<td>7148-8057</td>
<td>Name of an attribute that was replaced. Will have one of these for each unique attr (with a maximum of 145, note allowing 1 space in between each value).</td>
</tr>
</tbody>
</table>

Table 90. Event specific fields for LDAP modify DN event (Event code 9)

<table>
<thead>
<tr>
<th>Field name</th>
<th>Type</th>
<th>Length</th>
<th>Position</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAP_RESERVED_01</td>
<td>CHAR</td>
<td>16</td>
<td>3000-3015</td>
<td>Reserved</td>
</tr>
<tr>
<td>LDAP_REQ_TIMESTP</td>
<td>CHAR</td>
<td>16</td>
<td>3018-3033</td>
<td>Elapsed time for request completion (in microseconds)</td>
</tr>
<tr>
<td>LDAP_SERVER_URL</td>
<td>CHAR</td>
<td>32</td>
<td>3036-3067</td>
<td>The auditing server's URL for the connection that the client came in on (Listen URL)</td>
</tr>
<tr>
<td>LDAP_CONN_ID</td>
<td>CHAR</td>
<td>8</td>
<td>3070-3077</td>
<td>Internal connection ID. Used to indicate operations performed on the same connection</td>
</tr>
<tr>
<td>LDAP_MESSAGE_ID</td>
<td>CHAR</td>
<td>8</td>
<td>3080-3087</td>
<td>Message ID from BER. Used to connect events.</td>
</tr>
<tr>
<td>LDAP_BIND_DN</td>
<td>CHAR</td>
<td>512</td>
<td>3090-3601</td>
<td>Bind DN for the connection</td>
</tr>
<tr>
<td>LDAP_CLIENT_SECL</td>
<td>CHAR</td>
<td>32</td>
<td>3604-3635</td>
<td>LDAP client security label</td>
</tr>
<tr>
<td>LDAP_SRC_IP_ADDR</td>
<td>CHAR</td>
<td>16</td>
<td>3638-3653</td>
<td>IP of the client request (PC if request came across PC interface; XCF for sysplex replica requests; SLAPI for SLAPI internal requests)</td>
</tr>
<tr>
<td>LDAP_AUDIT_VERSION</td>
<td>CHAR</td>
<td>2</td>
<td>3656-3657</td>
<td>Version of the LDAP audit support in use</td>
</tr>
<tr>
<td>LDAP_EVENT_CODE</td>
<td>CHAR</td>
<td>2</td>
<td>3660-3661</td>
<td>Event number 9 (modify DN)</td>
</tr>
<tr>
<td>LDAP_MAPCERT_OPT</td>
<td>CHAR</td>
<td>2</td>
<td>3664-3665</td>
<td>Value is always 00</td>
</tr>
<tr>
<td>LDAP_MAPPED_SAFID</td>
<td>CHAR</td>
<td>8</td>
<td>3668-3675</td>
<td>SAF ID (if any) associated with the bind DN</td>
</tr>
<tr>
<td>LDAP_RESERVED_04</td>
<td>CHAR</td>
<td>242</td>
<td>3678-3919</td>
<td>Reserved</td>
</tr>
<tr>
<td>LDAP_PROTOCOL_VER</td>
<td>CHAR</td>
<td>2</td>
<td>3922-3923</td>
<td>Request version 2 or 3</td>
</tr>
<tr>
<td>LDAP_RETURN_CODE</td>
<td>INT</td>
<td>10</td>
<td>3926-3935</td>
<td>Return Code in hex (blank if no return code was provided)</td>
</tr>
<tr>
<td>LDAP_ERROR_MSG</td>
<td>CHAR</td>
<td>256</td>
<td>3938-4193</td>
<td>Reason Code, message number and text</td>
</tr>
<tr>
<td>LDAP_ENTRY_NM</td>
<td>CHAR</td>
<td>512</td>
<td>4196-4707</td>
<td>Target Entry of the operation (Bind DN, DN to be deleted, DN to be added, and so on)</td>
</tr>
<tr>
<td>LDAP_RELOC_REQ_TS</td>
<td>CHAR</td>
<td>32</td>
<td>4710-4741</td>
<td>Time the request was received (day month date hh:mm:ss year)</td>
</tr>
<tr>
<td>LDAP_ENTRY_SUFFIX</td>
<td>CHAR</td>
<td>64</td>
<td>4744-4807</td>
<td>Suffix of the DN.</td>
</tr>
</tbody>
</table>
Table 90. Event specific fields for LDAP modify DN event (Event code 9) (continued)

<table>
<thead>
<tr>
<th>Field name</th>
<th>Type</th>
<th>Length</th>
<th>Position</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAP_CNTRLS_PRESENT</td>
<td>CHAR</td>
<td>512</td>
<td>4810</td>
<td>Control OID - Criticality pairs for each control (Note, no value)</td>
</tr>
<tr>
<td>LDAP_MDN_NEW_RDN</td>
<td>CHAR</td>
<td>32</td>
<td>5324</td>
<td>New relative distinguished name for target entry.</td>
</tr>
<tr>
<td>LDAP_MDN_DOLD_RDN</td>
<td>CHAR</td>
<td>1</td>
<td>5358</td>
<td>T (true) or F (false), should the old RDN be deleted.</td>
</tr>
<tr>
<td>LDAP_MDN_NEW_SUP</td>
<td>CHAR</td>
<td>512</td>
<td>5361</td>
<td>New parent of target entry.</td>
</tr>
</tbody>
</table>

Table 91. Event specific fields for LDAP search event (Event code 10)

<table>
<thead>
<tr>
<th>Field name</th>
<th>Type</th>
<th>Length</th>
<th>Position</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAP_RESERVED_01</td>
<td>CHAR</td>
<td>16</td>
<td>3000</td>
<td>Reserved</td>
</tr>
<tr>
<td>LDAP_REQ_TIMESTP</td>
<td>CHAR</td>
<td>16</td>
<td>3018</td>
<td>Elapsed time for request completion (in microseconds)</td>
</tr>
<tr>
<td>LDAP_SERVER_URL</td>
<td>CHAR</td>
<td>32</td>
<td>3036</td>
<td>The auditing server’s URL for the connection that the client came in on (Listen URL)</td>
</tr>
<tr>
<td>LDAP_CONN_ID</td>
<td>CHAR</td>
<td>8</td>
<td>3070</td>
<td>Internal connection ID. Used to indicate operations performed on the same connection</td>
</tr>
<tr>
<td>LDAP_MESSAGE_ID</td>
<td>CHAR</td>
<td>8</td>
<td>3080</td>
<td>Message ID from BER. Used to connect events.</td>
</tr>
<tr>
<td>LDAP_BIND_DN</td>
<td>CHAR</td>
<td>512</td>
<td>3090</td>
<td>Bind DN for the connection</td>
</tr>
<tr>
<td>LDAP_CLIENT_SECL</td>
<td>CHAR</td>
<td>32</td>
<td>3604</td>
<td>LDAP client security label</td>
</tr>
<tr>
<td>LDAP_SRC_IP_ADDR</td>
<td>CHAR</td>
<td>16</td>
<td>3638</td>
<td>IP of the client request (PC if request came across PC interface; XCF for sysplex replica requests; SLAPI for SLAPI internal requests)</td>
</tr>
<tr>
<td>LDAP_AUDIT_VERSION</td>
<td>CHAR</td>
<td>2</td>
<td>3656</td>
<td>Version of the LDAP audit support in use</td>
</tr>
<tr>
<td>LDAP_EVENT_CODE</td>
<td>CHAR</td>
<td>2</td>
<td>3660</td>
<td>Event number 10 (search)</td>
</tr>
<tr>
<td>LDAP_MAPCERT_OPT</td>
<td>CHAR</td>
<td>2</td>
<td>3664</td>
<td>Value is always 00</td>
</tr>
<tr>
<td>LDAP_MAPPED_SAFID</td>
<td>CHAR</td>
<td>8</td>
<td>3668</td>
<td>SAF ID (if any) associated with the bind DN</td>
</tr>
<tr>
<td>LDAP_RESERVED_04</td>
<td>CHAR</td>
<td>242</td>
<td>3678</td>
<td>Reserved</td>
</tr>
<tr>
<td>LDAP_PROTOCOL_VER</td>
<td>CHAR</td>
<td>2</td>
<td>3922</td>
<td>Request version 2 or 3</td>
</tr>
<tr>
<td>LDAP_RETURN_CODE</td>
<td>INT</td>
<td>10</td>
<td>3926</td>
<td>Return Code in hex (blank if no return code was provided)</td>
</tr>
<tr>
<td>LDAP_ERROR_MSG</td>
<td>CHAR</td>
<td>256</td>
<td>3938</td>
<td>Reason Code, message number and text</td>
</tr>
<tr>
<td>LDAP_ENTRY_NM</td>
<td>CHAR</td>
<td>512</td>
<td>4196</td>
<td>Target Entry of the operation (Bind DN, DN to be deleted, DN to be added, and so on)</td>
</tr>
</tbody>
</table>
Table 91. Event specific fields for LDAP search event (Event code 10) (continued)

