

z/OS



Security Server LDAP Change Log APAR OA03857

10/20/2003

Contents

	General information	v
<hr/>		
Part 1. Overview		1
	Chapter 1. Overview	3
	Chapter 2. Software requirement	5
	Chapter 3. Updated publication	7
<hr/>		
Part 2. Publication updates		9
	Chapter 4. Change logging	11
	Configuring the GDBM backend	12
	Additional required configuration	12
	When changes are logged.	12
	Change log schema	13
	Change log entries	14
	Searching the change log	15
	Retrieving RACF password	15
	Unloading and loading the change log	15
	Trimming the change log	15
	RootDSE changes	16
	Multiserver considerations.	16
	How to set up and use the LDAP server for logging changes to RACF users	16
	Chapter 5. Preparing the backends, SSL/TLS, and password encryption	19
	Copying the configuration files	19
	Creating the DB2 database and table spaces for TDBM or GDBM	19
	Setting up for SDBM.	21
	Running SDBM with other backends	22
	Setting up for GDBM.	22
	Running GDBM with other backends	23
	Chapter 6. Customizing the LDAP server configuration.	25
	Creating the slapd.conf file	25
	Locating slapd.conf	25
	Configuration file format	25
	Configuration file checklist.	26
	Configuration file options	28
	Specifying a value for filename	28
	Configuration considerations	35
	Determining operational mode	35
	Operating in PC callable support mode	36
	Example configuration scenarios	37
	Configuring SDBM and GDBM backends	37
	Chapter 7. Accessing RACF information	39
	Mapping LDAP-style names to RACF attributes	39
	SDBM search capabilities	39
	Retrieving RACF user password envelope	39
	Using LDAP operation utilities with SDBM	40

	Chapter 8. Supported extended operations	41
I	changeLogAddEntry	41
	Trademarks	45

General information

This document applies to APAR OA03857.

Part 1. Overview

Chapter 1. Overview

APAR OA03857 provides change log support to z/OS™ Version 1 Release 3 and later releases. This support includes:

- New support to provide a log of changes made to user entries in RACF, including user password changes.
- New support to allow the log of changes to be searched by a client.
- New support to allow retrieval of an enveloped version of a RACF password.

This enhancement uses new support in z/OS Security Server RACF, available in z/OS Version 1 Release 3 and later releases. See Chapter 2, “Software requirement,” on page 5 for more information.

Chapter 2. Software requirement

The enhancements to this support require one of the following:

- z/OS Version 1 Release 4 Security Server (LDAP) (HRSL340) and later releases, including APAR OA03857
- z/OS Version 1 Release 3 Security Server (LDAP) (HRSL320) including APAR OA03857

For full exploitation of these enhancements, use the appropriate level from one of the following:

- z/OS Version 1 Release 4 Security Server (RACF) (HRF7707), and later releases, including APAR OA03853 and SAF APAR OA03854
- z/OS Version 1 Release 3 Security Server (RACF) (HRF7706), including APAR OA03853 and SAF APAR OA03854

Chapter 3. Updated publication

This document supplements the following LDAP z/OS publication, Version 1 Release 4 and previous releases:

SC24-5923 *z/OS Security Server LDAP Server Administration and Use*

Part 2. Publication updates

The chapters in this part supplement the following sections of *z/OS Security Server LDAP Server Administration and Use*.

Chapter	Supplements
Chapter 4, "Change logging," on page 11	New chapter, "Change logging"
Chapter 5, "Preparing the backends, SSL/TLS, and password encryption," on page 19	"Preparing the backends, SSL/TLS, and password encryption"
Chapter 6, "Customizing the LDAP server configuration," on page 25	"Customizing the LDAP server configuration"
Chapter 7, "Accessing RACF information," on page 39	"Accessing RACF information"
Chapter 8, "Supported extended operations," on page 41	"Supported extended operations"

Chapter 4. Change logging

This chapter contains new information on how change logging supports RACF. Change logging will:

- provide a log of changes made to user entries in RACF, including user password changes
- allow the log of changes to be searched by a client
- allow retrieval of an enveloped version of a RACF password

Change logging is information about each change to data controlled by an application (for example, a RACF password) and can be saved in a change log entry. 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. Thus, 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 a new type of backend, GDBM. The GDBM backend is envisioned as the LDAP server's global 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 or SDBM 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 or SDBM 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 configuration file. Change log processing is controlled by additional new configuration options in the GDBM backend. The `changeLogging` configuration option turns change logging on/off. The `changeLogMaxEntries` and `changeLogMaxAge` configuration options determine when removal of old change log entries takes place. See Chapter 6, "Customizing the LDAP server configuration," on page 25 for more information. If the `changeLogging` configuration option is not specified, change logging is started by default.

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.

When the LDAP server is started and change logging is enabled, an informational message is issued to indicate change logging is enabled, started/stopped, change log limits, and the number and range of change log entries. An error message is issued if the GDBM backend configuration fails to indicate that changes will not be logged.

Configuring the GDBM backend

Note: You cannot use the LDAP configuration utility, `ldapcnf`, to configure GDBM.

The following configuration file options are required to configure GDBM:

```
database GDBM GLDBGDBM [name]  
dbuserid dbowner  
servername string
```

The existing `attrOverflowSize`, `dsnaoini`, `include`, `multiserver`, `readOnly`, `sizeLimit`, and `timeLimit` configuration options can also be specified in the GDBM configuration section. The new `changeLogging`, `changeLogMaxEntries`, and `changeLogMaxAge` configuration options can also be specified in the GDBM backend. See Chapter 6, “Customizing the LDAP server configuration,” on page 25.

