One Identity Quick Connect for Mainframes 2.3.0

QCMF Bridge

Legend

WARNING: A WARNING icon indicates a potential for property damage, personal injury, or death.

CAUTION: A CAUTION icon indicates potential damage to hardware or loss of data if instructions are not followed.

IMPORTANT, NOTE, TIP, MOBILE, or VIDEO: An information icon indicates supporting information.

Quick Connect for Mainframes QCMF Bridge
Updated - October 2017
Version - 2.3.0

Contents

About One Identity Quick Connect for Mainframes (bridge 3.5.5)

This guide will help familiarize you with Quick Connect for Mainframes (bridge). The Installation Guide provides the information you need to install and use Quick Connect for Mainframes (bridge). It is intended for network administrators, consultants, analysts, and any other IT professionals using the product.

This guide is intended for security administrators and system programmers who are experienced in:

- basic LDAP concepts such as directory schema and LDAP operations
- mainframe concepts such as JCL, partitioned data sets, and job submission
- mainframe UNIX System Services (USS) concepts such as how to access USS, HFS file structure, and basic UNIX command syntax
- have the authority to access USS, enter UNIX commands, and create HFS files
- IBM RACF® concepts such as password verification and resource authorization
- CA Top Secret® concepts such as password verification and resource authorization
- CA ACF2™ concepts such as password verification and resource authorization.

These personnel must have authority to:

- edit mainframe data sets and submit jobs, and install exits
- access USS, run UNIX commands, and create HFS files.

The following variables refer to values specific to your site:

- `installDirectory` - refers to the root directory in Unix System Services that you choose for this product
- `HLQ` - refers to the high-level qualifier(s) you select of the MVS data sets installed by this product
- `systemName` - refers to the system name where the LDAP Bridge is installed

- secs - refers to your security system:
  - racf - RACF
  - tss - Top Secret
  - acf2 - ACF2

Examples of directory paths apply to a multi-system installation. Refer to Directories created during installation and configuration for the equivalent directories for a single-system installation.

Please refer to the One Identity Quick Connect documentation on http://support.oneidentity.com for additional information and guidance on One Identity Quick Connect.

Introduction

The Quick Connect for Mainframes (bridge) (referred to in this publication as the LDAP Bridge) is an LDAP gateway that provides access to IBM Resource Access Control Facility (IBM RACF®), CA Top Secret® (Top Secret) and CA ACF2™ (ACF2). By enabling you to access mainframe security-based data with LDAP, the LDAP Bridge extends mainframe authentication and authorization to your environment.

Overview of the LDAP Bridge

The LDAP Bridge consists of the following three components:

LDAP server component

The LDAP server publishes a copy of the RACF, Top Secret, or CA ACF2™ (mainframe security) database. The database copy that is published is a real-time image of the chosen subset of the mainframe security database as it resides on the host z/OS system.

LDAP Command Translator component

The LDAP Command Translator modifies the mainframe security database to reflect the changes that were initiated within the LDAP Bridge. Whenever users submit a change to the LDAP server, the LDAP Command Translator transforms the LDAP modify command into an equivalent RACF, Top Secret, or ACF2 command so that the mainframe security database is modified accordingly. When the change has been made to the mainframe security

database, the Synchronization Daemon processes and reflects the change in the LDAP database.

Synchronization Daemon component

The Synchronization Daemon updates the database copy to reflect the current status of the mainframe security database. Whenever a change is made to the mainframe security database, the Synchronization Daemon reads the audit record that is generated by RACF, Top Secret, or ACF2 in response to the command. The RACF, Top Secret, or ACF2 command is then translated into an equivalent LDAP command that updates the database copy accordingly. If the LDAP Bridge is stopped, RACF, Top Secret, and ACF2 changes accumulate in the Synchronization Daemon directory until it is restarted so that no changes are lost. The Synchronization Daemon consists of the two or more of the following plug-ins depending on the Mainframe Security database that you are working with:

racf2ldap

The racf2ladp plug-in provides automatic outbound synchronization functionality from RACF to the LDAP server. Whenever a change is made in the RACF database, the plug-in detects the change and propagates the new data to the LDAP directory.

acf22ldap

The acf22ladp plug-in provides automatic outbound synchronization functionality from ACF2 to the LDAP server. Whenever a change is made in the ACF2 database, the plug-in detects the change and propagates the new data to the LDAP directory.

tss2ldap

The tss2ladp plug-in provides automatic outbound synchronization from Top Secret to the LDAP server. Whenever a change is made in the Top Secret database, the plug-in detects the change and propagates the new data to the LDAP server.

ldap2racf

The ldap2racf plug-in provides inbound synchronization. This plug-in is designed to allow you to update RACF fields.

ldap2acf2

The ldap2acf2 plug-in provides inbound synchronization. This plug-in is designed to allow you to update ACF2 fields.

ldap2tss

The ldap2tss plug-in provides inbound synchronization. This plug-in is designed to allow you to update TopSecret fields.

Installing and configuring the LDAP Bridge

System requirements

The following are the requirements for installing, configuring, and using the LDAP Bridge.

Software requirements

The LDAP Bridge requires the following elements:

- a version of IBM® z/OS® that is currently supported by IBM
- a version of CA Top Secret® that is currently supported by CA (when working with Top Secret)
- a version of CA ACF2™ that is currently supported by CA (when working with CA ACF2™).

Functional requirements

The LDAP Bridge runs under UNIX System Services (USS), and uses TCP/IP to communicate with remote clients. The LDAP Bridge makes use of LE run-time libraries, with C-language support.

The LDAP Bridge uses the R_admin interface to communicate with IBM RACF®, CA ACF2™, and CA Top Secret®.

Upgrading from a previous version of the LDAP Bridge

When upgrading from a previous version of the LDAP Bridge, you must perform a complete re-installation of the product. It is recommended that the new version be installed using a different HFS directory and different MVS data sets than the previous version. This will allow you to transfer any customizations to the JCLLIB members, LOADLIB members, SRCLIB members, ATTR data set, and HFS configuration files and scripts from the previous installation to the new installation.

Before you begin

Selecting the installation type

Before you install the LDAP Bridge, you must determine the type of installation that you require: single-system or multi-system. Multi-system installations allow you to share the file system where the product is installed between two or more z/OS® systems.

Single-system installation

The single-system installation option involves fewer steps and is appropriate when you plan to run the LDAP Bridge on one system, or when you plan to run the LDAP Bridge on multiple systems that do not share a file system. The single-system install process allows the LDAP Bridge directory structure to be simplified without experiencing naming conflicts.

You can perform single-system installation on many systems by cloning the installation to those systems. In order for this cloning to succeed, the specific values entered during the configure script, such as the path to the install directory and the port number, must be valid on the other systems where the LDAP Bridge will be installed.

Multi-system installation

If you plan to share file systems between two or more z/OS® systems where the LDAP Bridge is installed, you must perform multi-system installation. The multi-system configuration allows:

- the maintenance of a single installation of the LDAP Bridge rather than many separate installations
- segregation with respect to storage of the configuration, data and executable files used by the LDAP Bridge (the conf, data, logs and sbin directories).

The directories that the LDAP Bridge creates during installation differ slightly between a single-system installation and a multi-system installation. For information on the directories that are created during installation and configuration, refer to Directories created during installation and configuration.

To perform a multi-system installation, you must determine where the LDAP Bridge install directory will be. It must be valid for all systems in the installation.

Preparing your environment

You must prepare the following elements of your environment before installing the LDAP Bridge.

User IDs

The functions that are performed when installing, configuring, and running the LDAP Bridge can be divided into groups according to the user ID that performs each. These user IDs and the permissions they require are listed below. If there are no policy restrictions preventing it, any or all of these tasks can be performed under the same user ID, given the appropriate permissions.

- The user ID that is used to install and configure the LDAP Bridge (**Install ID**). This user ID must:
  - have an OMVS segment
  - be authorized to create directories in the HFS
  - have READ access to the BPX.FILEATTR.APF, BPX.FILEATTR.PROGCTL, and BPX.FILEATTR.SHARELIB profiles in the FACILITY class.
- The user ID that is used to submit the JCL to build the LDAP database (**Database Admin ID**). This user ID must:
  - be authorized to write to the HFS directories that are created by the install user
  - have an OMVS segment
  - be a member of the same group as the group owner of the HFS directories
  - be authorized to run the RACF IRRDBU00 utility to unload the RACF database
  - be authorized to run TSSCFILE to extract the ACID data from CA Top Secret®
  - be authorized to run the CA ACF2™ BACKUP command to extract records from ACF2 or must have access to a recent backup.
- The user ID that is used to submit the JCL to start the LDAP Bridge (**LDAP Bridge Admin ID**). This user ID must:

    - have an OMVS segment
    - be able to run the scripts and write to directories in the HFS
    - be able to submit commands through the R_admin interface.
- The user ID that is used by your LDAP client to connect to the LDAP database and administer the mainframe security database (LDAP Client Admin ID). This user must have:
    - an OMVS segment
    - the mainframe security database authority to run the set of commands that are needed by your LDAP client.