<table>
<thead>
<tr>
<th>Field name</th>
<th>Type</th>
<th>Length</th>
<th>Position Start</th>
<th>Position End</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAP_RELOC_REQ_TS</td>
<td>CHAR</td>
<td>32</td>
<td>4710</td>
<td>4741</td>
<td>Time the request was received (day month date hh:mm:ss year)</td>
</tr>
<tr>
<td>LDAP_ENTRY_SUFFIX</td>
<td>CHAR</td>
<td>64</td>
<td>4744</td>
<td>4807</td>
<td>Suffix of the DN.</td>
</tr>
<tr>
<td>LDAP_CNTRLS_PRESNT</td>
<td>CHAR</td>
<td>512</td>
<td>4810</td>
<td>5321</td>
<td>Control OID - Criticality pairs for each control (Note, no value)</td>
</tr>
<tr>
<td>LDAP_SEARCH_FILTER</td>
<td>CHAR</td>
<td>256</td>
<td>5324</td>
<td>5579</td>
<td>Search filter attribute</td>
</tr>
<tr>
<td>LDAP_SEARCH_SCOPE</td>
<td>CHAR</td>
<td>12</td>
<td>5582</td>
<td>5593</td>
<td>Subtree search, base search</td>
</tr>
<tr>
<td>LDAP_SEARCH_ATTRS</td>
<td>CHAR</td>
<td>64</td>
<td>5596</td>
<td>5659</td>
<td>Attr requested to be returned on the search. One of these for each.</td>
</tr>
</tbody>
</table>
Appendix F. Guidelines for interoperability between non-z/OS TDS and z/OS TDS

This section contains information to consider when setting up a mixed platform environment where an LDAP directory is being replicated between non-z/OS TDS on Linux, Unix, or Windows platforms and z/OS TDS. This information also applies to migration of the schema and directory entries between these different platforms.

Schema considerations

1. Syntax and matching rules:
   TDS on non-z/OS platforms support more syntaxes and matching rules than z/OS TDS.
   Additional syntaxes and matching rules must be removed from the non-z/OS TDS schema before it can be used in z/OS TDS. 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 TDS. 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 non-z/OS TDS or z/OS TDS by publishing the schema (using ldapsearch –L) might not be acceptable input for a schema modification.
   a. When modifying the non-z/OS TDS 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 TDS schema, the entire LDIF can be processed in a single modify-replace operation, whether 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 TDS, the following schema modification replaces those schema elements and adds new attribute attr2 and object class objclass2:

```.ldif
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
}
```
For non-z/OS TDS, 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 )
)}

Import or export of directory entries

1. Exporting data from non-z/OS TDS to z/OS TDS:
   a. Non-z/OS TDS 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 TDS only supports some of these special entries. See [Enabling advanced replication] for more information about the special entries in z/OS TDS. Remove the other non-z/OS TDS special entries from the LDIF or use `db2ldif -s subtreeDN -x` to avoid unloading these suffixes.
   b. User passwords must be in clear text, SHA, or crypt. Other forms are not compatible with z/OS TDS. If using crypt, make certain to specify `pwCryptCompat off` in the z/OS TDS configuration file.
   c. z/OS TDS does not support the use of filtered ACLs in `aclEntry` attribute values (`ibm-filterAclEntry` attribute). You must remove these before importing to z/OS TDS.

2. Exporting data from z/OS TDS to non-z/OS TDS:
   a. For non-z/OS TDS, `aclEntry` and `entryOwner` attribute values must begin with the following format:
      "access-id:|group:|role:"
      This is not required for z/OS TDS, therefore, it might need to be added to these attribute values before importing to non-z/OS TDS. Always specify these on z/OS TDS to avoid this issue.
   b. User passwords must be in clear text, SHA, or crypt. Other forms are not compatible with non-z/OS TDS. If using crypt, make certain to specify `pwCryptCompat off` in the z/OS TDS configuration file. Use `ds2ldif -t` to unload passwords in the tagged format used by non-z/OS TDS.

Functional considerations

1. For non-z/OS TDS, deleting a person entry results in removing the DN of the entry from groups and ACLs. This is not done in z/OS TDS. Instead, applications need to do this themselves.
2. Similarly, non-z/OS TDS supports a control that allows deletion of all entries in a subtree. z/OS TDS does not support this control, therefore, applications need to do this themselves.
3. There are some capabilities that only non-z/OS TDS or only z/OS TDS supports. Restrict usage to capabilities supported on both platforms to facilitate replication of operations and migration of entries.
Appendix G. Accessibility

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully. The major accessibility features in z/OS enable users to:

- Use assistive technologies such as screen readers and screen magnifier software
- Operate specific or equivalent features using only the keyboard
- Customize display attributes such as color, contrast, and font size

Using assistive technologies

Assistive technology products, such as screen readers, function with the user interfaces found in z/OS. Consult the assistive technology documentation for specific information when using such products to access z/OS interfaces.

Keyboard navigation of the user interface

Users can access z/OS user interfaces using TSO/E or ISPF. Refer to z/OS TSO/E Primer and z/OS ISPF User’s Guide Vol I for information about accessing TSO/E and ISPF interfaces. These guides describe how to use TSO/E and ISPF, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.

z/OS information

z/OS information is accessible using screen readers with the BookServer/Library Server versions of z/OS books in the Internet library at: http://www.ibm.com/systems/z/os/zos/bkserv/
This information was developed for products and services offered in the USA.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information about 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
USA

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information, therefore, these changes are incorporated in new editions of the publication. IBM may make improvements or changes in the product(s) or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who want 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
Mail Station P300
245S South Road
Poughkeepsie, NY 12601-5400
USA
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 information 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.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Policy for unsupported hardware

Various z/OS elements, such as DFSMS™, HCD, JES2, JES3, and MVS, contain code that supports specific hardware servers or devices. In some cases, this device-related element support remains in the product even after the hardware devices pass their announced End of Service date. z/OS may continue to service element code; however, it will not provide service related to unsupported hardware devices. Software problems related to these devices will not be accepted for service, and current service activity will cease if a problem is determined to be associated with out-of-support devices. In such cases, fixes will not be issued.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at &zostm;.

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

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product or service names may be the trademarks or service marks of others.
Bibliography

This bibliography provides a list of publications that are useful when implementing the Tivoli directory server product.