Note: The `suffix` option is not allowed in the GDBM backend.

You can configure a maximum of one GDBM backend in the configuration file.

The GDBM backend uses DB2 to store its entries. The GDBM database is identical to the current TDBM database and is created in the same way using the same SPUFI scripts. A GDBM backend cannot share a database with a TDBM backend.

When the LDAP server is started with GDBM configured, the GDBM schema contains the object classes and attributes used by the change log root entry and the change log entries. The GDBM schema can be modified.

Additional required configuration

Additional configuration for RACF to be able to log changes to a RACF user:

- 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. SDBM is also needed to retrieve the RACF user’s new password or other changed fields.
- LDAP Program Callable 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
```

When changes are logged

A new extended operation, `changeLogAddEntryRequest`, 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, when a user profile is added, modified, or deleted. The RACF changes can be driven through the z/OS LDAP server or be made directly to RACF. For a user password change, RACF intends to include information that the password changed in the change log entry. For other user changes, RACF does not plan to 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 z/OS 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).

Change log schema

The following object classes and their attributes define a change log entry.

- objectclass: **changeLogEntry**

MUST

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 userid passed in by RACF and the SDBM suffix.

changeType

add | modify | delete

changeTime

the timestamp of when the change is made (not when this entry is created)

MAY

changes

the added entry or the modifications, in LDIF format. This attribute is not fully supported at this time for change log entries generated by changes to RACF. It is only present when a RACF user password is changed, and will contain:

```
replace: racfPassword
racfPassword: *ComeAndGetIt*
-
```

newRDN

not supported

deleteOldRdn

not supported

newSuperior

not supported

- objectclass: **ibm-changelog**

MAY

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.

The following object class and its attributes define a change log root entry:

- objectclass: **container**

MUST

cn naming attribute for the change log root entry

The object classes and attributes used by the change log are contained in the GDBM internal schema. No schema needs to be added to GDBM to do change logging.

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 and entry owner that allow only the administrator to access the change log. An appropriately authorized user can modify the root entry to change the ACL and entry owner. A modification of the change log root results in the creation of a change log option if change logging is on. Operations on the change log root are not replicated.

The generated root entry is:

```
dn: cn=changelog
objectclass: container
cn: changelog
aclentry: group:cn=Anybody
aclPropagate: 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. The root entry ACL and entry owner are always propagated to provide access control to the change log entries because change log entries are not created with their own ACL and entry owner. 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.

- When created, change log entries do not have an ACL and entry owner specified, thus inherit a propagated ACL and entry owner. The ACL and entry owner on the change log root entry are propagated to provide this inherited access control. An ACL and entry owner can be added to a change log entry using a modify operation.
- Change log entries are only created by the LDAP server. The user cannot directly add a change log entry.
- A change log entry can be modified, mainly to set an ACL and entry owner. Most of the attributes in the change log entry are not modifiable by the user.
- A change log entry cannot be renamed.
- Change log entries are deleted 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 through a normal delete operation.
- User operations (search, modify, 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.

- 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, with the changes attribute shown in plain text rather than base-64 encoded:

```
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: racfPassword
racfPassword: *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).
- Unlike other LDAP searches, 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.

Retrieving RACF password

Using SDBM, you can retrieve the RACF password envelope for a RACF user. See Chapter 7, "Accessing RACF information," on page 39, *Retrieving RACF user password envelope*, for more information.

Unloading and loading the change log

The unload utility (tdbm2ldif) 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 bulkload utility (ldif2tdbm) fail when processing change log options.

Trimming the change log

When change logging is on, the change log is periodically trimmed based on the limits set in the 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 until the number of entries is about 95% of the limit. For example, if `changeLogMaxEntries` is 1000 and the number of entries in the change log reaches 1000, the 50 lowest entries are deleted to reduce the number of entries to 950.

- Change log entries are always removed from lowest change number to highest.
- The change log is checked for trimming when the server is started and when change log entries are added. 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.
- Trimming is only performed when change logging is on.

RootDSE changes

The following attributes are added to the rootDSE to 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.

Multiserver considerations

Each server should have an identical GDBM backend configured. If a server does not have change logging on, changes made through that server are not logged, however, the change log can still be searched through that server. An LDAP Server with change logging on has to be running on the system where the RACF change is made.

How to set up and use the LDAP server for logging changes to RACF users

1. Update the LDAP server configuration file:
 - a. Add the GDBM backend section, including a change log size and age limit if desired. The following example starts change logging using a change log with a maximum size of 1000 entries. Entries are automatically deleted when they become a day old.

```
database gdbm GLDBGDBM
dbuserid dbu1
servername loc1
changeLogging on
changeLogMaxEntries 1000
changeLogMaxAge 86400
```

- b. Add the SDBM backend section. Following is an example:

```
database sdbm GLDBSDBM
suffix cn=myRacf
```

- c. 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
```

2. Create the DB2 database to be used by the change log. This involves updating and executing two SPUFI scripts. The database owner in the scripts must match the dbuserid value in the GDBM section of the configuration file. See *Creating the DB2 database and table spaces for TDBM and GDBM* for more information.
3. Perform the RACF configuration required to support creation of a password envelope, retrieval of the password envelope, and creation of an LDAP change log entry for changes to a RACF user. See *RACF Password Enveloping SPE* documentation for more information.
4. Restart the LDAP server. You should receive a message similar to the following:

```
GLD0244I Change logging is enabled
Logging started status (0 = off, 1 = on): 1
Limit in seconds on age of change log entries (0 = no limit): 86400
Limit on the number of change log entries (0 = no limit): 1000
Current number of change log entries: 0
First change number in use: 0
Last change number in use: 0
```

If the LDAP server fails to configure the change log during startup, the following message is issued:

```
GLD0245I Change log configuration has failed and change logging is not enabled.
```

5. At this point, change logging is started and a change to a RACF user 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. You may want to consider allowing the same user which will access RACF information to read the log.