Configuring UNIX system services

The LDAP Bridge runs on the mainframe under UNIX System Services (USS). USS must be properly configured before you can install the LDAP Bridge. Before you install the LDAP Bridge, you must:

- be able to access USS using either OMVS or telnet
- allocate an HFS directory of sufficient size for the LDAP Bridge. The amount of disk space that is required for the directory can be determined using the following formula:
  required disk space = 400MB + (3.2 * size of mainframe security database)
- ensure that the parent directories of the LDAP Bridge have execute access permission for OTHER. For example, if the parent directory for the product is `/usr/lpp`, ensure that the both `/usr` and `/usr/lpp` have execute permission for OTHER. To view the permissions of the `/usr/lpp` directory, for example, issue the following command:
  `ls -ld /usr/lpp`

  To add execute permission for OTHER to `/usr/lpp`, for example, issue the following command:
  `chmod o+x /usr/lpp`
- ensure that the directory for the LDAP Bridge itself has appropriate permissions:
    - OWNER: read/write/execute
    - GROUP: read/write/execute
    - OTHER: execute.

If, for example, you are installing the LDAP Bridge into the `/usr/lpp/rlb35` directory, assign the appropriate permissions by issuing the following command:
`chmod 0771 /usr/lpp/rlb35`

- ensure that the owner of the LDAP Bridge directory (and its subdirectories) is the user ID under which the LDAP Bridge runs (either the batch job ID or the ID that is associated with the started task), or the group owner of the directory must be one of the groups that is associated with that ID.

For example, if the installation directory is `/usr/lpp/rlb35`, and you plan to run the LDAP Bridge under the SLAPD user ID, that is a member of the ADMIN and USERS group, then either the owner of the directory must be SLAPD or the group owner must be ADMIN or USERS.

NOTE: To see the owner and group owner of the `/usr/lpp/rlb35` directory, for example, issue the following command: `ls -ld /usr/lpp/rlb35`

To change the owner of this directory to SLAPD, that requires superuser authority, issue the following command:
`chown SLAPD /usr/lpp/rlb35`

To change the group owner of this directory to ADMIN, issue the following command:
`chgrp ADMIN /usr/lpp/rlb35`

Configuring your network

The LDAP Bridge communicates using TCP/IP. You must enable the following ports for TCP/IP access if you plan to use:

- unencrypted access for all or part of the application, enable a port for unencrypted access. Port 389 is the default, but you can use any port that works in your environment. If users from outside your firewall will be accessing the LDAP Bridge, you must modify your firewall to enable access to this port
- SSL access for all or part of the application, enable a port for SSL access. Port 636 is the default, but you can use any port that works in your environment. If users from outside your firewall will be accessing the LDAP Bridge, you must modify your firewall to enable access to this port.

Ensuring sufficient region size

LDAP Bridge processes run as submitted jobs or started tasks. All JCL and configuration parameters are delivered optimized for a 50,000 user installation. Under this configuration, all LDAP Bridge processes require a region size of approximately 400 megabytes.

The default REGION parameter that is coded in the JCL for the LDAP Bridge jobs is 0M. This indicates that no limit should be placed on the job's region size. However, at some sites, there are specific limitations that apply regardless of the REGION=0M parameter. These limitations, usually found in an IEFUSI installation exit, can be based on your user ID, job class, or other factors.

Verify with the system programmer that the job class and user ID under which you plan to run the LDAP Bridge can allocate a region size of 400 megabytes or more. If a process fails to allocate memory, it will exit with a return code 9. This indicates that the region size is too small and needs to be adjusted upwards.

Verify privileges

The LDAP Bridge executable must be APF authorized and program controlled to perform authentications against the mainframe security database that the Install ID is using (RACF, Top Secret, or ACF2). In order to enable the required APF-authorization, you must, at minimum, have READ access to the BPX.FILEATTR.APF profile in the FACILITY class.

Installation overview

The CD or downloaded version of the LDAP Bridge release media contains the compressed files `qslbv35fp5.pax.Z` and `openldap-2.4.11.pax.Z`. These files are used to install the LDAP Bridge into an HFS file system.

When you have expanded the `pax.Z` archives, the install directory contains subdirectories and the configuration script. The subdirectories are:

- backup
- bin
- conf
- data
- install
- lib
- logs
- sbin

The configure script prompts for certain variable values and then makes customized versions of the files from the install directory using the values input at the prompts. These customized files, along with the binaries and scripts, are placed in the `conf`, `data`, `sbin`, `bin` and `lib` directories. A log of the install process is placed in `logs`. During the configure script, a set of MVS data sets are created. Customized JCL and source members and load modules are copied into the data sets. Among the values input at the prompts are the high level qualifier for the MVS data sets, directory paths, port numbers, and system names that will be specific to the installation machine.

The multi-system configuration path names are referenced in this document. For information in the equivalent single-system directories refer to Directories created during installation and configuration.

Directories created during installation and configuration

The following directories are created and populated during installation and configuration of the LDAP Bridge. The directories vary slightly depending on whether you are doing a single-

system or multi-system installation.

In a multi-system installation, the directories that the LDAP Bridge creates during the installation (`conf`, `logs`, `sbin`, `bin`, `lib`, and `data`) each have a subdirectory with the system name entered during the configuration. These subdirectories are not present when the single-system installation is performed.

These system-named subdirectories hold the configuration files, logs, and data that are used by that system (for example, the binaries are contained in the sbin directory, and the system-specific subdirectory will contain the user customized/developed binaries.)

Table 1: System-named subdirectories

Contents	Single-system installation	Multi-system installation
Backups of existing configuration files, source files, data files, executable files, and load modules replaced by the configure script.	`installDirectory/backup/`	`installDirectory/backup/`
Binaries that are shipped with the LDAP Bridge.	`installDirectory/bin`	`installDirectory/bin/system Name/`
All of the configuration files for the system.	`installDirectory/conf/`	`installDirectory/conf/systemName/`
The certificates that are used for SSL.	`installDirectory/conf/certs/`	`installDirectory/conf/systemName/certs/`
A listing of all installed plugins.	`installDirectory/conf/plugins/`	`installDirectory/conf/plugins/`
The LDAP schema	`installDirectory/conf/schema/`	`installDirectory/conf/systemName/schema/`

Contents	Single-system installation	Multi-system installation
files.		
The customized versions of the configuration files, that are tailored at run-time.	`installDirectory/conf/tmp/`	`installDirectory/conf/systemName/tmp/`
All data files for the system.	`installDirectory/data/`	`installDirectory/data/systemName/`
All of the bdb databases that are used by the LDAP server.	`installDirectory/data/bdb/`	`installDirectory/data/systemName/bdb/`
The bdb database containing the plug-in configuration information.	`installDirectory/data/bdb/config/`	`installDirectory/data/systemName/bdb/config/`
The bdb database containing the audit log database.	`installDirectory/data/bdb/log/`	`installDirectory/data/systemName/bdb/log/`
The bdb database that holds the product version information.	`installDirectory/data/bdb/sb/`	`installDirectory/data/systemName/bdb/sb/`
The main bdb database that holds	`installDirectory/data/bdb/racf/`	`installDirectory/data/systemName/bdb/racf/`

Contents	Single-system installation	Multi-system installation
the RACF security database information.		
The LDIF files that are used to load the correspondi ng bdb databases.	`installDirectory/data/ldif/`	`installDirectory/data/systemName/ldif/`
The directories that are used by the Synchroniza tion Daemon when working with RACF.	`installDirectory/data/racf2lda p/`	`installDirectory/data/systemName/racf2l dap/`
The new files that are written by the security exit when working with RACF.	`installDirectory/data/racf2lda p/new/`	`installDirectory/data/systemName/racf2l dap/new/`
Files that have been successfully processed by the Synchroniza tion Daemon when working with RACF.	`installDirectory/data/racf2lda p/old/`	`installDirectory/data/systemName/racf2l dap/old/`
Files that have been	`installDirectory/data/racf2lda p/error/`	`installDirectory/data/systemName/racf2l dap/error/`

Contents	Single-system installation	Multi-system installation
unsuccessfully processed by the Synchronization Daemon when working with RACF.		
The main bdb database that holds the CA Top Secret® security database information.	`installDirectory/data/bdb/tss/`	`installDirectory/data/systemName/bdb/tss/`
The directories that are used by the Synchronization Daemon when working with Top Secret.	`installDirectory/data/tss2ldap/`	`installDirectory/data/systemName/tss2ldap/`
The new files that are written by the security exit when working with Top Secret.	`installDirectory/data/tss2ldap/new/`	`installDirectory/data/systemName/tss2ldap/new/`
Files that have been successfully processed	`installDirectory/data/tss2ldap/old/`	`installDirectory/data/systemName/tss2ldap/old/`