IBM z/OS Security Server publications

- z/OS Migration, GA22-7499
- z/OS Security Server RACF Command Language Reference, SA22-7687
- z/OS Security Server RACF Callable Services, SA22-7691
- z/OS Integrated Security Services Network Authentication Service Administration, SC24-5926
- z/OS Integrated Security Services Network Authentication Service Programming, SC24-5927

IBM Tivoli Directory Server for z/OS

- IBM Tivoli Directory Server Client Programming for z/OS, SA23-2214
- IBM Tivoli Directory Server Plug-in Reference for z/OS, SA76-0148

IBM C/C++ language publications

- z/OS XL C/C++ Programming Guide, SC09-4765
- z/OS XL C/C++ Run-Time Library Reference, SA22-7821

IBM DB2 publications

- DB2 ODBC Guide and Reference, SC18-7423
- DB2 Application Programming and SQL Guide, SC18-7415
- DB2 Installation Guide, GC18-7418
- DB2 Command Reference, SC18-7416
- DB2 Messages and Codes, GC18-7422
- DB2 SQL Reference, SC18-7426
- DB2 Data Sharing: Planning and Administration, SC18-7417
- DB2 Utility Guide and Reference, SC18-7427
- DB2 ODBC Guide and Reference, SC18-7423

IBM z/OS Cryptographic Service publications

- z/OS Open Cryptographic Services Facility Application Programming, SC24-5899
- z/OS Cryptographic Services ICSF Administrator’s Guide, SA22-7521

Other IBM publications

- z/OS Communications Server: IP Configuration Guide, SC31-8775
- z/OS Communications Server: IP Configuration Reference, SC31-8776
- z/OS Program Directory, GI10-0670
- z/OS Cryptographic Services System SSL Programming, SC24-5901
- z/OS UNIX System Services Planning, GA22-7800
- z/OS Parallel Sysplex Overview, SA22-7661
- z/OS MVS Setting Up a Sysplex, SA22-7625
- z/OS SDSF Operation and Customization, SA22-7670
- z/OS DCE Command Reference, SC24-5909
- z/OS Planning for Installation, GA22-7504
- z/OS Introduction and Release Guide, GA22-7502
- z/OS DCE Application Development Guide: Directory Services, SC24-5906
- z/OS Licensed Program Specifications, GA22-7503
- z/OS Collection, SK3T-4269
- Policy Director Authorization Services for z/OS and OS/390 Customization and Use, SC24-6040
- ServerPac: Using the Installation Dialog, SA22-7815

© Copyright IBM Corp. 2009
Glossary

This glossary defines technical terms and abbreviations used in z/OS LDAP documentation. If you do not find the term you are looking for, refer to the index of the appropriate z/OS LDAP manual or view IBM Dictionary of Computing, available from www.ibm.com/ibm/terminology

This glossary includes terms and definitions from:
- Information Technology—Portable Operating System Interface (POSIX), from the POSIX series of standards for applications and user interfaces to open systems, copyrighted by the Institute of Electrical and Electronics Engineers (IEEE).
- American National Standard Dictionary for Information Systems, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies may be purchased from the American National Standards Institute, 11 West 42nd Street, New York, New York 10036. Definitions are identified by the symbol (A) after the definition.
- Information Technology Vocabulary, developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1.SC1).
- Open Software Foundation (OSF).

A

access control. Ensuring that the resources of a computer system can be accessed only by authorized users in authorized ways.

access control list (ACL). Data that controls access to a protected object. An ACL specifies the privilege attributes needed to access the object and the permissions that may be granted, to the protected object, to principals that possess such privilege attributes.

ACL. See access control list.

attribute. Information of a particular type concerning an object and appearing in an entry that describes the object in the directory information base (DIB). It denotes the attribute's type and a sequence of one or more attribute values, each accompanied by an integer denoting the value's syntax.

B

backend. A subsystem of the LDAP server which implements access to a persistent storage mechanism for information.

C

CDBM. A file-based backend that stores configuration information.

certificate. Used to prove your identity. A secure server must have a certificate and a public-private key pair. A certificate is issued and signed by a Certificate Authority (CA).

cipher. A method of transforming text in order to conceal its meaning.

CKDS. Cryptographic Key Data Set.

client. A computer or process that accesses the data, services, or resources of another computer or process on the network. Contrast with server.

configuration. The manner in which the hardware and software of an information processing system are organized and interconnected.

Cryptographic Key Data Set (CKDS). (1) A data set that contains the encrypting keys used by an installation. (2) In ICSF, a VSAM data set that contains all the cryptographic keys. Besides the encrypted key value, an entry in the cryptographic key data set contains information about the key.

cryptography. (1) The transformation of data to conceal its meaning. (2) In computer security, the principles, means, and methods for encrypting plaintext and decrypting ciphertext. (3) In ICSF, the use of cryptography is extended to include the generation and verification of MACs, the generation of MDCs and other one-way hashes, the generation and verification of PINs, and the generation and verification of digital signatures.

D

daeamon. A long-lived process that runs unattended to perform continuous or periodic system-wide functions.
such as network control. Some daemons are triggered automatically to perform their task; others operate periodically.

**Data Encryption Standard (DES).** In computer security, the National Institute of Standards and Technology (NIST) Data Encryption Standard, adopted by the U.S. government as Federal Information Processing Standard (FIPS) Publication 46, which allows only hardware implementations of the data encryption algorithm.

**data hierarchy.** A data structure consisting of sets and subsets such that every subset of a set is of lower rank than the data of the set.

**data model.** (1) A logical view of the organization of data in a database. (2) In a database, the user’s logical view of the data in contrast to the physically stored data, or storage structure. (3) A description of the organization of data in a manner that reflects information structure of an enterprise.

**database.** A collection of data with a given structure for accepting, storing, and providing, on demand, data for multiple users.

**Database 2 (DB2).** An IBM relational database management system.

**DB2.** Database 2.

**DES.** Data Encryption Standard (DES).

**directory.** (1) A logical unit for storing entries under one name (the directory name) in a CDS namespace. Each physical instance of a directory is called a replica. (2) A collection of open systems that cooperates to hold a logical database of information about a set of objects in the real world.

**directory schema.** The set of rules and constraints concerning directory information tree (DIT) structure, object class definitions, attribute types, and syntaxes that characterize the directory information base (DIB).

**directory service.** The directory service is a central repository for information about resources in a distributed system.

**distinguished name (DN).** One of the names of an object, formed from the sequence of RDNs of its object entry and each of its superior entries.

**DN.** Distinguished name.

**environment variable.** A variable included in the current software environment that is available to any called program that requests it.

**EXOP.** A backend that supports various LDAP extended operations. No entries are kept in this backend.

**G**

**GBDM.** A backend that manages change log entries created because of changes to the LDAP schema or to entries in other backends (including RACF entries). The change log entries can be kept in UNIX System Services files or in a DB2 database.

**ICSF.** Integrated Cryptographic Service Facility.

**Integrated Cryptographic Service Facility (ICSF).** A licensed program that runs under z/OS and provides access to the hardware cryptographic feature for programming applications. The combination of the hardware cryptographic feature and ICSF provides secure high-speed cryptographic services.

**J**

**JCL.** Job control language.

**Job control language (JCL).** A control language used to identify a job to an operating system and to describe the job’s requirements.

**K**

**key generator utility program (KGUP).** A program that processes control statements for generating and maintaining keys in the cryptographic key data set.

**KGUP.** Key generator utility program.

**L**


**LDBM.** A general purpose backend that stores its entries in UNIX System Services files.

**Lightweight Directory Access Protocol (LDAP).** A client/server protocol for accessing a directory service.

**M**

**master replica.** The first instance of a specific directory in the namespace. After copies of the directory have been made, a different replica can be designated as the master, but only one master replica of a directory can exist at a time.

**MD5.** Message Digest 5. A hash algorithm.
MKKF. Make Key File.

MKKF utility. A command-line utility used to create public/private key pairs and certificate requests, receive certificate requests into a key ring, and manage keys in a key ring.

N

NOID. Numeric object identifier

O

object class. An identified family of objects that share certain characteristics. An object class can be specific to one application or shared among a group of applications. An application interprets and uses an entry’s class-specific attributes based on the class of the object that the entry describes.

OCSF. Open Cryptographic Services Facility.

ODBC. Open database connectivity.

Open Cryptographic Services Facility (OCSF). A derivative of the IBM Keyworks technology which is an implementation of the Common Data Security Architecture (CDSA) for applications running in the UNIX System Services environment.

z/OS Cryptographic Services. A z/OS offering that supplies a set of interfaces for cryptographic functions.

P

private key. Used for the encryption of data. A secure server keeps its private key secret. A secure server sends clients its public key so they can encrypt data to the server. The server then decrypts the data with its private key.

public key. Used for the encryption of data. A secure server makes its public key widely available so that its clients can encrypt data to send to the server. The server then decrypts the data with its private key.

R

RACF. Resource Access Control Facility.

RDN. Relative distinguished name.

referral. An outcome that can be returned by a directory system agent that cannot perform an operation itself. The referral identifies one or more other directory system agents more able to perform the operation.

relative distinguished name (RDN). A component of a DN. It identifies an entry distinctly from any other entries which have the same parent.

replica. A directory in the CDS namespace. The first instance of a directory in the namespace is the master replica. See master replica.

replication. The making of a shadow of a database to be used by another node. Replication can improve availability and load-sharing.