For example, to give read access to the change log to RACF user CLREADER, create an ldif file, cl.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 -p nnn -D adminDn -w adminPw -f cl.ldif
```

7. You can search, delete, and modify (to a limited extent) 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 -p nnn -D adminDn -w adminPw -b "cn=changelog" "objectclass=**"
```

Typical output with the changes attribute shown in plain text rather than base-64 encoded:

```
cn=changelog
objectclass=top
objectclass=container
cn=changelog
```

```
CHANGENUMBER=1,CN=CHANGELOG
objectclass=CHANGELOGENTRY
```

```
objectclass=IBM-CHANGELOG
objectclass=TOP
changenumber=1
targetdn=RACFID=U2,PROFILETYPE=USER,cn=myRacf
changetime=20030611204814.257756Z
changetype=MODIFY
changes=replace: racfPassword
racfPassword: *ComeAndGetIt*
-
```

```
ibm-changeinitiatorsname=RACFID=SUSET3,PROFILETYPE=USER,cn=myRacf
```

8. If the changes attribute of a change log entry contains:

```
racfPassword: *ComeAndGetIt*
```

then the RACF password envelope containing the new password can be obtained using a command similar to the following:

```
ldapsearch -p nnn -D "racfid=admin1,profiletype=user,cn=myRacf" -w admin1Pw -L
-b "racfid=u2,profiletype=user,cn=myRacf" "objectclass=*" racfpasswordenvelope
```

The beginning of typical output would look as follows:

```
dn: racfid=U3,profiletype=USER,cn=myRacf
racfpasswordenvelope::MIIFHQYJKoZIHvcNAQcDoIIFDjCCBQoCAQAxcgswgcgCAQAwMTAsMQ
swCQYDVQQGEwJVUzEMMAoGA1UEChMDSUJNMQ8wDQYDVQQDEwZSQUNGGQ0ECAQIwDQYJKoZIHvcNAQ
EBBQAEgYBA0vsirICCwD00LYrns5HS1iYV4Xjz0YL6XVDAaIA6vARW9xk0g7Wm9uGtTyAEU574CW
...
```

Note: the password envelope is a binary value that is base-64 encoded

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:

```
ldapsearch -p nnn -D adminDn -w adminPw -V 3 -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.

Chapter 5. Preparing the backends, SSL/TLS, and password encryption

The following supplements the *Preparing the backends, SSL/TLS, and password encryption* chapter.

This chapter discusses what you need to do to prepare the backends, SSL/TLS, and password encryption.

If you plan to use:	You must:	See:
A backend, such as TDBM or SDBM	Copy the configuration files.	<i>Copying the configuration files</i>
The sample server to set up a DB2 database	Use the set of example files shipped with the code.	<i>Creating a DB2 database for the sample server below</i>
TDBM backend	Create the DB2 database and table spaces using SPUFI.	<i>Creating the DB2 database and table spaces for TDBM</i>
SDBM backend	Set up your configuration files.	<i>Setting up for SDBM</i>
GDBM backend	Set up your configuration files.	<i>Setting up for GDBM</i>
EXOP backend	Set up your configuration files.	<i>Setting up for extended operations</i>
SSL/TLS	Enable SSL/TLS support.	<i>Setting up for SSL/TLS</i>
Password encryption	Determine the type of encryption and set up the configuration.	<i>Configuring for user password encryption</i>

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 install directory because any service to the files will overwrite the modifications. Instead, modify them in **/etc/ldap**. The following commands copy the configuration files:

```
cp /usr/lpp/ldap/etc/slapd.* /etc/ldap/.
```

The **cp** command creates a copy of the **/usr/lpp/ldap/etc** files into the **/etc/ldap** directory. The individual files can now be modified using **oedit** or **vi**.

Creating the DB2 database and table spaces for TDBM or GDBM

When using TDBM or GDBM, the LDAP server DB2 database must be created by running two SPUFI (SQL Processor Using File Input) scripts from DB2 Interactive (DB2I). The same scripts are used for both TDBM and GDBM. 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, SC26-9933*. Sample DB2I SPUFI scripts to create the LDAP server DB2 database are provided. To use them, do the following:

1. Copy the SPUFI scripts over to your SPUFI input data set.

The SPUFI script for creating the database and table spaces can be found in *GLDHLQ.SGLDSAMP(TDBMDB)* and the script for creating the table indexes can be found in *GLDHLQ.SGLDSAMP(TDBMINDX)*. (*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 these SPUFI files, as shown in the *Configuring an LDAP server without the ldapcnf utility* chapter. The SPUFI scripts provide specific instructions and information to help you determine the values to use in the table. *The TDBMDB SPUFI file* and *The TDBMINDX SPUFI file* chapters show examples of the files to edit and run in the SPUFI facility.

Table 1. TDBM or GDBM value overview

Attribute	Value	Suggested value	Variable name in SPUFI script
Database information for TDBMDB and TDBMINDX members			
Database name		LDAPSRV	-DDDDDDDD-
Database owner		LDAPSRV	-UUUUUUUU-
Table space definitions for TDBMDB member			
Entry table space name		ENTRYTS	-AAAAAAA-
Buffer pool name for the LDAP entry table space		BP0	-BBBB-
Long entry table space name		LENTYTS	-CCCCCCC-
Buffer pool name for the LDAP long entry		BP0	-DDDD-
Long attribute table space name		LATTRTS	-EEEEEEEE-
Buffer pool name for the LDAP long attribute		BP0	-FFFF-
Miscellaneous table space name		MISCTS	-GGGGGGG-
Search table space name		SEARCHTS	HHHHHHHH-
Buffer pool name for the LDAP search table		BP0	-IIII-
Replica table space name		REPTS	-JJJJJJJ-
Descendants table space name		DESCTS	-KKKKKKKK-
Storage group		SYSDEFLT	-SSSSSSSS-
Search column truncation size (VALUE in DIR_SEARCH)		32	-TTTT-
DN truncation size (DN_TRUNC in DIR_ENTRY)		32	-MMMM-
Maximum size of a DN (DN in DIR_ENTRY)		512	-NNNN-

3. **Modify the scripts.**

Use the values from Table 1 to modify the scripts. You must have a unique database name and owner for each database you are creating.

4. **Run the scripts from DB2I SPUFI.**

Use the DB2 SPUFI (SQL Processor Using File Input) facility to create the database and table spaces.

Be sure to run the two scripts that were copied and modified in the previous steps under a user ID with DB2 **SYSADM** authority. When the scripts complete running, scan the output data set to ensure that they ran successfully.

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

In order to run the LDAP server and server utilities, 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 or LDAP server utility, *yyy* is the database name identified in the **slapd.conf** (for TDBM only) and SPUFI file for the **databasename** option and *zzz* is the CLI plan name as specified in your DB2 CLI initialization file. Run the following statements through SPUFI (DB2 Interactive):

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

These privileges may be granted by any user ID with **SYSADM** authority. The commands above can be run using the DB2 SPUFI facility.

The LDAP server requires **SELECT** access to the SYSIBM.SYSCOLUMNS table in DB2. If **SELECT** access to this table is tightly controlled in your DB2 installation, then it may be necessary to grant this access to the user ID under which the LDAP server runs by performing the following operation (either through SPUFI or another means of issuing SQL commands):

```
grant select on sysibm.syscolumns 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 will fail during start-up with an SQL -551 return code.

Setting up for SDBM

The LDAP server can provide LDAP access to the user and group information stored in RACF. See *Accessing RACF information* chapter for details about how you can use this RACF information.

In order to configure your LDAP server to run with the SDBM database 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* in this chapter).
- You need to use the following lines in your **slapd.conf** file:

```
database sdbm GLDBSDBM  
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"
```

An SDBM suffix should not contain an alias name for an attribute. For example, the suffix cannot use the **surName** attribute (it can use the **sn** attribute instead). Also, the suffix can contain a case-sensitive attribute, but SDBM ignores case when processing the suffix.

- Be sure to set **STEPLIB** to point to the *GLDHLQ.SGLDLNK* data set before running the LDAP server with SDBM, otherwise an error will be returned. (*GLDHLQ* refers to the high-level qualifier that was used to install the LDAP server data sets.)

Notes:

1. Only one SDBM database can be defined in any given LDAP server.
2. SDBM contains an internal schema that it uses to check entries that it is adding. The schema cannot be modified.

Running SDBM with other backends

The following table gives you information on running SDBM alone or with other backend databases.

Table 2. SDBM with other backends

Backend	Description
SDBM with TDBM	<p>The TDBM schema will be used for all initial DN normalization if TDBM is configured. DN normalization is performed by the server to aid in selecting the appropriate backend. All attribute types that might appear in a RACF-style DN must be defined to the TDBM schema.</p> <p>When starting TDBM and SDBM together, ensure that the attribute types in the SDBM suffix are also present in the TDBM schema. Examine the schema LDIF files shipped with the LDAP server to determine which schema must be loaded into TDBM.</p>
SDBM only or SDBM with GDBM	<p>If you are running SDBM without TDBM, be sure to comment out the TDBM database definitions in the slapd.conf file. Prefix each line to comment with a # (pound sign). When running without TDBM, replication and referrals are not supported.</p>

Setting up for GDBM

The LDAP server can provide a change log containing information about changes to RACF users.

In order to configure your LDAP server to run with the GDBM database 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* in this chapter).
- You need to use the following lines in your **slapd.conf** file:

```
database gdbm GLDBGDBM
dbuserid userid
servername string
```

See Chapter 4, “Change logging,” on page 11 for additional configuration options that can be specified.

- You must also configure an SDBM backend and enable the LDAP Program Callable support. See *Setting up for SDBM*, in the previous section, for more information and Chapter 4, “Change logging,” on page 11, *Additional required configuration*.

- Be sure to set **STEPLIB** to point to the *GLDHLQ.SGLDLNK* data set before running the LDAP server with GDBM, otherwise an error will be returned. (*GLDHLQ* refers to the high-level qualifier that was used to install the LDAP server data sets.)

Notes:

1. Only one GDBM database can be defined in any given LDAP server.
2. GDBM contains an internal schema that it uses to define entries that it is adding. The schema can be modified.

Running GDBM with other backends

The following table gives you information on running GDBM with other backend databases.