Contents	Single-system installation	Multi-system installation
by the Synchronization Daemon when working with Top Secret.		
Files that have been unsuccessfully processed by the Synchronization Daemon when working with Top Secret.	`installDirectory/data/tss2ldap/error/`	`installDirectory/data/systemName/tss2ldap/error/`
The main bdb database that holds the ACF2 security database information.	`installDirectory/data/bdb/acf2/`	`installDirectory/data/systemName/bdb/acf2/`
The directories that are used by the Synchronization Daemon when working with ACF2.	`installDirectory/data/acf22ldap/`	`installDirectory/data/systemName/acf22ldap/`
The new files that are written by the	`installDirectory/data/acf22ldap/new/`	`installDirectory/data/systemName/acf22ldap/new/`

Contents	Single-system installation	Multi-system installation
security exit when working with ACF2.		
Files that have been successfully processed by the Synchronization Daemon when working with ACF2.	`installDirectory/data/acf22ldap/old/`	`installDirectory/data/systemName/acf22ldap/old/`
Files that have been unsuccessfully processed by the Synchronization Daemon when working with ACF2.	`installDirectory/data/acf22ldap/error/`	`installDirectory/data/systemName/acf22ldap/error/`
Installation and configuration materials for the LDAP Bridge. Nothing in this directory should be modified.	`installDirectory/install/`	`installDirectory/install/`
Dynamic libraries that are shared by sbin and bin	`installDirectory/lib`	`installDirectory/lib/systemName/`

Contents	Single-system installation	Multi-system installation
binaries.		
All of the LDAP Bridge log files.	`installDirectory/logs/`	`installDirectory/logs/systemName/`
Executable files.	`installDirectory/sbin`	`installDirectory/sbin`
Any executable files that have been customized for the specific system. (This directory is typically empty.)	`N/A`	`installDirectory/sbin/systemName/`

All examples in this guide reflect the paths for a multi-system installation. Use the table above to determine the path for a single-system installation.

Installing the LDAP Bridge

To install the LDAP Bridge

1. Transfer the product media to the machine where you want to install the LDAP Bridge. Transfer `qslbv35fp5.pax.Z` and the `openldap-2.4.11.pax.Z` to your HFS directory using FTP. Specify binary mode for the FTP transfer.
2. Expand the `.pax.Z` files that were on the product media. To do this, issue the following commands for each `.pax.Z` file that you received:

```
cd installDirectory

pax -rv -px -f <path-to-pax-file-name>/qslbv35fp5.pax.Z
```

Configuring the LDAP Bridge

Run the configuration script to configure the LDAP Bridge. The script performs the following tasks:

- prompts you for the site-specific variables and records the values in the site.variables file
- customizes the JCL and configuration files
- allocates the EXITLIB, SRCLIB, LOADLIB, JCLLIB, and ATTR data sets
- copies the JCL and source members, load modules, and attributes file from the HFS to MVS data sets
- configures the LDAP server and configuration database along with the Synchronization Daemon, LDAP Command Translator.

Running the configuration script

The configuration script configures your LDAP Bridge installation. The first time that the configuration script is run, you are prompted for site-specific information that is used to create a configuration file. The configuration script can be run as many times as necessary. Whenever the configuration script is re-run, the script deletes the previous files and creates new ones based on the new information that is provided.

During the running of the configuration script, backups are made of any modified files. At the end of the script, a prompt gives the total size of the backup files and offers to delete them. The size of these backups is not significant on the initial run of the script, but if the configuration script is re-run, the backups could take up a significant amount of disk space. At the beginning of any run of the configuration script, a check is made to see if the installDirectory/backup directory takes up more than 1MB of space. If so, a prompt offers to delete the old backup files before taking the new backup.

How to run the configuration script

1. Gather the following site-specific information:
    - do you want to perform a single-system or multi-system configuration?
    - the name of the system that you want to configure. If you choose to perform a multi-system installation, you can perform configuration for the current system or a different system. The default is the system name that was discovered by the LDAP Bridge configuration script
    - HFS root directory
    - MVS data set high level qualifier(s)
    - permanent data set unit
    - temporary data set unit
    - LDAP root
    - LDAP server port
    - LDAP server SSL port
    - security database locale.

2. Run the configuration script. To do this, issue the following commands from an OMVS or telnet command prompt:

```
cd installDirectory

sh ./configure
```

3. Respond to the configure script prompts as appropriate for your environment, using the information that you gathered in step 1.

   NOTE: Pressing **Enter** for a particular query results in the default value being used for that variable. Some variables do not have default values.

When you are finished, a message is displayed to indicate the successful completion of the installation script.

Configuring the SMF installation exits when working with IBM RACF®

The synchronization daemon runs as a stand-alone UNIX daemon in a separate address space from the LDAP server. It reads the SMF records that are generated whenever RACF changes are made, and propagates the changes to the LDAP Bridge using LDAP. The SMF records are written to the `installDirectory/data/systemName/racf2ldap/new` directory by SLAPU83, a program that runs in the SMF IEFU83 exit point.

To use the synchronization daemon, you must activate the SMF installation exits described below.

Enabling the IEFU83 installation exit points

Before implementing the IEFU83 installation exit program, ensure that installation exit points are enabled on your system for the following environments:

- SYSSTC.IEFU83 installation exit point
- SYSTSO.IEFU83, SYSJES2.IEFU83, or SYS.IEFU83 installation exit point. The installation exit point that the LDAP Bridge requires varies depending on your system configuration:
  - if TSO is defined as a separate SMF subsystem, use the SYSTSO.IEFU83 installation exit point
  - if JES2 is defined as a separate SMF subsystem, use the SYSJES2.IEFU83 installation exit point
  - if neither TSO nor JES are defined as separate SMF subsystems, use the SYS.IEFU83 installation exit point.

The procedure for enabling SYSSTC.IEFU8, SYSTSO.IEFU83, SYSJES2.IEFU83, and SYS.IEFU83 is described in the IEFU83 section of the *IBM z/OS MVS Installation Exits Manual*.

To enable the required exit points

1. Edit the SMFPRMnn member of the SYS1.PARMLIB data set, where nn is the SMF parameter member that is currently active on your system.
2. Verify that IEFU83 is specified in the EXITS clause of the SUBSYS(STC) parameters. For example:

   `SUBSYS(STC,EXITS(IEFU83,xxx))`
   where xxx represents other keywords and parameters used in your environment.
3. If TSO is defined as a separate SMF subsystem, verify that IEFU83 is specified in the EXITS clause of the SUBSYS(TSO) parameters. For example:

   `SUBSYS(TSO,EXITS(IEFU83,xxx))`
   where xxx represents other keywords and parameters used in your environment.
4. If JES2 is defined as a separate SMF subsystem, verify that IEFU83 is specified in the EXITS clause of the SUBSYS(JES2) parameters. For example:

   `SUBSYS(JES2,EXITS(IEFU83,xxx))`
   where xxx represents other keywords and parameters used in your environment.
5. If neither TSO nor JES2 are defined as a separate SMF subsystem, verify that IEFU83 is specified in the EXITS clause of the SYS parameters. For example:

   `SYS(xxx,EXITS(IEFU83,xxx)xxx )`
   where xxx represents other keywords and parameters used in your environment.

Activating the IEFU83 dynamic installation exit program

You must activate the IEFU83 installation exit. The procedure for activating a dynamic IEFU83 installation exit program is described in the IEFU83 section of the *IBM z/OS MVS Installation Exits Manual*.

Activating SLAPU83

You must activate the SLAPU83 program. You can activate it dynamically (temporarily) or permanently.

Activating SLAPU83 dynamically

The SLAPU83 program can be installed dynamically (temporarily), for testing, from the system console with the following commands:

```
SETPROG EXIT,ADD,EXITNAME=SYS\STC.IEFU83,MODNAME=SLAPU83,DSNAME=HLQ.LOADLIB
```

and one of the following:

```
SETPROG EXIT,ADD,EXITNAME=SYSTSO.IEFU83,MODNAME=SLAPU83, DSNAME=HLQ.LOADLIB
```

– OR –

```
SETPROG EXIT,ADD,EXITNAME=SYSJES2.IEFU83,MODNAME=SLAPU83, DSNAME=HLQ.LOADLIB
```

– OR –

```
SETPROG EXIT,ADD,EXITNAME=SYS.IEFU83,MODNAME=SLAPU83, DSNAME=HLQ.LOADLIB
```

Activating installation exit points using these commands remains in effect only until the next IPL.