Resource Access Control Facility (RACF). An IBM licensed program, that provides for access control by identifying and verifying the users to the system, authorizing access to protected resources, and logging the detected unauthorized access to protected resources.

S

SASL. Simple Authentication Security Layer.

schema. See directory schema.

SDBM. A backend that provides access to RACF data. All entries are kept in RACF.

Secure Sockets Layer (SSL) security. A facility used to protect LDAP access.

server. On a network, the computer that contains programs, data, or provides the facilities that other computers on the network can access. Contrast with client.


Simple Authentication Security Layer (SASL). Refers to a method of binding using authentication information outside the client and server.

SLAPD. A stand-alone LDAP daemon.

SPUFI. SQL Processor Using File Input.

SQL Processor Using File Input (SPUFI). A facility of the TSO attachment subcomponent that enables the DB2I user to run SQL statements without embedding them in an application program.

SQL. Structured Query Language.

SSL. Secure Sockets Layer.

Structured Query Language (SQL). A standardized language for defining and manipulating data in a relational database.

T

TDBM. A general purpose backend that stores its entries in a DB2 database.

TDS. Tivoli Directory Server.
thread. A single sequential flow of control within a process.

Tivoli Directory Server (TDS). Implementations of LDAP servers, clients, and utilities provided by IBM, supporting many IBM and non-IBM platforms.

Time Sharing Option (TSO). An operating system option that provides interactive time sharing from remote terminals.

Transport Layer Security (TLS). A security protocol that provides communication privacy over the Internet. The protocol allows client/server applications to communicate in a way that is designed to prevent eavesdropping, tampering, or message forgery. TLS is based upon SSL Version 3.0.

TSO. Time Sharing Option.

U

UCS Transformation Format (UTF). The LDAP Version 3 protocol specifies that data is passed between client and server in the UTF-8 character set.

UTF. UCS Transformation Format.

X

X.500. The CCITT/ISO standard for the open systems interconnection (OSI) application-layer directory. It allows users to register, store, search, and retrieve information about any objects or resources in a network or distributed system.
Index

Special characters
passwd ()
errno values returned by 270
/etc directory
using 41
/usr/lpp/ldap directory
LDAP server install directory 41
(file-based)
setting up GDBM 53
' (apostrophe) 224
> (greater than sign) 224
# (pound sign support in SDBM) 282
# (pound sign) 224
+ (plus sign) 224
= (equal sign) 224

Numerics
7-bit ASCII 72

A
abandon behavior 472
ABSTRACT object class type 232, 241
access
determining 347
access classes
attribute 343
determining 342
permissions 344
specifying for TDBM 232, 240
access control
See also Access Control List (ACL)
attributes 341
configuring for Kerberos 306
groups 352
LDAP server capability 7
using 341
using RACF 269
access control and ownership
modify DN 256
Access Control List (ACL)
aclEntry attribute 342
aclPropagate attribute 345
aclSource attribute 345
attribute classes 349
creating a group for 361
creating in TDBM 354
creating owner for entry 358
deleting in TDBM 357
deleting owner for entry 360
description 6, 341
entryOwner attribute 345
examples 351
filters 349
groups 352
information, retrieving 353

Access Control List (ACL) (continued)
modifying in TDBM 356
modifying owner for entry 359
override example 350
ownerPropagate attribute 346
ownerSource attribute 346
propagation 346, 350
requested attributes 349
searching 349
access protection 105, 109
accessibility 809
ACL (Access Control List)
See Access Control List (ACL) 6
ACL attributes
entry owner attributes 341
ACL restrictions group gathering 332
acl restrictions group membership 332
aclEntry attribute
description 342
aclEntry syntax 342
aclPropagate attribute
description 345
aclSource attribute
description 345
activity logging 147
adding, modifying, deleting group entries
examples 332
adminDN option 74, 118
administration
restricting access 6
administrator
configuring DN of 118
password, specifying 75
roles for config 27
adminPW option 74
advanced configuration options 32
Advanced replication 174
replication 377
Advanced replication configuration examples 399
Advanced replication error recovery example 434
Advanced replication features 383
Advanced replication maintenance mode 422
Advanced replication overview 380
Advanced replication related entries summary 400
Advanced replication related extended operations 426
Advanced replication terminology 377
AES encryption method 95, 98
alias
description 439
alias entry 439
aliases
LDAP server capability 8
aliasing on search performance
search performance 439
alternate server
specifying 76
analyzing
schema errors 246
CDBM configuration entries 110
CDBM performance considerations 494
certificate
  authenticating 103
  client 9
digital
  server 59
Certificate bind validation 178
change log
  additional required configuration 446
  multiserver considerations 451
  set up and using LDAP server for logging changes 451
  when changed are logged 447
change log entries 448
change log information root DSE entry 450
change log schema 447
change logging 445
  LDAP server capability 8
changed information xxii, xxiv
changelog
  addentry 777
  changeLogAddEntry 777
checklist for configuration file 72
CICS (Customer Information Control System)
  updating related attributes 288
classes, access
  attribute 343
  determining 342
  permissions 344
  specifying for LDBM 232
clear text password 63, 96, 185
CLI (Call Level Interface) 16
  CLI initialization file name 39
Client connections 152
  client, LDAP
    See LDAP client 467
cn=configuration 110
cn=Log Management,cn=Configuration 112
cn=Replication,cn=configuration 110
cn=Replication,cn=Log Management,cn=Configuration 112
code page IBM-1047 69
command-line parameters
  for LDAP server 129
commmands
  cp 46
  commThreads option 109
complex modify DN replication 364
Component Broker
  configuring for 13
concurrent database instances 6, 7
configuration
  specifying value for distinguished name 71
  specifying value for filename 71
configuration file 63
  administrator DN, specifying 119
  alternate 114
  copying 46
  creating 69
  data set versions of 133
configuration file (continued)
  default referral 457
  deprecated options 109
ds.conf 69
  format 69
  Kerberos authentication 304
  locating 69
  master server 374
  options 75
  options checklist 72
  password, specifying 119
  scenarios 120
  setting SSL keywords 373
  specifying as data set name 132
  specifying as DD name 132
  using to configure 114
configuration file options
  aclSourceCacheSize 75
  adminDN 75
  adminPW 75
  allowAnonymousBinds 76
  altServer 76
  armName 76
  attrOverflowCount 77
  attrOverflowSize 77
  audit 77
  changeLogging 78
  changeLoggingParticipant 78
  changeLogMaxAge 78
  changeLogMaxEntries 79
  commitCheckpointEntries 79
  commitCheckpointTOD 79
  commThreads 79
  database 80
  databaseDirectory 80
db2StartUpRetryInterval 81
db2StartUpRetryLimit 81
db2Terminate 81
dbuserid 80
digestRealm 82
dnCacheSize 82
dnToEidCacheSize 82
dsnaiomi 82
enableResources 83
entryCacheSize 83
entryOwnerCacheSize 83
extendedGroupSearching 83
fileTerminate 84
filterCacheBypassLimit 84
filterCacheSize 84
idleConnectionTimeout 85
include 85
krbIdentityMap 85
krbKeytab 85
krbLDAPAdmin 86
listen 86
logfile 88
masterServer 89
masterServerDN 89
masterServerPW 90
maxConnections 90
Index 821
configuration file options (continued)
  multiserver 90
  nativeAuthSubtree 91
  nativeUpdateAllowed 91
  operationsMonitor 91
  operationsMonitorSize 92
  pcIdleConnectionTimeout 92
  pcThreads 92
  peerServerDN 92
  persistentSearch 93
  plugin 94
  port 94
  pwCryptCompat 95
  pwEncryption 95
  pwSearchOutput 96
  readOnly 96
  referral 97
  schemaPath 97
  schemaReplaceByValue 97
  secretEncryption 98
  securePort 98
  security 99
  securityLabel 99
  sendV3stringsoverV2as 99
  serverCompatLevel 100
  serverEtherAddr 100
  serverKrbPrinc 101
  serverName 101
  serverSysplexGroup 102
  sizeLimit 102
  srvStartUpError 103
  sslAuth 103
  sslCertificate 103
  sslCipherSpecs 103
  sslKeyRingFile 104
  sslKeyRingFilePW 104
  sslKeyRingPWStashFile 105
  sslMapCertificate 105
  suffix 106
  supportKrb5 106
  tcpTerminate 106
  timeLimit 107
  useAdvancedReplication 107
  useNativeAuth 108
  validateincomingV2strings 108
  wlmExcept 108