Table 3. GDBM with other backends

Backend	Description
GDBM with SDBM and TDBM	The TDBM schema will be used for all initial DN normalization if TDBM is configured. DN normalization is performed by the server to aid in selecting the appropriate backend. All attribute types that might appear in a RACF-style DN must be defined to the TDBM schema.
GDBM with SDBM	If you are running GDBM without TDBM, be sure to comment out the TDBM database definitions in the slapd.conf file. Prefix each line to comment with a # (pound sign). When running without TDBM, replication and referrals are not supported.

Chapter 6. Customizing the LDAP server configuration

The following supplements the *Customizing the LDAP server configuration* chapter.

This chapter contains information on how to set up the **slapd.conf** configuration file and how to configure the LDAP server to run with the options you choose. The **slapd.conf** file is also used by the LDAP utilities. Unchanged configuration options are not shown.

Creating the slapd.conf file

This section discusses what is necessary for creating the **slapd.conf** configuration file. Specifically, this section:

- Describes where the **slapd.conf** file is located
- Shows the configuration file format
- Provides a checklist for the configuration file options
- Lists all of the configuration file options
- Describes how to establish the administrator DN and password
- Discusses encryption using the **pwEncryption** option

Locating slapd.conf

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

```
cp /usr/lpp/ldap/etc/slapd.conf /etc/ldap/slapd.conf
```

and edit **/etc/ldap/slapd.conf**.

An alternate configuration file can be specified through a command-line option to the LDAP server and other LDAP programs.

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

Configuration file format

The **slapd.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.

GDBM backend-specific section

Contains configuration options that apply to the GDBM 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.

EXOP backend-specific section

Contains only the **database** statement necessary for the EXOP backend.

Noted below are some rules for setting up **slapd.conf**:

- The configuration file always contains a global section followed by one or more database backend sections that contain information specific to a backend instance.
- The configuration file or files must be in code page IBM-1047.
- For single-valued options that appear more than once, the last appearance in the **slapd.conf** file is used.
- Blank lines and comment lines beginning with the pound sign character (#) are ignored.
- If a line begins with one or more blank spaces, it is considered a continuation of the previous line.
- If an argument contains one or more blank spaces, the argument should be enclosed in double quotation marks (for example, "argument one"). If an argument contains a double quotation mark or a backslash character (\), the double quotation mark or backslash character should be preceded by a backslash character (\).
- A pound sign (#) **cannot** be used at the end of a configuration line to denote commentary.

```

| # Global options - these options apply to every database
| <global configuration options>
|
| # SDBM database definition and configuration options
| database sdbm GLDBSDBM
| <configuration options specific to SDBM backend>
|
| # GDBM database definition and configuration options
| database gdbm GLDBGDBM
| <configuration options specific to GDBM backend>
|
| # TDBM database definition and configuration options
| database tdbm GLDBTDBM
| <configuration options specific to TDBM backend>
|
| # EXOP database definition and configuration options
| database exop GLDXPDIR

```

Figure 1. General format of slapd.conf

Configuration file checklist

The following table is provided to assist you in determining which configuration file options you will need to use in your **slapd.conf** file. Depending on the section in the configuration file (Global, SDBM, GDBM, TDBM, or EXOP), certain topics (SSL, schema, replication, and so on) have options that are required or optional.

Table 4. Configuration file options checklist

Section/topic	Check	Options
Global		adminDN is required adminPW , altServer , commThreads , digestRealm , idleConnectionTimeout , include , listen , logfile , maxConnections , pcThreads , referral , sendV3stringsoverV2as , serverEtherAddr , sizeLimit , timeLimit , and validateincomingV2strings are optional.

Table 4. Configuration file options checklist (continued)

Section/topic	Check	Options
SSL/TLS		sslAuth , sslCertificate , sslKeyRingFilePW , sslKeyRingPWStashFile , and sslCipherSpecs are optional 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 once the connection is established.
Sysplex		sysplexGroupName and sysplexServerName are optional
Kerberos		supportKrb5 and serverKrbPrinc are required krbKeytab and krbLDAPAdmin are optional
SDBM backend		database and suffix are required sizeLimit and timeLimit are optional
Kerberos		krbIdentityMap is optional
GDBM backend		database , dbuserid , and servername are required attrOverflowSize , changeLogging , changeLogMaxAge , changeLogMaxEntries , dsnaoini , include , readOnly , sizeLimit , and timeLimit are optional
Multi-server		multiserver is optional
TDBM backend		database , databasename , dbuserid , servername , and suffix are required attrOverflowSize , dsnaoini , extendedGroupSearching , readOnly , sizeLimit , and timeLimit are optional
Password encryption		pwEncryption is optional
Replication		masterServer , masterServerDN , and masterServerPW are optional
Multi-server		multiserver is optional
Kerberos		krbIdentityMap is optional
Native authentication		useNativeAuth and nativeAuthSubtree are required nativeUpdateAllowed is optional
EXOP backend		database is required

Note: Be sure to specify **adminDN**. You can specify the **adminPW** here or in a database entry. 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 alphabetical listing of the configuration file options. For each option, a table shows an **X** in the areas (Global, TDBM, SDBM, and EXOP) of the configuration file where the option can be used.

Specifying a value for filename

In the configuration file options, the value for *filename* can be specified in one of the following ways:

/pathname/filename

Specifies the full path name of a file in the USS Hierarchical File System (HFS).

filename

Specifies a path name that is relative to the current working directory of the LDAP server. Note that when running from a started task or batch, there is no current working directory defined. This format is not recommended.

//dataset.name'

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

//dataset.name(member)'

Specifies the fully-qualified name of a configuration file stored in a partitioned dataset.