Activating SLAPU83 permanently

To install the SLAPU83 program permanently

1. Edit the PROGnn member of the SYS1.PARMLIB data set, where *nn* is the program parameter member currently active on your system.
2. Add the following statements:

   ```
   EXIT ADD
   EXITNAME(SYSSTC.IEFU83)
   MODNAME(SLAPU83)
   STATE(ACTIVE)
   DSNAME(HLQ.LOADLIB)
   ```

   and one of the following:

   ```
   EXIT ADD
   EXITNAME(SYSTSO.IEFU83)
   MODNAME(SLAPU83)
   STATE(ACTIVE)
   DSNAME(HLQ.LOADLIB)
   ```

   – OR –

   ```
   EXIT ADD
   EXITNAME(SYSJES2.IEFU83)
   MODNAME(SLAPU83)
   STATE(ACTIVE)
   ```

```
DSNAME(HLQ.LOADLIB)
```

– OR –

```
EXIT ADD
EXITNAME(SYS.IEFU83)
MODNAME(SLAPU83)
STATE(ACTIVE)
DSNAME(HLQ.LOADLIB)
```

Alternatively, you can copy SLAPU83 from HLQ.LOADLIB to the LPALIB, in which case you can omit the DSNAME statement in the above example.

After the PROGnn member has been edited in SYS1.PARMLIB, it may have to be activated by editing the COMMNDnn member to include the following statement:

```
COM='SET PROG=nn'
```

where nn corresponds to the suffix for the PROGnn member.

Setting the IBM RACF® system options (SETROPTS)

To ensure that the LDAP Bridge database is always synchronized with IBM RACF®, several RACF system options must be enabled by issuing the following command:

```
SETROPTS AUDIT(*) SAUDIT OPERAUDIT
```

where the:

- AUDIT(*) parameter instructs RACF to create SMF records whenever any RACF profiles are added, modified, or deleted. Without these SMF records, the racf2ldap plug-in cannot propagate RACF changes to the LDAP Bridge.
- SAUDIT parameter instructs RACF to create SMF records whenever RACF profiles are changed by administrators with the SPECIAL and GROUP-SPECIAL attributes. Without these SMF records, the racf2ldap plug-in cannot propagate RACF changes made by these administrators to the LDAP Bridge.
- OPERAUDIT parameter instructs RACF to create SMF records whenever RACF profiles are changed by administrators with the OPERATION attribute. Without these SMF records, the racf2ldap plug-in cannot propagate RACF changes made by these administrators to the LDAP Bridge.

These commands do not cause RACF to audit violations or access attempts involving these profiles. They instruct RACF to audit administrative changes. These changes generate a small amount of SMF activity and will not have a significant impact on the performance or size of your SMF data sets.

Configuring the CA Top Secret® installation exit

To use the Synchronization Daemon, you must activate the CA Top Secret® installation exit, TSSINSTX. The configure script assembles and link-edits this exit into HLQ.LOADLIB (TSSINSTX). If you do not already have a version of TSSINSTX installed, you must install this version. If you already have a version of TSSINSTX installed, you must integrate the LDAP Bridge version of the exit into the one that currently exists on your system, as described below.

The Synchronization Daemon runs as a stand-alone UNIX daemon in a separate address space from the LDAP Bridge. It reads the change records that are generated whenever Top Secret changes are made, and propagates the changes to the LDAP Bridge using LDAP. The change records are written to the `installDirectory/data/systemName/tss2ldap/new` directory by the TSSINSTX program that runs as a Top Secret installation exit.

Activating TSSINSTX

Activating TSSINSTX dynamically

To activate this exit dynamically

1. Copy HLQ.LOADLIB(TSSINTSX) to a link-listed library.
2. Refresh the link list by issuing the following command from the operator console:
   `F LLA,REFRESH`
3. Issue the following command from the operator console to temporarily activate the exit:
   `F TSS,EXIT(ON)`

   NOTE: You can also temporarily activate the exit by issuing the following command from the TSO command line: `TSS MODIFY(EXIT(ON))`

When you have tested the exit, you must permanently activate it.

Activating TSSINSTX permanently

To activate this exit permanently

1. Edit SYS1.PARMLIB(TSSPARM0)
2. Add the following statement:
   `EXIT(ON)`

The change will take effect at the next IPL or when CA Top Secret® is restarted.

Integration with an existing version of TSSINSTX

If you are running a site-specific, customized version of TSSINSTX, then the LDAP Bridge version of TSSINSTX must be integrated into your version.

To integrate a customized version of TSSINSTX with the LDAP Bridge version

1. Customize the JCLLIB(RQXASMEV) member (as documented in the member).
2. Submit the JCLLIB(RQXASMEV) job.
3. Copy the resulting LOADLIB(TSSINSTX) module to a link-listed library.
4. Activate the CA Top Secret® Site Installation Exit by following the steps in Activating TSSINSTX permanently.

   NOTE: You can also integrate the LDAP Bridge version of TSSINSTX by running the `installDirectory/install/plugins/t2l/bin/asmevt` script, following the procedure documented in the script

Configuring the SMF installation exits when working with CA ACF2™

The Synchronization Daemon runs as a stand-alone UNIX daemon in a separate address space from the LDAP Bridge. It reads the SMF records that are generated whenever CA ACF2™ changes are made, and propagates the changes to the LDAP Bridge using LDAP. The SMF records are written to the `installDirectory/data/systemName/acf22ldap/new` directory by SLAPU83A, a program that runs in the SMF IEFU83 exit point.

To use the Synchronization Daemon, you must activate the SMF installation exits described below.

Prerequisites to activating IEFU83

Before implementing the IEFU83 installation exit program, ensure that installation exit points are enabled on your system for the following environments:

- Started Tasks - SYSSTC.IEFU83 installation exit point
- SYSTSO.IEFU83, SYSJES2.IEFU83, or SYS.IEFU83 installation exit point. The installation exit point that the LDAP Bridge requires varies depending on your system configuration:
  - if TSO is defined as a separate SMF subsystem, use the SYSTSO.IEFU83 installation exit point

- if JES2 is defined as a separate SMF subsystem, use the SYSJES2.IEFU83 installation exit point
- if neither TSO nor JES are defined as separate SMF subsystems, use the SYS.IEFU83 installation exit point.

The procedure for enabling SYSSTC.IEFU8, SYSTSO.IEFU83, SYSJES2.IEFU83, and SYS.IEFU83 is described in the IEFU83 section of the *IBM z/OS MVS Installation Exits Manual*.

To enable the required exit points

1. Edit the SMFPRMnn member of the SYS1.PARMLIB data set, where nn is the SMF parameter member currently active on your system.
2. Verify that IEFU83 is specified in the EXITS clause of the SUBSYS(STC) parameters. For example:

   ```
   SUBSYS(STC,EXITS(IEFU83,xxx))
   ```
   where xxx represents other keywords and parameters used in your environment.
3. Verify that IEFU83 is specified in the EXITS clause parameters. This is only required when TSO is defined as a separate SMF subsystem, then this member contains a statement starting with "SUBSYS(TSO)".
   For example:

   ```
   SUBSYS(TSO,EXITS(IEFU83,xxx))
   ```
   where xxx represents other keywords and parameters used in your environment.
4. Verify that IEFU83 is specified in the EXITS clause parameters. This is only required when JES2 is defined as a separate SMF subsystem, then this member contains a statement starting with "SUBSYS(JES2)". For example:

   ```
   SUBSYS(JES2,EXITS(IEFU83,xxx))
   ```
   where xxx represents other keywords and parameters used in your environment.
5. Verify that IEFU83 is specified in the EXITS clause parameters for the SYS statement. This is only required when neither TSO nor JES2 are defined as separate SMF subsystems. For example:

   ```
   SYS(xxx,EXITS(IEFU83,xxx)xxx )
   ```
   where xxx represents other keywords and parameters used in your environment.

Activating IEFU83

The procedure for activating a dynamic IEFU83 installation exit program is described in the IEFU83 section of the *IBM z/OS MVS Installation Exits Manual*.

Activating SLAPU83A dynamically

The SLAPU83 program can be installed temporarily, for testing, from the system console with the following commands:

```
SETPROG EXIT,ADD,EXITNAME=SYSSTC.IEFU83,MODNAME=SLAPU83A,DSNAME=HLQ.LOADLIB
```

and either:

```
SETPROG EXIT,ADD,EXITNAME=SYSTSO.IEFU83,MODNAME=SLAPU83A,DSNAME=HLQ.LOADLIB
```

– OR –

```
SETPROG EXIT,ADD,EXITNAME=SYSJES2.IEFU83,MODNAME=SLAPU83A,DSNAME=HLQ.LOADLIB
```

– OR –

```
SETPROG EXIT,ADD,EXITNAME=SYS.IEFU83,MODNAME=SLAPU83A,DSNAME=HLQ.LOADLIB
```

where `HLQ` is the high-level qualifier you created for the LDAP Bridge.

Whether to use the command to activate SYSTSO.IEFU83, SYSJES2.IEFU83, or SYS.IEFU83 depends on whether you have TSO or JES2 defined as separate SMF subsystems in your SMF parameter file:

- if TSO is defined as a separate SMF subsystem, use the command that references SYSTSO.IEFU83
- if JES2 is defined as a separate SMF subsystem, use the command that references SYSJES2.IEFU83
- if neither TSO nor JES2 are defined as separate SMF subsystems, use the command that references SYS.IEFU83.