configuration utility 21

configuring (continued)
  roadmap for 35
  running with SDBM 51
  scenarios 120
  SDBM backend 35, 122, 123
  SLAPD 114
  SSL 37, 121
  TDBM backend 35, 121, 123
  using dsconfig utility 21
  WebSphere Application Server 13
  configuring DB2-based GDBM backend
  DB2-based GDBM backend 446
  configuring file-based GDBM backend
  file-based GDBM backend 446
  Configuring for user password encryption 65
  configuring GDBM backend
  GDBM backend 445
  Configuring LDAP with WLM examples 490
  conflict resolution
    peer to peer 372
  connection
    group 287
  connections
    specifying number of 90
    specifying timeout 85
  Consumer server entries 395, 401
  Control of access to RACF data
    access to RACF data 282
  Control replication 779
  Control replication error log 780
  Control replication queue 781
  Control values in ds2ldif LDIF output file 175
  control, program
    See program control 44
  controls, server
    See server controls 9
  conventions used in book xix
  converting
    See migrating 155
  copying
    configuration files 46
    files into data sets 133
  Copying an LDBM backend
    LDBM backend 52
  Copying database
    TDBM database 50
  cp command 46
  CRAM-MD5 and DIGEST-MD5 323
  configuration option 324
    TDBM, LDBM, or CDBM backend 323
  CRAM-MD5 and DIGEST-MD5 authentication 9
  CRAM-MD5 support 471
  creating
    ACL 354
    DB2 CLI initialization file 18
    ds.conf 69
    group for ACL 361
    owner for entry 358
    referral entries 455
    SYSTEM member 199
    versions of environment variable file 133
ds.conf
creating 69
format 69
locating 69
See configuration file 70
ds.db2.profile file 32
ds.envvars file 219
ds.profile file 24
ds.racf.profile file 32
ds2ldif utility
description 188
replicating 369
running from JCL 182
running in shell 181
running in shell TDBM backend
running utilities for 181
running in TSO 182
dconfig utility
advanced options 32
capabilities 22
capability 8
configuration confirmation 31
input files 24
overview 21
restrictions 22
steps for configuring with 28
using 23
DSE, root
See root DSE 9
DSNAOINI environment variable 83
DSNAOINI file 18, 133
dssrv user ID 129
DSTDBMDB SPUFI file 743
dynamic debugging
using 138
dynamic groups 329
dynamic link libraries (DLL)
using in startup 129
dynamic schema
TDBM LDBM 227
using TDBM backend 9
dynamic workload management
configuring LDAP server for 8
entries (continued)
creating for referrals 455
data model 223
defining 223
description 4
entryOwner attribute 345
identifying 223
loading 7
loading from TDBM 188
loading into TDBM 196
loading large number of 196
ownerPropagate attribute 346
ownerSource attribute 346
permissions 344
protecting 341
referral 456
replica 365
replica, adding in TDBM orLDBM 367
TDBM directory schema 227
entryOwner attribute
description 345
environment variable file
data set versions of 133
environment variables
LANG 219
LDAP_DS_ENVVARS_FILE 181
NLSPATH 219
setting for utilities 181
Environment variables
variables used by the LDAP server 135
environments
operating, Kerberos 309
equal sign 224
errno values returned by _passwd() 270
error codes 472
errors
schema, analyzing 246
escape mechanism for UTF-8 characters 72
directory hierarchy
using 41
example
CRAM-MD5 and DIGEST-MD5 324
elements
schema, defining new 242
enabling
SSL 59
Enabling advanced replication 384
encryption
configuring for 63
for communication channel 59
installing ICSF for 19
encryption, password
See password encryption 95
entries
access allowed for 342
aclSource attribute 345
arranging 4

E

824  z/OS V1R11.0 IBM Tivoli Directory Server Administration and Use for z/OS
examples (continued)
files shipped with LDAP server 46
LDAP URL 87
modifying user in RACF 297
object class hierarchy 233
overrides 350
partitioning DB2 tables for TDBM
   - EID value 49
permissions 348
propagation 350
referral entries 457
referrals, distributing namespace 460
refreshing the raclist for the facility class 300
removing user from group in RACF 300
removing user from RACF 300
replica entry definition 367
samples 46
schema entry 228
searching for user information in RACF 298
searching for user’s connection to group 299
setting up Kerberos directory 307
setting up native authentication 318
SPUFI script (TDBM) 743, 755
using ref attribute 455
EXOP backend
   - configuring 37
   - configuring, example 127
   - GetDnForUserId 327, 782
   - GetPrivileges 327, 783
   - PC callable support mode 115, 118
   - section in ds.conf 70
   - using 327
   - using for Policy Director 8
extended group membership searching 9, 84
extended operations
   - messages 669
extended operations backend
   - See EXOP backend 8
extensibleObject object class 241
external bind, SASL 9
external security manager 41

F
File system 152
files
   - configuration, global options 75
   - copying configuration 46
   - DB2 CLI initialization 18
   - ds.conf 70
   - ds.db2.profile file 32
   - ds.profile 24
   - ds.racf.profile file 32
   - DSTDBMDB SPUFI 743
   - profile 21
   - SPUFI 743
   - TDBMMGRT SPUFI 755
filters
   - searching 349
Finalizing setup
   - setup of LDAP backends 134
finding
   - subschemaSubentry DN 247
first time installing 14
formats
   - ds.conf 70
   - GIF 4
   - JPEG 4
Forwarding (cascading) replication 381
front end
   - for X.500 6

G
Gateway replication 382
Gateway topology
   - Creating gateway topology 415
   - gathering group memberships 84
GDBM (DB2-based) backend-specific section
   - section in ds.conf 69
GDBM (file-based) backend-specific section
   - section in ds.conf 70
GDBM backend
   - configuring 36
   - configuring SDBM and GDBM 122
   - initializing ACLs 347
   - setting up for 53
   - verifying 134
GDBM change log performance considerations
   - changelog performance 506
GDBM change logging for RACF resources 173
GDBM messages
   - advanced replication 669
GDS (Global Directory Service)
   - See Global Directory Service (GDS) 6
general performance considerations
   - server performance 487
Generalized Time syntax 246
GetDnForUserId 327, 782
GetPrivileges 327, 783
GIF format 4
global configuration file options 75
Global Directory Service (GDS) 6
global section in ds.conf 69
glossary of terms 815
group examples 332
group membership 331, 765
groups
   - access control 352
   - connecting to in RACF 287
   - creating for ACL 361
   - extended, membership searching 9, 84
   - universal, searching 295
GSS API bind 303, 305
Guidelines for interoperability
   - interoperability
      - non-z/OS TDS and z/OS TDS 805

H
HFS (Hierarchical File System)
   - See Hierarchical File System (HFS) 6
Hierarchical File System (HFS)
  naming the LDAP server 6
  hierarchical tree
  defining 4
hierarchy
  directory 223
  directory, example of 4
  example, object class 233
  referrals 456

I
  IBM attribute types
    description 232
    format 240
    usage 240
  IBM-1047 character set 75
  ibm-entryuuid
    replication 364
    ibm-entryuuid.attribute 10
  IBMAttributeTypes schema attribute 232, 241
  IBMLdapProxyControl 765
  IBMSchemaReplaceByValue 767
  IBMSchemaReplaceByValueControl 767
  ICSF (Integrated Cryptographic Service Facility) 19
  identity mapping
    Kerberos 305
    identity, Kerberos 43
  Improved DB2 partition support
    DB2 partition 179
  indexes, create DB2 46, 755
  information
    arranging 4
    protecting 6
    referencing 5
  inheritance
    default 346
  Initial validation of compatible server versions
    consumer and replica servers 267
  initialization file, DB2 CLI 18
  initializing
    native authentication 311
    replica directory 369
  Initializing ACLs with schema entry
    Initializing ACLs 347
  Installation Verification Procedure (IVP)
    for configuring LDAP server 31
  installing
    DB2 16
    first time 14
    for CLI 16
    for ODBC 16
    LDAP server 13
    migrating 155
    Policy Director 19
    RACF 18
    related products 15
    roadmap 13
    System SSL 19