//DD:DDNAME

Specifies the DDNAME of a configuration that has been specified as a DD card in the JCL for the batch job or started task.

attrOverflowSize *num-of-bytes*

Global	TDBM	SDBM	GDBM	EXOP
	X		X	

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 value must be between 1 and 2147483647.

Default = 255

changeLogging {on | off}

Global	TDBM	SDBM	GDBM	EXOP
			X	

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

changeLogMaxAge *nnn*

Global	TDBM	SDBM	GDBM	EXOP
			X	

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.

Default = 0

changeLogMaxEntries *nnn*

Global	TDBM	SDBM	GDBM	EXOP
			X	

Specifies the maximum number of entries that the change log can contain. If the number of change log entries reaches this value, change log entries with the lowest change numbers are deleted until the number of remaining entries is 95% of the maximum, except if **changeLogging off** is specified. The value must be between 0 and 2147483647. A value of 0 indicates that there is no maximum.

Default = 0

database *dbtype dblibpath [name]*

Global	TDBM	SDBM	GDBM	EXOP
	X	X	X	X

Marks the beginning of a new database section.

- For *dbtype*:
 - IBM supports **tdbm** (DB2), **sdbm** (RACF), **gdbm** (DB2) and **exop** (extended operations). The type **config** is reserved by the LDAP server and should not appear as **dbtype** in your configuration files.
- For *dblibpath*:
 - 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 **GLDBTDBM** when *dbtype* is **tdbm**, **GLDBSDBM** when *dbtype* is **sdbm**, **GLDBGDBM** when *dbtype* is **gdbm**, and **GLDXPDIR** when *dbtype* is **exop**.
- For *name*:
 - This optional value is a name that is used to identify this backend. If specified, the *name* must be different (ignoring case) than the *name* specified on any other **database** option in this configuration file. You cannot specify **cdbm1** as the *name*.

databasesname *dbname*

Global	TDBM	SDBM	GDBM	EXOP
	X			

Specifies the name of the DB2 database this backend uses to store directory data.

dbuserid *userid*

Global	TDBM	SDBM	GDBM	EXOP
	X		X	

Specifies a z/OS user ID that will be the owner of the DB2 tables.

dsnaoini *filename*

Global	TDBM	SDBM	GDBM	EXOP
	X		X	

Specifies the name of the CLI Initialization file or sequential data set (or PDS member) you created in the steps described in *Installing and setting up related products* chapter. If the **dsnaoini** option is set in the configuration file, the LDAP server will export the **DSNAOINI** environment variable to the value specified for the configuration option.

In addition to using the **dsnaoini** configuration option or the **DSNAOINI** environment variable, a DSNAOINI DD card can be used to specify the CLI Initialization file. If the DSNAOINI DD card is specified in the JCL for the job/started task for the LDAP server, then neither the **dsnaoini** configuration option value nor the **DSNAOINI** environment variable need to be specified. For more information on this process, see *Running the LDAP server using data sets* chapter. See *DB2 for OS/390 and z/OS: ODBC Guide and Reference* book for details on ways to specify the CLI initialization file.

See *DB2 for OS/390 and z/OS: ODBC Guide and Reference* for other ways in which the DSNAOINI ODBC initialization file can be specified. In order for the TDBM or GDBM backend to run, the initialization file must be specified in one of the ways indicated.

include *filename*

Global	TDBM	SDBM	GDBM	EXOP
X	X	X	X	X

Specifies the path and file name of a file to be included as a part of the LDAP server configuration.

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.

listen *ldap_URL*

Global	TDBM	SDBM	GDBM	EXOP
X				

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 parameter 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 optional start-up parameter for **listen**.

Default = **INADDR_ANY** (that is, `ldap://:389`) on the nonsecure default port 389.

The format of *ldap_URL* for the **listen** option to listen on a TCP/IP socket interface is the following:

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

The format of *ldap_URL* for the **listen** option to listen on the z/OS SAF interface is the following:

```
{ldap:// | ldaps://}: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.

ldaps://

Used to have the server listen on secure addresses or ports. Once a connection is established to the server, the client must begin the SSL/TLS handshake protocol.

IP_address

Specifies the IP address.

hostname

Specifies the host name. If the host name is used for the **listen** option, all of the IP addresses are obtained and the LDAP server listens on each of these IP addresses.

portNumber

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

Range = 1-65536

If the **sysplexGroupName** and **sysplexServerName** options are 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 same *group_name* in the sysplex 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.

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.

|
|
|

Note that when the **listen** option is initialized to listen for PC calls on the LDAP server, the **listen** parameter must not include an IP address or a host name.

Also, there is no difference if you specify **ldap** or **ldaps** as part of *ldap_URL*. Both produce the same result.

Following are some examples of how you can specify *ldap_URL*.

- If you specify:

```
ldap://
```

the LDAP server binds and listens on all available IP addresses (**INADDR_ANY**) on the system on the nonsecure default port of 389.

- If you specify:

```
ldap://us.endicott.ibm.com:489
```

the LDAP server binds and listens on all of the IP addresses associated with host name `us.endicott.ibm.com` on the nonsecure port of 489 for incoming client requests.

- If you specify:

```
ldap://9.130.77.27
```

the LDAP server binds and listens on IP address `9.130.77.27` on the default nonsecure port of 389 for incoming client requests.

- If you specify:

```
ldaps://us.endicott.ibm.com
```

the LDAP server binds and listens for incoming client requests on the IP address or 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 IP 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 IP 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 IP addresses (**INADDR_ANY**) on the system on the secure port of 777 for incoming client requests.

- If you specify:

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

the LDAP server binds and listens for PC calls from Policy Director 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 will be 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.

To migrate from one or more of the **port**, **securePort**, or **security** options to the **listen** configuration option.

multiserver {Y | y | N | n}

Global	TDBM	SDBM	GDBM	EXOP
	X		X	

Indicates the operating mode in which this server will run. Specifying either **y** or **Y** indicates the server runs in multi-server mode with or without dynamic workload management enabled (see page *z/OS Integrated Security Services LDAP Server Administration and Use* for a description of multi-server operating modes). Specifying either **n** or **N** indicates the server runs in single-server mode.

If **n** or **N** is specified, and both **sysplexGroupName** and **sysplexServerName** options are present in the configuration file, the **multiserver** option value is overridden and the server operates in multi-server mode.

The **multiserver** keyword may be present without the **sysplexGroupName** and **sysplexServerName** keywords, in which case replication is disabled in the server and sysplex Workload Management features are also disabled.

If **sysplexGroupName**, **sysplexServerName**, and **multiserver** keywords are all omitted from the server configuration file, the server will operate in single-server mode and replication will be enabled.

readOnly {on | off}

Global	TDBM	SDBM	GDBM	EXOP
	X		X	

Specifies the ability to modify the database. Any attempt to use the LDAP server to modify the database will fail if **readOnly** is turned **on**.

Note: For GDBM, change log entries continue to be created and trimmed (deleted) by the LDAP server even when **readOnly** is on.
Default = off

servername *string*

Global	TDBM	SDBM	GDBM	EXOP
	X		X	

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 which is specified by the **dsnaoini** option in the configuration file.

sizeLimit *int*

Global	TDBM	SDBM	GDBM	EXOP
X	X	X	X	

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.

0 = no limit
 Default = 500
 Range = 0 - 2147483647

This option applies to all backends, unless specifically overridden in a backend definition. Specifying this prior to a **database** line in the configuration sets the option for all backends. Specifying it after a **database** line sets the option just for the backend defined by the **database** line.

Note that the following behavior is used when referring to the **sizeLimit** parameter. This parameter is also valid on an **ldapsrch** from the client.

- If a client has passed a limit, then the smaller value of the client value, and the value read from **slapd.conf** will be used.
- If the client has not passed a limit, and has bound as the **adminDN**, then the value specified in **slapd.conf** will be ignored.
- If the client has not passed a limit, and has not bound as the **adminDN**, then the limit will be that which was read from the **slapd.conf** file.

Note: When using the RACF access support of the z/OS LDAP server (SDBM), the number of entries returned will be subject to the limits of the SDBM implementation. The minimum of all of the restrictions and **sizeLimit** settings and specifications is the value used for any one search.

timeLimit *int*

Global	TDBM	SDBM	GDBM	EXOP
X	X	X	X	

Specifies the maximum number of seconds (in real time) the LDAP server will spend answering a search request or a Modify DN 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.

0 = no limit
 Default = 3600
 Range = 0 - 2147483647

This option applies to all backends, unless specifically overridden in a backend definition. Specifying this prior to a **database** line in the

configuration sets the option for all backends. Specifying it after a **database** line sets the option just for the backend defined by the **database** line.

Note that the following behavior is used when referring to the **timeLimit** parameter. This parameter is also valid on an **ldapsearch** from the client.

- If a client has passed a limit, then the smaller value of the client value, and the value read from **slapd.conf** will be used.
- If the client has not passed a limit, and has bound as the **adminDN**, then the value specified in **slapd.conf** will be ignored.
- If the client has not passed a limit, and has not bound as the **adminDN**, then the limit will be that which was read from the **slapd.conf** file.

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 5. Configuration considerations

Dependency	More information
<p>HFS environment versus PDS (dataset) environment</p> <p>Depending on whether you use an HFS or PDS environment, there are some areas to consider for each.</p>	<p><i>Setting up and running the LDAP server in the z/OS shell and Setting up and running the LDAP server as a started task</i></p>
<p>Operational mode</p> <p>You need to determine the type of operational mode your LDAP server will run in. For example, single-server mode or multi-server mode with or without dynamic workload management enabled.</p>	<p><i>Determining operational mode</i></p>
<p>TDBM backend</p> <p>You can use a TDBM backend database based on DB2.</p>	<p><i>LDAP directory schema for TDBM</i></p>
<p>SDBM backend</p> <p>You can use an SDBM backend database based on RACF.</p>	<p><i>Setting up for SDBM</i></p>
<p>GDBM backend</p> <p>You can use a GDBM backend database based on DB2.</p>	<p><i>Setting up for GDBM</i></p>
...	

Determining operational mode

- **Program call (PC) callable support mode**

The program call (PC) callable support in LDAP provides a program call interface to the LDAP extended operations backend (EXOP). 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 a RACF user 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), SDBM, GDBM, and EXOP backends are supported. The GDBM backend requires the SDBM backend for full functionality.

Notes:

1. A single LDAP server instance can have one SDBM backend and one GDBM backend, but it can have multiple TDBM 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 will be 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.

Operating in PC callable support mode

The program call (PC) callable support in LDAP provides a program call interface to the LDAP extended operations backend (EXOP). 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 a RACF user in the LDAP 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 normal 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 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 will be 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, which 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 normal 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 LDAP in a sysplex, configure one LDAP server on each system in the sysplex to run the PC callable support. Each system should share the DB2 and RACF databases to ensure that they return the same results.

Example configuration scenarios

This section shows scenarios of LDAP server configurations.

Configuring SDBM and GDBM backends

The configuration example in this section uses SDBM and GDBM backends and shows the configuration file checklist next to the corresponding sample configuration file.

Table 6. Sample checklist and slapd.conf (using SDBM and GDBM)

Section	Check	Sample slapd.conf
Global	√	# Filename slapd.conf
SSL/TLS		# Global section
Sysplex		sizeLimit 500
Kerberos		timeLimit 3600
		adminDn "racfid=ldadmin,profiletype=user,cn=myRACF"
SDBM backend	√	listen ldap://:pc
		listen ldap://:389
Kerberos		
GDBM backend	√	# SDBM backend section
		database sdbm GLDBSDBM
Multi-server		suffix "cn=myRACF"
TDBM backend		# GDBM backend section
Password encryption		database gdbm GLDBGDBM
		servername LOC1
		dbuserid LDAPSRV
Replication		attrOverflowSize 500
Multi-server		
Kerberos		
Native authentication		
EXOP backend		

Chapter 7. Accessing RACF information

The following supplements the *Accessing RACF information* chapter.

RACF provides definitions of users and groups, as well as access control for resources. The LDAP server can provide LDAP access to the user and group information stored in RACF.

Using SDBM, the RACF database backend of the LDAP server, you can:

- Add new users and groups to RACF
- Add users to groups (connections)
- Modify RACF information for users and groups
- Retrieve RACF information for users and groups
- Delete users and groups from RACF
- Remove users from groups (connections)
- Retrieve RACF user password envelope

Mapping LDAP-style names to RACF attributes

Following are tables that show the RACF attribute name and the corresponding LDAP-style attribute name for user, group, and connection.

Table 7. Mapping of LDAP-style names to RACF attributes (user)

RACF segment name	RACF attribute name in altuser/adduser string	LDAP-style attribute name
User base or group base	DATA	racfInstallationData
User base or group base	MODEL	racfDatasetModel
User base	Not modifiable; listuser displays as CREATED	racfAuthorizationDate
User base	OWNER	racfOwner
User base	Multi-value: ADSP, SPECIAL, OPERATIONS, GRPACC, AUDITOR, OIDCARD, UAUDIT	racfAttributes
User base	PASSWORD	racfPassword
User base	password envelope — not modifiable	racfPasswordEnvelope
.....		

SDBM search capabilities

Retrieving RACF user password envelope

SDBM is enhanced to return this envelope when the **racfPasswordEnvelope** attribute is specified in the attributes to be returned from a search of a RACF user. The envelope is returned by the LDAP server as a binary data berval (binary data

and length). If the **racfPasswordEnvelope** attribute is not specified on the search request, the RACF password envelope is not returned.

Note: When using a utility such as `ldapsearch` to retrieve the password envelope, the returned value is base-64 encoded.

Using LDAP operation utilities with SDBM

The LDAP operation utilities described in the *z/OS Security Server LDAP Client Programming* 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: retrieving RACF user password envelope

The following search returns the **racfPasswordEnvelope** attribute:

```
ldapsearch -D racfid=admin1,profiletype=user,sysplex=sysplexa -w passwd -L
-b racfid=yyy,profiletype=user,sysplex=sysplexa "objectclass=*" racfpasswordenvelope

dn: racfid=yyy,profiletype=user,sysplex=sysplexa
racfpasswordenvelope::base-64_encoded_password_envelope
```

Chapter 8. Supported extended operations

The following supplements the *Supported extended operations* chapter.

The sections that follow describe the supported extended operations. For information on ASN.1 (Abstract Syntax Notation One) and BER (Basic Encoding Rules), go to the following Web site:

<ftp://ftp.rsa.com/pub/pkcs/ascii/layman.asc>

changeLogAddEntry

- **Name:** `changeLogAddEntry`
- **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:** TBD
- **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,
    changeType ENUMERATED {
        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 1. 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 was 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 was created, modified, deleted, or renamed. The RACF application can specify a value for both `userid` and `group` to both 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.

`changeType` ::= an enumerated 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: `yyyymmddhhii:ss.uuuuuuZ`

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 SET OF AttributeValue,
    action ENUMERATED {
        add (0),
        replace (1),
        delete (2) },
    requestValue Boolean }
```

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.

- **Response object identifier:** TBD
- **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.

```
ResponseValue ::= SEQUENCE {
    changeLogResultCode ENUMERATED {
        success (0),
        loggingFailed (1),
        invalidCredentials (2),
        remoteNotSupported (3),
        notConfigured (4),
        notActive (5),
        decodeFailed (6),
        valueOutOfRange (7),
        dnConvertFailed (8)
    }
    errorMessage LDAPString
}
```

where,

Distinguished-name LDAPDN

- **Response detailed description:**
The following table summarizes some different error scenarios and the `changeLogAddEntry` response to such scenarios.

Error scenario	changeLogAddEntry's response
An internal error prevents the logging operation from completing	Returns a loggingFailed return code

Error scenario	changeLogAddEntry's response
The caller is not in supervisor state	Returns an invalidCredentials return code
Change log is not configured	Returns a notConfigured return code
Change log is not active	Returns a notActive return code
LDAP server is unable to parse the request	Returns a decodeFailed return code
Value is outside the range of allowable values	Returns a valueOutOfRange return code
LDAP server is unable to convert a RACF userid to an LDAP DN	Returns a dnConvertFailed return code

Trademarks

The following terms are trademarks of the IBM Corporation in the United States, or other countries, or both:

IBM

MVS

RACF

z/OS

z/OS.e

Other company, product, and service names might be trademarks or service marks of others.