Activating installation exit points using these commands remains in effect only until the next IPL.

Activating SLAPU83A permanently

To install the SLAPU83A program permanently

1. Edit the PROGnn member of the SYS1.PARMLIB data set, where nn is the program parameter member currently active on your system.
2. Add the following statements:

   ```
   EXIT ADD
   EXITNAME(SYSSTC.IEFU83)
   MODNAME(SLAPU83A)
   STATE(ACTIVE)
   DSNAME(HLQ.LOADLIB)
   ```

   and one of the following:

   ```
   EXIT ADD
   ```

```
EXITNAME(SYSTSO.IEFU83)
MODNAME(SLAPU83A)
STATE(ACTIVE)
DSNAME(HLQ.LOADLIB)
```

– OR –

```
EXIT ADD
EXITNAME(SYSJES2.IEFU83)
MODNAME(SLAPU83A)
STATE(ACTIVE)
DSNAME(HLQ.LOADLIB)
```

– OR –

```
EXIT ADD
EXITNAME(SYS.IEFU83)
MODNAME(SLAPU83A)
STATE(ACTIVE)
DSNAME(HLQ.LOADLIB)
```

Where HLQ is the high-level qualifier you created for the LDAP Bridge. Alternatively, move SLAPU83A from HLQ.LOADLIB to the LPALIB; in this case you can omit the DSNAME statement in the above example.

Whether you use the statements to activate SYSTSO.IEFU83. SYSJES2.IEFU83, or SYS.IEFU83 depends on whether you have TSO or JES defined as separate SMF subsystems in your SMF parameter file if:

- TSO is defined as a separate SMF subsystem, use the statements that reference SYSTSO.IEFU83
- JES2 is defined as a separate SMF subsystem, use the statements that reference SYSJES2.IEFU83
- neither TSO nor JES2 are defined as separate SMF subsystems, use the statements that reference SYS.IEFU83.

Once the PROGnn member has been edited in SYS1.PARMLIB, activate it by editing the COMMNDnn member to include the following statement:
`COM='SET PROG=nn'`

where nn corresponds to the suffix for the PROGnn member.

Loading the LDAP directory

The LDAP Bridge uses a directory database which contains data from your mainframe security repositories. After the LDAP Bridge database is initially loaded, the LDAP Bridge Synchronization Daemon keeps the databases synchronized.

Running the LDAP Bridge

This chapter describes how to start, stop and test the LDAP Bridge. You can run the LDAP Bridge as a z/OS batch job or started task using BPXBATCH.

Starting the LDAP Bridge

Whether you run the LDAP Bridge as a started task or a submitted job, you must use the LDAP Bridge Admin ID to start the LDAP Bridge. For more information on user IDs, refer to User IDs.

Submitted jobs

For testing purposes, it is recommended that you start the LDAP Bridge as a submitted job. Add job card information to the START member of HLQ.JCLLIB data set, then submit the job. All condition codes return as zero. The RQXSTART job runs until the RQXSTOP job is submitted to stop the LDAP Bridge.

Started tasks

To create a started task that starts the LDAP Bridge, customize the RQXTASK JCL that is provided in the HLQ.JCLLIB data set.

Starting the synchronization daemon

The synchronization daemon starts automatically using the same RQXSTARTJCL that is used to start the LDAP Bridge. Whenever you start the LDAP Bridge, the synchronization daemon is also active.

REGION parameter

Setting the REGION parameter of the RQXSTART JCL to REGION=0M is recommended so that there is no limit on storage and the LDAP Bridge can acquire as much storage as it needs. As delivered, the LDAP Bridge requires approximately 400MB of storage. If your site restricts the amount of storage available for various jobs or initiators, you must make certain to run the LDAP Bridge in an initiator that permits sufficient storage.

However, specifying REGION=0M does not always guarantee sufficient memory. Refer to Ensuring sufficient region size for further information on allocating a sufficient region size.

TIME parameter

Setting the TIME parameter of the RQXSTART JCL to TIME=NOLIMIT is recommended so that there is no preset time limit on how long the LDAP Bridge can run. Without this parameter, the LDAP Bridge eventually abends with a system code of 522. If your site restricts the amount of time available for various jobs or initiators, you must ensure that the LDAP Bridge is run in a class that permits no time restrictions.

Stopping the LDAP Bridge

You can stop the LDAP Bridge, either

- with the RQXSTOP member of the JCLLIB data set. Add job card information to the JCL, then submit the job. All condition codes return as zero, or
- by issuing the `dostop` command from a UNIX command prompt, or
- with the MVS STOP command. At startup, a message is written to the slapd.log file and to the MVS system log showing the job name and ASID to specify on the command to issue, for example:

  ```
  /P RQXSTART,A=7A
  ```

NOTE: You must specify the correct ASID, as indicated in the startup message in order for the MVS STOP command to succeed.

Testing the LDAP Bridge

Test the LDAP Bridge by running the dotestserver script as described below.

Verifying that the LDAP server is running

To verify that the LDAP server is running

1. Enter OMVS from TSO.
2. Enter the following commands:

   ```
   cd /installDirectory/sbin
   ```

   ```
   dotestserver
   ```

3. At the prompts, enter your mainframe security database user ID and password. This test returns information on your security database user ID as stored in the LDAP repository.

Testing the synchronization daemon with IBM RACF®

To test the synchronization daemon with IBM RACF®

1. Verify that the RACF Exit program is enabled and start the LDAP Bridge if it is not already running.
2. From TSO, issue the following command:

   ```
   ALTUSER(testuserID) NAME('RACF2LDAP TEST')
   ```

   where `testuserID` is any valid RACF user ID.
3. Wait briefly, enter OMVS from TSO.
4. Enter the following commands:

   ```
   cd /installDirectory/sbin
   ```

   ```
   dotestr2l
   ```

5. At the prompts, enter your RACF user ID and password along with `testuserID`. This test returns the distinguished name of the entry along with the following text:

   ```
   cn= RACF2LDAP TEST
   ```

   If you do not receive this result, consult `installDirectory/logs/systemName/racf2ldap.log` to determine the cause of the error.

Testing the synchronization daemon with CA Top Secret®

To test the synchronization daemon with CA Top Secret®

1. Verify that the TSSINSTX program is enabled and start the LDAP Bridge if it is not already running.
2. From TSO, issue the following command:

   ```
   TSS REPLACE(testuserID) NAME('TSS2LDAP TEST')
   ```

   where `testuserID` is any valid Top Secret user ID.
3. Wait briefly, enter OMVS from TSO.
4. Enter the following commands:

   ```
   cd /installDirectory/sbin
   dotestt2l
   ```
5. At the prompts, enter your mainframe user ID and password along with `testuserID`. This test returns the distinguished name of the entry along with the following text:

   ```
   cn: TSS2LDAP TEST
   ```

   If you do not receive this result, consult `installDirectory/logs/systemName/tss2ldap.log` to determine the cause of the error.

Testing the synchronization daemon with CA ACF2™

To test the synchronization daemon with CA ACF2™

1. Verify that the CA ACF2™ Exit program is enabled and start the LDAP Bridge if it is not already running.
2. From TSO, issue the following command:

   ```
   ACF
   CHANGE testuserID NAME('ACF22LDAP TEST')
   ```

   where `testuserID` is any valid ACF2 user ID.
3. Wait briefly, enter OMVS from TSO.
4. Enter the following commands:

   ```
   cd /installDirectory/sbin
   dotesta2l
   ```
5. At the prompts, enter your ACF2 user ID and password along with `testuserID`. This test should return the distinguished name of the entry along with the following text:

   ```
   cn=ACF22LDAP TEST
   ```

   If you do not receive this result, consult `installDirectory/logs/systemName/acf22ldap.log` to determine the cause of the error.

Testing IBM RACF® administration from an LDAP client

To test IBM RACF® administration from an LDAP client

1. Verify that the LDAP Bridge is running.
2. Enter OMVS from TSO.
3. Enter the following commands:

   ```
   cd /installDirectory/sbin

   dotestl2r
   ```

4. At the prompts, enter your RACF user ID and password along with a new user ID that will be created on your RACF database. Your user ID must have sufficient authority in RACF to create a user in order to complete this step. When complete, the LDAP information for the new RACF user ID that was created will be returned.

Testing CA Top Secret® administration from an LDAP client

To test CA Top Secret® administration from an LDAP client

1. Verify that the LDAP Bridge is running.
2. Enter OMVS from TSO.
3. Enter the following commands:

   ```
   cd /installDirectory/sbin

   dotestl2t
   ```

4. At the prompts, enter your Top Secret user ID and password along with a new user ID that will be created on your Top Secret database. Your Top Secret user ID must have sufficient authority in Top Secret to create a user in order to complete this step. When complete, the LDAP information for the new Top Secret user ID that was created will be returned.