Installing a z/OS UNIX System Services file system for
  LDBM, GDBM (file-based), and CDBM backends 19
Installing and setting up WLM (Workload
  Management) 15
Integrated Cryptographic Service Facility (ICSF)
  installing for encryption 19
  interactions, backend variables 38
  international characters
    retrieving 9
    storing 9
  IVP (Installation Verification Procedure) 31

J
  Japanese messages 219
  JCL (Job Control Language)
    See Job Control Language (JCL) 129
  JOB card, modifying 182
  Job Control Language (JCL)
    for running SLAPD as started task 129
    generated by config 21
    running DB2 server utilities from 182
  JPEG format 4

K
  KDC (Key Distribution Center) 303
  Kerberos
    authentication
      description 303
      bind capability 9
      configuring access control 306
      configuring administrator for 120
      identity
        defining 43
        identity mapping 305
        identity mapping option 85
      installing 20
      operating environments 309
      setting up for 303
      setup up directory example 307
      specifying administrator identity 86
      specifying attribute in SDBM 287
      specifying keytab 86
      specifying participation in 106
      updating directory schema 304
      using dsconfig with 33
    key database file
      specifying 109
      specifying for server 104
    Key Distribution Center (KDC) 303
    keyboard 809
    keytab file
      generating for server 303
      krbLDAPAdmin option 120

L
  IA5 character set 471
  LANG environment variable 219
  LANG parameter 41
Index 827

LDAP 3
LDAP (Lightweight Directory Access Protocol)

See Lightweight Directory Access Protocol (LDAP)

4

LDAP administrator

role for config 27

LDAP client

authenticateOnly server control 765
cipher specifications 104
considerations 467
using in LDAP 6
UTF-8 data 471

LDAP client requests

listening for 86

LDAP configuration utility 21

LDAP Data Interchange Format (LDIF)

loading entries from TDBM 188
loading entries into TDBM 196

LDAP directory

protecting information in 6

LDAP schema attributes 235

ldap server

threads 487

LDAP server 220

access control 341
administrator distinguished name (DN) 118
alternate server 76
attribute types supported 239
authenticateOnly server control 765
capabilities 6
changing replica to master 371
command-line options 129
configuration options 69
configuring 114
configuring for multi-server mode 116
configuring with dsconf utility 21
data model 223
DB2 server location 101
debugging facility 138
default directory 41
defining started task for 129
defining user ID 41
enabling SSL for 59
equivalent server 76
extended group membership searching 9, 84
IBM attribute types supported 240
IBM Tivoli Directory Server for z/OS 3
installing 13, 15
LDAP syntaxes supported 235
master and replica 373
matching rules supported 237
messages 511
migrating 155
model for 4
multi-server mode 114
multiple single-server mode LDAP servers 114
name, installation 41
naming 6

LDAP server (continued)

NLS 219
object classes supported 241
planning for 13
preparing 41
RACF 269
replica 369
replication 363
requirements for user ID 41
retrieving Policy Director data 8
running as started task 129
sample 14
sample files 46
SDBM backend 269
setting up LDBM 51
Setting up multiple LDAP servers with DB2-based
backends 116
setting up SDBM 50
single-server mode 114
started task, defining 129
starting from console 129
starting in SDSF 129
starting up 16
stopping from console 132
stopping in SDSF 132
TDBM collation 220
user ID 39
using 6
using for user ID 41
verifying 133
Version 3 protocol 8

LDAP server abnormal termination 152

LDAP syntaxes

description 233
format 235
supported, general use 235
supported, server use 236
usage 235

LDAP URL

starting LDAP server 130
LDAP URL format

specifying for listen 86
LDAP_DEBUG environment variable 131
ldap_ssl_client_init API

using for SASL bind 63
ldap_ssl_init API

using for SASL bind 63
ldapadd utility

loading entries into TDBM 200
ldapdiff

Comparing directories 10
ldapdiff utility

description 205
ldapexop

Extended operations utility 11
ldapexop utility 182
description 212
LDAPResult construct 472
ldapsearch utility

using to verify LDAP server 133
LDAPSRV user ID 41
LDBM and TDBM backend
  object class supported 241
LDBM backend 124, 311
  configuring 36, 37
  messages 652
  section in ds.conf 70
  setting up for 51
LDIF (LDAP Data Interchange Format) 196
ldif2db utility
  using with replica server 364
ldif2ds utility
  allocating datasets for 198
  description 196
  entering ACLs 341
  loading adminDN entry 119
  recovery 202
  running from JCL 182
  running in shell 181
  running in TSO 182
less than sign 224
levels, debug 130
Lightweight Directory Access Protocol (LDAP)
  description 4
  how it works 6
  schema publication 227
limitations, referrals 458
listen option 109, 118
loading
  entries 7
  entries from TDBM 188
  entries into TDBM 196
  large number of entries 196
locating
  ds.conf 69
  subschemaSubentry DN 247
M
mail attribute 4
mainframe
  education xx
maintenance mode
  peer to peer 368
manageDsaiT 459
manageDsaiT server control 766, 767, 768
managing
  replication 426
  managing password
    managing administrator DN 119
mapping
  identity, Kerberos 305
  Kerberos default 305
  LDAP attributes 271
mask for debug level 130
master
  backup of 363
  changing replica to 371
  communicating with replica 373
  database, description 363
  server 363
  server DN 89
master (continued)
  server password 90
  server, setting up 373
  server, specifying 89
Master-replica replication 380
matching rules
  description 233
  EQUALITY values 230
  EQUALITY, defaults 230
  format 237
  ORDERING values 230
  SUBSTR values 230
  supported 237
  usage 237
  maxThreads option 79, 109
  MAY attribute type 232, 241
  MD5 encryption method 95
mdofy DN
  access control 256
  membership 84
  extended group, searching 9
messages
  LDAP server 511
  LDBM 652
  started task 133
migrating
  from previous LDAP releases 155
minimum schema for TDBM 721
modes
  choosing multiserver 91
  multi-server 114
  multi-server, operating in 116
  PC callable support 115, 118
  single-server mode 114
  single-server mode LDAP servers 114
  single-server, operating in 115
modify
  updating ACLs 354
modify DN
  access control changes 258
  basic replication synchronization 268
  complex replication 364
  considerations 255
  considerations in the use of 253
  entries for rename 254
  operation syntax 249
  operations and replication 266
  relocating an entry 257
  scenario constraints 260
  SDBM schema 282
  suffix DNs 260
Modify DN
  validation 267
monitor search examples 502
monitor support 471
Monitoring LDAP server resources
  server resources 151
monitoring performance 497
multi-server mode
  configuring for 116
  running in 114
multi-valued ref attribute 455
multiple databases
  concurrent 6, 7
multiple directories
  replication of 363
multiple LDAP servers
  setting up 116
multiple socket ports 10
MUST attribute type 232, 241
mutual authentication 9

N
namespace
  entries, RACF 280
  hierarchy diagram for RACF 281
National Language Support (NLS)
  setting variables for 219
native authentication
capability 8
  defining participation 312
description 311
  example of setting up 318
  initializing 311
  installing RACF for 18
  operating modes 312
  password changes 91
  specifying DN 91
  updating schema 311
  using with Web servers 321
native operations
  running 314
  nativeAuthSubtree option 312
  nativeUpdateAllowed option 312
nested groups 331
Network Authentication and Privacy Service 303
Network communications 151
new information xxi, xxiii, xxv
NLSPATH environment variable 219
normal access class 343
Notices 811
installing
  Kerberos 20
numeric object identifier (OID) 242

O
object class
  adding for LDBM and TDBM 241
definitions, adding for LDBM and TDBM 241
description 223, 232
  extensibleObject 241
  format for LDBM and TDBM 241
  hierarchy 233
  Kerberos 304
  LDBM and TDBM usage 241
  referral 455
  replicaObject 365
  TDBM directory schema
    LDBM directory schema 228
  object class attribute
description 5
  object class definitions
    adding for LDBM and TDBM 241
    specifying for LDBM and TDBM 241
  object identifiers
    oid
      supported and enabled capabilities 468
  objectClass attribute 228
  objects
    protecting 341
    ODBC (Open Database Connectivity) 16
    OGET command 133
    OID (numeric object identifier) 242
    One-way encryption formats 63
    Open Database Connectivity (ODBC)
      setting up for 16
    operating environments
      Kerberos 309
    Operating in multiple single-server mode
      multiple single-server mode 115
    operating modes
      native authentication 312
    operations
      defining 6
      operationsMonitorSize
        monitor 92
    operator console
      starting SLAPD from 129
    Optional Products 16
    options
      checklist 72
      configuration file 75
    organization of book xix
    organizing
      information 4
    out-of-sync conditions 376
    overhead, reducing 7
    override, ACL example 350
    owner
      creating for entry 358
      deleting for entry 360
      modifying for entry 359
      ownerPropagate attribute 346
      ownerSource attribute 346
      ownerPropagate attribute
description 346
      ownerSource attribute
description 346

P
Parallel Sysplex
  using 114
Partial replication 383
Participation in multilevel security
  multilevel security 11
partitioned data set (PDSE)
  DLLs 129
password
  administrator, specifying 75
password (continued)
changing 91
master server 90
replication key database 109
SSL key database file 105
storing in stash file 105
password encryption
configuring for 37, 38
db2pwden utility 185
replication 364
password or password phrase
RACF
changing using SDBM 296
password or password phrases
native modify 314
password phrase 296
Password phrase support 177
passwords in change log entries 450
PC (program call)
See program call (PC) 92
PDSE (partitioned data set)
See partitioned data set (PDSE) 129
peer replica
Adding to existing server 372
peer server
Downgrade to read-only replica 373
peer to peer
peer to peer replication 371
peer-to-peer replication 381
Peer-to-peer replication 421
Peer-to-peer replication 381
performance improvements 8
performance information
capturing 140
Performance tuning
tuning 487
performance, LDAP server 7
Periodic validation of compatible server versions in
basic replication replicas 267
permissions
access 342
attribute access classes 344
determining 347
entry 344
examples 348
Persistent 768
Persistent search 10
PersistentSearch 768
PKCS #11 tokens 183
plan name 38
PLANNNAME value 38
planning
LDAP server setup 13
plug-in extension 37
Plug-in extensions to the LDAP server
Plug-in extensions 179
Plug-in support 10
plus sign 224
Policy Director
data, accessing 327
data, retrieving for LDAP 8
Policy Director (continued)
retrieve DNs 782
retrieving data 327, 783
setting up to support 19
setting up to use 54
port
multiple socket 10
numbers, default 130
TCP/IP for SSL 98
TCP/IP, specifying 94
port option 109
pound sign (#) support in SDBM 282
pre-configuration set up 15
preparing
for backend interactions 38
LDAP server 41
Preparing WLM, backends, sysplex, SSL/TLS, and
encryption 45
procedure
JCL to run SLAPD 129
profile files
used with config 21
program call (PC)
callable support mode 115, 118
initializing threads for 92
program control
for LDAP server 44
propagation, ACL
description 350
example 350
indicating, flag for 345
protecting
evironment for LDAP server 44
information 6
information using ACLs 341
protecting access 105, 109
Protecting replication topology entries 421
protection, scope of
attribute privileges 342
determining 342
protocol
directory 4
Version 3 8
PROXY segment 42
pseudo DN 343
publication, schema 227
publications, related 813
pwEncryption option 185, 364
Q
query command 140
querying
root DSE 467
schema 228
subschemaSubentry 229
querying group membership examples 335
Quiesce or unquiesce context 784

830 z/OS V1R11.0 IBM Tivoli Directory Server Administration and Use for z/OS
R

RACF (Resource Access Control Facility)
- See Resource Access Control Facility (RACF) 41
- RACF class and general resource support 173
- RACF custom fields support 177
- RACF key rings
  - using 183
  - racfAttributes
  - racfConnectAttributes 280
  - RACFFIELD 240
- RDN (relative distinguished name)
  - See relative distinguished name (RDN) 5
  - read-only replica
    - Upgrading to be a peer replica of the master server 372
- reason codes 472
  - recovering
    - from out-of-sync conditions 376
    - ldif2ds 202
  - Recovering from advanced replication errors 432
- ref attribute 455
- references
  - setup, recommended 456
  - referencing information 5
- referrals
  - default 97
  - default, defining 457
  - description 455
  - example of distributing namespace 460
  - LDAP server capability 7
  - limitations with version 2 458
  - manageDsaIT server control 766, 767, 768
  - processing 457
  - replication 370
  - specifying 97
  - suppressing 766, 767, 768
  - using without dynamic WLM 116
  - Version 2 protocol 458
  - Version 3 protocol 458

- Refresh DB2RUNSTATS
  - DB2 RUNSTATS 153

- relational database
  - loading entries into TDBM 196

- relative distinguished name (RDN)
  - description 5, 223

- Release 10 (z/OS LDAP)
  - new and changed functions 176
  - update summary 176

- Release 11 (z/OS LDAP)
  - new and changed functions 173
  - update summary 173

- relocating an entry with DN realignment 257

- Replacing individual schema values
  - Replacing values 244

- replica
  - changing to master 371
  - communicating with master 373
  - entry 374
  - master server 89
  - setting up 373

- Replica groups 385

- Replica server with SSL/TLS enablement 425
- Replica subentries 386
- replicateOperational 771
- replicateOperationalAttributes 771
- replicating
  - server 365

- Replicating server with SSL/TLS enablement 425

- replication
  - benefits 363
  - command line tasks 426
  - configuration information 428
  - database, description 363
  - description 363
  - entries 365
  - entries, adding in TDBM orLDBM 367
  - ibm-entryuuid 364
  - LDAP server capability 7
  - partial replication 423
  - password encryption 364
  - server 369
  - server, configuring 369
  - specifying key database file for 109
  - TDBM 115

- troubleshooting 375

- Replication agreements 387

- Replication conflict resolution 383

- replication consumer 207

- Replication contexts 385

- Replication scheduling 383

- Replication schema updates 421

- replication supplier 206

- Replication topology 379, 785

- Replication topology hints and tips 420

- replKeyRingFile option 109

- replKeyRingPW option 109

- Request for Comments (RFC)
  - supported by z/OS LDAP 11

- requested attributes 349

- Required products 15

- Resource Access Control Facility (RACF)
  - accessing information in 50, 51, 53
  - administrator DN 120
  - changing password or password phrase with SDBM 296
  - command to create LDAPSERV 42
  - commands for defining started task 129
  - configuring LDAP server for 7
  - connection to group 287
  - distinguished names 224
  - input file for config 32
  - installing for native authentication 18
  - installing for SDBM 18
  - LDAP access to 269
  - namespace entries 280
  - password 120
  - PROXY segment 42
  - RACF fields 271
  - RACF fixed fields 271
  - restriction on amount of output 295
  - Subsystem function 19
Resource Access Control Facility (RACF) (continued)
universal groups
searching 295
using 41
retrieving racf
racf password envelope
user password or password phrase
envelopes 296
return codes 472
RFC (Request for Comments)
See Request for Comments (RFC) 11
roadmap
configuring LDAP server 35
installing and running LDAP server 13
roles
configuration utility 27
root DSE
searching 467
support of 9
RRSAF (Resource Recovery Services attachment facility) 18
rules, matching
See matching rules 237
running
DB2 server utilities 181, 182
LDAP server as started task 129
LDAP server using data sets 133
LDAP server using DB2 16
LDAP tools with SDBM 282
native operations 314
roadmap 13
Running and using the LDAP server utilities
Running and using server utilities 181
running utilities for 181, 182