Testing CA ACF2™ administration from an LDAP client

To test CA ACF2™ administration from an LDAP client

1. Verify that the LDAP Bridge is running.
2. Enter OMVS from TSO.
3. Enter the following commands:

   ```
   cd /installDirectory/sbin
   ```

   ```
   dotestl2a
   ```

4. At the prompts, enter your ACF2 user ID and password along with a new user ID that will be created in your ACF2 database. In order to complete this step, your ACF2 user ID must have sufficient authority in ACF2 to create a user. When complete, the LDAP information for the newly created ACF2 user ID will be returned.

Tuning the LDAP Bridge

This chapter describes how to tune the LDAP Bridge. You can use the LDAP Bridge without tuning it. However, by tuning it, you can make changes to the default operations of the LDAP Bridge.

Logging

The following types of logging are available with the LDAP Bridge:

Audit logging

Audit logging logs activity from LDAP clients into a new LDAP database located in the `installDirectory/data/systemName/bdb/log/` directory. A new entry is created in the log database for every logged action. The entries are identified by timestamp, and have attributes that describe the action that was taken.

There are performance considerations contributing to the decision whether or not to enable audit logging. When audit logging is enabled there is a relatively small impact on CPU overhead and an impact on disk space for the log database. The disk space that your log database requires depends on the amount of activity that your LDAP clients generate.

When you are using the ldap2racf, ldap2acf2, or ldap2tss plug-in, the LDAP Bridge client changes that are intercepted and redirected to the external security manager are logged in the audit log even though no direct changes are made to the LDAP directory. When the resulting changes in the external security manager, if any, are synchronized back to LDAP by the ldap2racf, ldap2acf2, or ldap2tss plug-in, those changes are separately logged.

When you have audit logging enabled, you can specify:

- the location of the audit log database
- who can access the audit log database
- the types of operations that are logged (`abandon`, `add`, `bind`, `compare`, `delete`, `extended`, `modify`, `modrdn`, `search`, and `unbind`)
- a filter for matching against Deleted and Modified entries so that if the entry matches the filter, the old contents of the entry will be logged along with the current request
- the maximum length of time that log entries are retained in the database
- the frequency at which the database will be scanned for old entries
- whether you want all requests or only successful requests to be recorded.

Review the contents of slap.accesslog.conf, where there are verbose comments describing all the options available for audit logging.

Enabling audit logging

To enable audit logging

1. Edit `slapd.mainframesecuritydatabase.conf` and un-comment the following `#include` line (where mainframesecuritydatabase is the mainframe security database that you are working with):

   ```
   # The accesslog overlay provides audit log functionality.

   #include %filedir%/slapd.accesslog.conf
   ```

2. Restart the LDAP Bridge.

   When audit logging is initially enabled, it uses the following default settings:

- the root entry of the log database is cn=log
- the root DN of the log database is cn=logManager,cn=log with password = secret
- only write operations (add, delete, modify, modrdn) are logged
- "Before" images of all modified entries are saved in the log entries
- entries in the log database are purged after 30 days.

  You can change these default settings by editing the following files:

- slapd.conf

- slapd.acesslog.db.conf
- slapd.acesslog.conf

You can tune the performance of the log database using the DB_CONFIG.log file. See Tuning the LDAP Bridge for data recoverability and durability for details on how to tune the DB_CONFIG parameters.

Activity logging

Activity logging logs the activity on the LDAP server. The logging information is written to the `installDirectory/logs/systemName/slapd.log` file during the operation of the LDAP server.

Setting the LDAP Bridge logging Level for activity logging

You can set the logging level using the DEBUG parameter that is found in the RQXSTART JCL. The logging level cannot be changed once the LDAP Bridge is started. To change the logging level, stop the LDAP Bridge, make the required changes, then restart the LDAP Bridge.

The following table describes the debugging levels:

Table 2: Debugging levels

Debug parameter setting	Type of trace performed
DEBUG= -1	Enable all debugging.
DEBUG= 1	Trace function calls.
DEBUG= 2	Trace function handling.
DEBUG= 4	Display all processing.
DEBUG= 8	Trace connections and results.
DEBUG= 16	Display packets being sent and received.
DEBUG= 32	Trace search filter processing.
DEBUG= 64	Display configuration parameters.
DEBUG= 128	Trace access control list processing.
DEBUG= 256	Trace connections/operations/results.
DEBUG= 512	Trace entries sent.

Debug parameter setting	Type of trace performed
DEBUG= 1024	Trace shell backend processing.
DEBUG= 2048	Trace entry parsing.

Some of the loglevels result in extremely large log files and are intended to be used in a diagnostic, rather than a production, scenario. It is recommended that you contact technical support for advice when changing the RQXSTART loglevels.

To enable multiple debugging levels, add the various individual DEBUG parameter settings together. For example, to trace function calls (DEBUG=1) and display configuration parameters (DEBUG=64), set the debugging level to DEBUG=65.

Logs are stored in a file called slapd.log. Every day at midnight the logging is redirected to a new file with the date embedded in the name as follows: slapd.log.*date*. To keep the logs directory from filling up the disk, a cleanup script is provided that is started when the LDAP Bridge starts and runs until the LDAP Bridge is shut down. It is located in `installDirectory/sbin/dopurge`. You can specify the number of days to keep a log file and the file mask including the directory containing the file in the config file `installDirectory/conf/systemName/dopurge.conf`

The default settings of dopurge.conf are:

```
30 %logdir%/*.log.*

30 %datadir%/bdb/%secs%/log.*
```

This means that:

- all files older than 30 days containing the string '.log.' in the logs/systemName directory will be purged
- all files older than 30 days beginning with the string 'log.' in the data/systemName/bdb/%secs% directory will be purged.

LDAP server configuration files

Managing archived IBM RACF®, CA Top Secret® and CA ACF2™ changes

While archiving SMF records provides a useful resource for debugging purposes, you must ensure that the archive is periodically purged so that your HFS system does not run out of space. To accomplish this task, you must set the RETAIN parameter.

To set the RETAIN parameter

The racf2ldap.conf, tss2ldap.conf and acf22ldap.conf configuration files contain the parameters that control the operation of the synchronization daemon. In these files, the RETAIN parameter determines how SMF records are archived by the synchronization daemon.

- racf2ldap.conf controls how IBM RACF® SMF records are archived by the synchronization daemon for the racf2ldap plug-in
- tss2ldap.conf controls how CA Top Secret® SMF records are archived by the synchronization daemon for the tss2ldap plug-in
- acf22ldap.conf controls how CA ACF2™ SMF records are archived by the synchronization daemon for the acf22ldap plug-in

To set the RETAIN parameter

1. Open the required file, located in:

   ```
   installDirectory/conf/systemName/
   ```

2. Set the RETAIN parameter to the appropriate setting:
   - 0 = SMF records are written to plug-in/old and are not deleted, where plug-in is the name of the LDAP Bridge plug-in that you are working with. For example, records for the tss2ldap plug-in are stored in tss2ldap/old.
   - -1 = SMF records are deleted once they are processed and are not written to plug-in/old.
   - nn = SMF records are written to plug-in/old and records older than nn days are deleted where nn is a number between 0 and 999.

Encryption (SSL/TLS with mutual authentication and CRL checking)

The LDAP Bridge uses OpenLDAP V2.4 LDAP server and OpenSSL. The LDAP Bridge supports encrypted LDAP communications using the Secure Sockets Layer (SSL) and its successor, Transport Layer Security (TLS). You can choose to use SSL/TLS with or without Mutual Authentication and with or without CRL checking.

NOTE: CRL checking requires that the CRLs must be located on the file system where the LDAP Bridge is located. The CRLs can either be concatenated to the CA certificate file in a USS directory or MVS data set, or they can be present in USS with the CA certificate as standalone files.

The locations of the certificates, keys, and CRLs that the LDAP server reads at startup time are stored in the slapd.conf configuration file.

Performance implications

Encrypting all LDAP communications increases resource utilization and response times, often by more than 100%. This is especially noticeable and detrimental for high-volume authentication and authorization applications. Even with hardware acceleration, the SSL/TLS handshake and key exchange is subject to network latency and a variety of other performance factors that will increase response time.

Samples

Sample certificates and keys, are provided for testing purposes. These files are located in `installDirectory/conf/systemName/certs`. The default values that are used in slapd.conf correspond to the sample files. In addition to these defaults, by default, in slapd.conf, client verification is allowed but not required, and CRL checking is not specified.