S
SAF (Security Authorization Facility) 19
samples
ds.envvars file 219
DSNAOINI file 18
LDAP server 14
object class hierarchy 233
schema entry 228
SPUFI script (TDBM) 743, 755
SASL CRAM-MD5 and DIGEST-MD5 9
SASL external bind 9
SASL GSS API Kerberos bind 9
scenarios
configuration 120
Schedule entries 391
schema
dynamic 9
modifying 196
updating for Kerberos 304
updating for native authentication 311
schema (LDBM and TDBM)
defining new elements 242
object classes 241
schema (LDBM)
updating LDBM 243
schema (TDBM)
attribute syntax
schema (LDBM) 234
attribute types
schema (LDBM) 239
directory 227
entry, displaying 247
errors, analyzing 246
IBM attribute types
schema (LDBM) 240
LDAP attributes
schema (LDBM) 235
LDAP syntaxes 235
matching rules
schema (LDBM) 237
minimum schema 721
publication 227
replica entry 365
retrieving TDBM 247
sample entry
schema (LDBM) 228
searching for schema entry 247
update 227
updating TDBM 243
scope of protection
attribute privileges 342
determining 342
script
modifying for TDBM 47
scripts
running for TDBM 48
SDBM authorization 269
SDBM backend
changing password in RACF 296
configuration utility 22
configuring for 35
configuring with GDBM 122
connection to group 287
deleting attributes 301
dconfig utility 22
implementing 7, 269
installing RACF for 18
mapping, Kerberos 306
namespace hierarchy 281
pound sign (#) support 282
preparing for 38
RACF-style DNs 224
running LDAP tools with 282
schema publication 227
section in ds.conf 69
setting BINDPW in PROXY segment 42
setting up for 50
setting up user ID 42
support of PROXY segment 42
using for authentication 341
using LDAP operation utilities with 297
verifying 134
SDBM performance considerations 507
SDBM schema
modify DN 282
SDBM search capabilities
  SDBM search 289
SDSF
  file 132
  starting LDAP server in 129
  search 10
searching
  across multiple servers 455
  anonymous 350
  directories 6
  entire RACF database 295
  permissions required 349
  replication 363
  root DSE 467
  schema entry 247
  using attributes 349
searching the change log 449
Secure Sockets Layer (SSL)
  certificate 103
  certificate authentication 103
  cipher specifications 104
  configure LDAP server using 8
  configuring 37
  enablement 373
  enabling 59
  key database file
    protecting access to 105, 109
    specifying 109
    specifying for server 104
  password for key database file 105
  preparing for 19
  replication 373
  setting up options for 58
  specifying in configuration file 99
  stash file 105
  TCP/IP port for 98
securePort option 109
SecureWay Security Server Network Authentication and Privacy Service for z/OS 20
security
  options, setting up 58
  specifying type of 99
  security administrator
    role for config 27
  Security Authorization Facility (SAF)
    using for Policy Director support 19
security manager, external 41
security option 59, 109
segment, PROXY 42
sendV3stringsoverV2as option 471
sensitive access class 343
server
  associating with referrals 456
  certificate 59
  master 373
  master, problems 375
  master, specifying 89
  parent 456
  pointing to others 456
  referrals 455
  replica 369, 373
server (continued)
  retrieving ACL information 353
  using in LDAP 6
server and bulkload messages
  messages 511
Server backends and plug-ins during startup 151
server configuration
  peer to peer 372
server configuration file
  data set 182
server controls
  authenticateOnly 84, 765
  Do Not Replicate 765
  LDAP server capability 9
  manageDsaIT 766, 767, 768
  No Replication Conflict Resolution 768
  PersistentSearch 768
  Refresh Entry 771
  replicateOperationalAttributes 771
  Replication Supplier ID Bind 772
  Server Administration 773
server ID 400
server messages
  ldif2ds messages 511
server name 39
server replication
  modify DN 268
server utilities, DB2 181, 182
Setting AUTOCOMMIT 18
setting time zone
  time zone 33, 39
setting up
  buffer pools 17
  DB2 16
  for CLI 16
  for ODBC 16
  Kerberos directory 307
  LDAP server using roadmap 13
  multiple LDAP servers 116
  native authentication example 318
  related products 15
  TEMP datasets 17
  TEMP space 17
Setting up for Policy Director extended operations
  setting up for 54
Setting up for WLM (workload management) 45
setup for sysplex 43
SHA encryption method 95
shortcut keys 809
shutting down
  LDAP server 132
single-server mode
  configuring for 115
  multiple 114
  replicating in 363
  running in 114
Size limitations 147
SMF records
  Record Type 83, subtype 3 791
socket ports, multiple 10
space, white 224
spaces, blank 224
Special usage of racfAttributes and racfConnectAttributes
racfAttributes and racfConnectAttributes 280
SPUFI (SQL Processor Using File Input) facility
   See SQL Processor Using File Input (SPUFI) facility 46
SQL Processor Using File Input (SPUFI) facility
   creating TDBM database with 46
   DSTDBMDB SPUFI script (TDBM) 743
   TDBMMGRRT SPUFI script (TDBM) 755
SSL/TLS
   creating key database, RACF key ring, or PKCS #11 token 57
   enabling 58
   LDAP utilities 182
   obtaining a certificate 58
   protected communications 56
   setting up an LDAP client 62
   setting up for 56
SSL/TLS and advanced replication 425
sslAuth option 63
sslKeyRingFile option 109
sslKeyRingPWStashFile option 109
stand-alone LDAP daemon
   See LDAP server 4
start TLS 787
started task
   changing debug setting 138
   defining 129
   defining user ID for 41
   messages 133
   running LDAP server as 129
starting
   LDAP server in SDSF 129
   loading DLLs 129
   SLAPD from console 129
stash file 105
static
   dynamic
      nested groups 329
   static groups 329
steps for installing and running LDAP 13
stopping
   LDAP server from console 132
   LDAP server in SDSF 132
storing information 4
STRUCTURAL object class type 232, 241
subject
   determining rights for 342
   submit command 182
subschemaSubentry attribute 229
subschemaSubentry DN 247
suffix
   option 119
summary of changes xxii, xxiii, xxv
Supplier server entries 385, 400
Suppliers and consumers 399
Support for tagged, encrypted passwords
   encrypted passwords 178
   supported extended operations 775
Symmetric encryption keys 64
synchronizing directories 363
syntax
   DN 224
   EQUALITY matching rules 230
   ORDERING matching rules 230
   schema attribute 234
   SUBSTR matching rules 230
sysplex
   See Parallel Sysplex 114
system administrator
   role for config 27
SYSTEM member, creating 199
system programming
   role for config 27

T
   table spaces for TDBM
      creating 46
      partitioning 48
   task, started
      See started task 129
TCP/IP (Transaction Control Protocol/Internet Protocol)
      See Transaction Control Protocol/Internet Protocol (TCP/IP) 4
TDBM and GDBM (DB2-based) backend
   installing and setting up 16
TDBM backend 63
   access control
      pseudo DN 343
   attribute types supported
      LDBM backend 239
   buffer pools 17
   collation 220
   configuration utility 22
   configuring 35
   configuring administrator DN 119
   configuring password 119
   configuring sample server with 46
   configuring with SDBM 123
   configuring with SSL and password encryption 121
   creating DB2 databases 46
   creating table spaces 46
   Default ACLs 346
   directory schema 227
ds2ldif utility 188
dsconfig utility 22
dynamic schema 9, 227
IBM attribute types supported
   LDBM backend 240
   LDAP syntaxes supported
      schema (LDBM) 235
   ldif2ds utility 196
   matching rules supported
      LDBM backend 237
   messages 635
   native authentication 311
   partitioning DB2 tables 48
   preparing for 38
   running utilities for 182
Version 2 protocol
  output data format  99
  referrals  458
  referrals, limitations  458
  validating data  108
Version 3 protocol
  LDAP support of  8
  referrals  458
  TDBM schema  227
  UTF-8 characters  219, 471

W
Waiting for DB2 to start  180
waitingThreads option  79, 109
Web servers
  using native authentication with  321
WebSphere Application Server
  configuring for  13
white space  224

X
X.500
  data model  223
  description  6
X.509 standard
  digital certificate  59

Z
z/OS Basic Skills information center  xx
z/OS Workload Manager  174
Readers’ Comments — We’d Like to Hear from You

z/OS
IBM Tivoli Directory Server Administration and Use for z/OS
Publication No. SC23-5191-03

We appreciate your comments about this publication. Please comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. The comments you send should pertain to only the information in this manual or product and the way in which the information is presented.

For technical questions and information about products and prices, please contact your IBM branch office, your IBM business partner, or your authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you. IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you state on this form.

Comments:

Thank you for your support.
Submit your comments using one of these channels:
• Send your comments to the address on the reverse side of this form.
• Send your comments via e-mail to: mhvrdfs@us.ibm.com

If you would like a response from IBM, please fill in the following information:

Name
Address
Company or Organization
Phone No.
E-mail address