The following sample files are provided:

Table 3: Sample files

Name	Description
ca_cert.pem	Sample CA certificate for testing SSL/TLS only with a CRL listing revoked_cert.pem as revoked for CRL testing
server_cert.pem	Sample server certificate for testing SSL/TLS only
server_key.pem	Sample server key for testing SSL/TLS only
ldapusr_cert.pem	Sample client certificate for testing SSL/TLS with Mutual Authentication from the `installDirectory/sbin/dotestserver` script
ldapusr_key.pem	Sample client key for testing SSL/TLS with Mutual Authentication from the `installDirectory/sbin/dotestserver` script
revoked_cert.pem	Sample client certificate listed as revoked in the sample CRLs listed above. For testing SSL/TLS with Mutual Authentication with CRL from the `installDirectory/sbin/dotestserver`

Name	Description
	script
revoked_key.pem	Sample client key for the revoked certificate in the sample CRLs listed above. For testing SSL/TLS with Mutual Authentication with CRL from the `installDirectory/sbin/dotestserver` script

Encryption prerequisite

In order to use SSL/TLS encryption, you must specify an encrypted port. To specify the port, edit the `installDirectory/conf/systemName/site.variables` file and change the sslport parameter to the required port. By default, the LDAP port for encrypted communications is 636. If you want to use a port other that 636, select an unreserved port that is available on the host that is running the LDAP Bridge. The SSL/TLS port value in site.variables is read by the LDAP Bridge at startup and used for that session.

Configuring encryption

There are many options when configuring encryption to suit your needs. Configuration is controlled in the slapd.conf file. To change your encryption configuration you must edit slapd.conf and restart the LDAP Bridge.

Configuring mutual authentication

To configure mutual authentication

1. Upload the set of certificates, keys, and CRLs for production use to your certificates directory.
2. Edit `installDirectory/conf/systemName/slapd.conf` to reflect the names of the production certificates.
3. In slapd.conf enable **Mutual Authentication**.
4. Restart the LDAP Bridge.
5. Test the SSL/TLS settings certificates, keys, and CRLs, in the controlled environment of the LDAP Bridge directory structure, using the dotestserver script that is supplied in `installDirectory/sbin`.

Configuring CRL checking

To configure CRL checking

1. Upload the set of certificates, keys, and CRLs for production use to your certificates directory.
2. Edit `installDirectory/conf/systemName/slapd.conf` to reflect the names of the production certificates.
3. In slapd.conf add a new directive to enable CRL checking.
4. Restart the LDAP Bridge.
5. Test the SSL/TLS settings certificates, keys, and CRLs, in the controlled environment of the LDAP Bridge directory structure, using the dotestserver script that is supplied in `installDirectory/sbin`.

Testing

By default, the `installDirectory/sbin/dotestserver` script is set up to test a non-encrypted connection. You can configure the LDAP Bridge to test Mutual Authentication and CRL checking by configuring a file called .ldaprc that is located in the OMVS home directory of the user that is running dotestserver.

The following examples demonstrate settings for various SSL/TLS tests using the sample certificates, keys and CRL supplied with the LDAP Bridge.

Testing SSL/TLS

To test SSL\TLS

1. In the OMVS home directory for the userID that is running the test, create a file called **.ldaprc**
2. This file is referenced in the prompts of the dotestserver script. It provides the location of the client certificate and key for the script to present to the LDAP server. When this file is present, the dotestserver script operates in SSL mode.
3. Populate the .ldaprc file as appropriate for your site. The following sample .ldaprc contents correspond to the sample certificates and keys provided.

```
TLS_CACERT installDirectory/conf/systemName/certs/ca_cert.pem

TLS_CERT installDirectory/conf/systemName/certs/ldapusr_cert.pem

TLS_KEY installDirectory/conf/systemName/certs/ldapusr_key.pem
```

NOTE: In .ldaprc, explicit paths must be used to specify the location of the certificate and key files, not variables.

4. Set the slapd.conf directive `TLSVerifyClient` to **allow**.
5. Restart the LDAP server.
6. Navigate to the `installDirectory/sbin` and issue the following command:

   ```
   sh ./dotestserver
   ```

7. Answer **y** to the **Do you want to connect using SSL? (y/n)** prompt.
8. Answer **n** to the **Do you want to authenticate using the client certificate referenced in ~/.ldaprc? (y/n)** prompt (this is intended for Mutual Authentication).
9. Answer the remaining prompts as appropriate for your site. An LDAP search command line containing: `-H ldaps://127.0.0.1:sslport` will be generated and issued where `sslport` is the value for the SSL port that you supplied during the configuration script. The results that are returned to the screen contain the `cn` of the userid entered at the prompts.

Testing SSL/TLS with mutual authentication

To test SSL\TLS with mutual authentication

1. In the OMVS home directory for the userID that is running the test, create a file called **.ldaprc**

   This file is referenced in the prompts of the dotestserver script. It provides the location of the client certificate and key for the script to present to the LDAP server.

2. Populate the .ldaprc file as appropriate for your site. The following sample .ldaprc contents correspond to the sample certificates and keys provided:

   ```
   TLS_CACERT installDirectory/conf/systemName/certs/ca_cert.pem
   TLS_CERT installDirectory/conf/systemName/certs/ldapusr_cert.pem
   TLS_KEY installDirectory/conf/systemName/certs/ldapusr_key.pem
   ```

NOTE: In .ldaprc, explicit paths must be used to specify the location of the certificate and key files, not variables.

3. Set the slapd.conf directive `TLSVerifyClient` to demand.
4. Restart the LDAP server.
5. Navigate to the `installDirectory/sbin` and issue the following command:

   ```
   sh ./dotestserver
   ```

6. Answer **y** to the **Do you want to connect using SSL? (y/n)** prompt.
7. Answer **n** to the **Do you want to authenticate using the client certificate referenced in ~/.ldaprc? (y/n)** prompt (this is intended for Mutual Authentication).
8. Answer the remaining prompts as appropriate for your site. An LDAP search command line containing:

```
-H ldaps://127.0.0.1:<sslport> -Y EXTERNAL
```

will be generated and issued where sslport is the value for the SSL port that you supplied during the configuration script. The results that are returned to the screen contain the cn of the userid entered at the prompts.

Testing SSL/TLS with mutual authentication and CRL checking using the CA certificate file method

To test SSL\TLS with mutual authentication and CRL checking using the CA certificate file method

1. To use the CA certificate file method, the CRL must be concatenated to the CA certificate file that is specified in slapd.conf and .ldaprc.
2. In the OMVS home directory for the userID that is running the test, create a file called **.ldaprc**
3. This file is referenced in the prompts of the dotestserver script. It provides the location of the client certificate and key for the script to present to the LDAP server.

   In the .ldaprc file, specify the revoked certificate and its key.
4. Populate the remainder of the .ldaprc file as appropriate for your site. The following sample .ldaprc contents correspond to the sample certificates and keys provided and point to the revoked client cert:

   ```
   TLS_CACERT installDirectory/conf/systemName/certs/ca_cert.pem
   ```

   ```
   TLS_CERT installDirectory/conf/systemName/certs/revoked_cert.pem
   ```

   ```
   TLS_KEY installDirectory/conf/systemName/certs/revoked_key.pem
   ```

NOTE: In .ldaprc, explicit paths must be used to specify the location of the certificate and key files, not variables.

5. Set the slapd.conf directives as follows:

   ```
   TLSVerifyClient demand
   ```

   ```
   TLSCRLCheck peer
   ```
6. Restart the LDAP server.
7. Navigate to the `installDirectory/sbin` and issue the following command.

   ```
   sh ./dotestserver
   ```
8. Answer **y** to the **Do you want to connect using SSL? (y/n)** prompt.
9. Answer **n** to the **Do you want to authenticate using the client certificate referenced in ~/.ldaprc? (y/n)** prompt (this is intended for Mutual Authentication).
10. Answer the remaining prompts as appropriate for your site. An LDAP search command line containing:

    ```
    -H ldaps://127.0.0.1:2413 -Y EXTERNAL
    ```

```
The results returned to the screen will contain the following error:
ldap_sasl_interactive_bind_s: Can't contact LDAP server (-1)
additional info: error:14094414:SSL
routines:SSL3_READ_BYTES:sslv3 alert
certificate revoked
```

Testing SSL/TLS with mutual authentication and CRL checking using the CA certificate path method

Using the CA certificate path method allows the CRL to be stored in separate files through use of the TLSCACertificatePath directive in slapd.conf. This method requires that the files are in USS, not MVS data sets. (The CRL and CA certificate can be concatenated in the same file.) This method requires special maintenance of the directory contents.

NOTE: The USS directory that contains individual CA certificates and CRLs in separate files (TLSCACertificatePath) must be specially managed using the OpenSSL c_rehash utility. When using this feature, the OpenSSL library will attempt to locate certificate files based on a hash of their name and serial number. The c_rehash utility is used to generate symbolic links with the hashed names that point to the actual certificate files. OpenSSL is supplied in the LDAP Bridge `installDirectory/bin` directory.

A script called dohashcerts in the LDAP Bridge `installDirectory/sbin` is run at startup time. It will automatically create the symbolic links of the certificates in the directory that is specified in the dohashcerts file. The default is:

```
certdir=$confdir/certs
```

If you store your certs and keys outside of the install directory, as recommended, you must edit this value in dohashcerts to match this location.

To test SSL\TLS with mutual authentication and CRL checking using the CA certificate path method:

1. In the OMVS home directory for the userID that is running the test, create a file called **.ldaprc**
2. This file is referenced in the prompts of the dotestserver script. It provides the location of the client certificate and key for the script to present to the LDAP server.

   In the .ldaprc file, specify the revoked certificate and its key.
3. Populate the remainder of the .ldaprc file as appropriate for your site. The following sample .ldaprc contents correspond to the sample certificates and keys provided and point to the revoked client cert.

   ```
   TLS_CACERT installDirectory/conf/systemName/certs/ca_cert.pem
   TLS_CERT installDirectory/conf/systemName/certs/revoked_cert.pem
   TLS_KEY installDirectory/conf/systemName/certs/revoked_key.pem
   ```

NOTE: In .ldaprc, explicit paths must be used to specify the location of the certificate and key files, not variables.

4. Set the slapd.conf directives as follows:
    - TLSVerifyClient `demand`
    - TLSCRLCheck `peer`
    - TLSCACertificatePath The path to the directory holding the CA certificate file and the CRL file
    - Comment out the TLSCACertificatePath directive.
5. Ensure that the TLSCACertificatePath directory that you specified in slapd.conf contains:
    - The CA certificate file. (The CRL may be standalone or concatenated to the CA certificate file.) This is supplied in ca_cert.pem.
    - The symbolic links that are generated by the OpenSSL c_rehash utility for each certificate file and each CRL file. (This is supplied by dohashcerts. Restart the LDAP server.)
6. Navigate to the `installDirectory/sbin` and issue the following command.

    ```
    sh ./dotestserver
    ```
7. Answer **y** to the **Do you want to connect using SSL? (y/n)** prompt.
8. Answer **n** to the **Do you want to authenticate using the client certificate referenced in ~/.ldaprc? (y/n)** prompt (this is intended for Mutual Authentication).
9. Answer the remaining prompts as appropriate for your site. An LDAP search command line containing:

    ```
    -H ldaps://127.0.0.1:<sslport> -Y EXTERNAL
    The results returned to the screen will contain the following error:
    ldap_sasl_interactive_bind_s: Can't contact LDAP server (-1)
    additional info: error:14094414:SSL
    routines:SSL3_READ_BYTES:sslv3 alert
    certificate revoked
    ```

Sample TLSCACertificatePath directory listing

The following is a listing of the TLSCACertificatePath directory that is used in the example above. It shows the hash-named links to the certificate files. In this case:

- ca_cert.pem is the file with CA certificate and CRL
- revoked_cert.pem is the client certificate that is listed as revoked in ca_cert.pem. It is also the TLS_CERT specified in the .ldaprc.

```
3808b051.0@ -> revoked_cert.pem
7b47a073.0@ -> ca_cert.pem
7b47a073.r0@ ->crl_0001.pem
```

```
ccd42347.0@ -> server_cert.pem
f9144561.0@ -> ldapusr_cert.pem
ca_cert.pem
ldapusr_cert.pem
ldapusr_key.pem
revoked_cert.pem
revoked_key.pem
server_cert.pem
server_key.pem
```

Certificate export and conversion

The production certificate and key files for the LDAP Bridge must be stored in a USS directory or in MVS data sets. You must update the slapd.conf file directives with the names and locations of the certificate, its private key, and the CA certificate.

Many sites store a stock of certificates and keys in the mainframe security database. In order to use such certificates and keys with the LDAP Bridge, they must be exported to USS or MVS data sets and converted to PEM format.

To facilitate the export and conversion process, a script called doexportcerts is provided in `installDirectory/sbin`. The doexportcerts script references the target directory for the exported and converted files with a parameter exportdir. The value of the exportdir parameter is:

```
exportdir=$confdir/certs/export
```

To locate the desired certs and keys in the mainframe security database, doexportcerts uses a configuration file called exportcerts.conf in `installDirectory/conf/systemName`.

The exportcerts.conf file must be pre-populated with records specifying the type, scope, and label parameters. For details on these parameters, see the header of the exportcerts.conf file.

Exporting and converting certificates

To export and convert certificates

1. Navigate to the `installDirectory/sbin` and issue the following command:

   ```
   sh ./doexportcerts
   ```

2. Convert the certificates to PEM format.
3. Relocate the certificates to the appropriate protected directory or data set for use with the LDAP Bridge.

NOTE: This doexportcerts script will not work for a server certificate whose key is stored in hardware (in ICSF). There is no method for using such a certificate and key with the LDAP Bridge.

Key protection

To implement SSL/TLS in production, protection of the keys is very important. Unauthorized read access to the key could enable decryption of communication, impersonation of the server or other security breaches.

Storing keys in USS

If you choose to store keys in USS, store them in a directory outside of `installDirectory` so that they will be unaffected by any maintenance that is applied to the LDAP Bridge. Also, when working with technical support, support may request the subsets of the `installDirectory` directory be collected and sent. Storing the keys outside of `installDirectory` will prevent the unintended transmission of the keys to technical support.

Limit permissions on key files to read-only for the LDAP Bridge Admin ID.

NOTE: These permissions will not prevent any USS superuser (root) from intentionally or unintentionally accessing the key file.

Storing keys in MVS data sets

If you choose to store the files in MVS data sets the exposure to USS superuser access does not apply.

The syntax for specifying the location of MVS data sets in slapd.conf is as follows:

```
directive //fully qualified data set name
```

For example:

```
TLSCACertificateFile //'RQX.CERTS.CACERT'
TLSCertificateFile //'RQX.CERTS.SRVCERT'
TLSCertificateKeyFile //'RQX.CERTS.SRVKEY'
```

These files must all be in base64 format also referred to as PEM format.

When storing the CA certificate in an MVS data set, the TLSACertificatePath method for managing CRLs is not supported.

Tuning the LDAP server

The LDAP Bridge uses the OpenLDAP LDAP Server called slapd from www.OpenLDAP.org. There are several configuration files that govern the behavior of slapd.

In the `installDirectory/conf/systemName` directory, the slapd.conf file contains the following online configuration parameters for your site. Some parameters are for customer tuning, others should only be changed for support and diagnostic purposes. The customer settings are documented here.

Online configuration file

In the `installDirectory/conf/systemName` directory, the slapd.conf file contains the following online configuration parameters for your site:

Table 4: Online configuration file parameters

Parameter	Description
Include	Do not modify these settings.
Pidfile	Denotes the file that contains the UNIX program-id number.
Argsfile	Denotes the file that contains the arguments used at startup.
Sizelimit	Controls the maximum number of entries that the LDAP Bridge returns for an individual search operation. This parameter must be set to a number larger than the total number of profiles in your IBM RACF®, CA Top Secret®, or CA ACF2™ database.
Timelimit	Controls the maximum number of seconds that the LDAP Bridge spends attempting to service a search operation.
Idletimeout	The number of seconds the connector will keep an inactive session alive. Decreasing this parameter can improve performance by removing inactive sessions. However, if it is too low, clients will have to reconnect frequently. This will degrade performance. It is recommended that this parameter is set to 0 (timeout disabled).
'ldap2esm-process -missing-entries	When this parameter is set to true (the default value), add operations will succeed when the parent entry of the entry to be added does not yet exist in the LDAP

Parameter	Description
	directory, and modify operations will succeed when the entry to be updated does not exist in the directory. When set to false, the operations described above will fail. This setting has no affect on deletes. A delete will only be processed if the entry is in the LDAP directory.
Allow bind_v2	This enables back-level support for LDAP version 2 binds. This setting cannot be changed.

Backend configuration file

There is a backend configuration file specific to each of the following:

- IBM RACF® security system - slapd.racf.conf
- CA Top Secret® security system - slapd.tss.conf
- CA ACF2™ security system - slapd.acf2.conf

The back-end security files contain the following online configuration parameters:

Table 5: Backend configuration file parameters

Parameter	Description
Database	This parameter must always be set to **bdb**.
Lastmod	This parameter controls whether the LDAP Bridge stores the last time that any entry was modified. To improve performance, set this parameter to **Off**.
Readonly	This parameter must always be set to **Off**.
Suffix	The LDAP root entry for the LDAP Bridge. There must be one suffix parameter: `o=%company%`

Directory	This parameter must be set to:
	`%datadir%/bdb/racf in slapd.racf.conf`
	`%datadir%/bdb/tss in slapd.tss.conf`
	`%datadir%/bdb/acf2 in slapd.acf2.conf`

rootdn	This is the dn used by Synchronization Daemon to connect to the LDAP server. It must be kept in sync with the value in the following file: racf.conf when working with the racf2ldap plug-in. The default value is `cn=racfManager,o=%company%`

Parameter	Description
	tss.conf when working with the tss2ldap plug-in. The default value is `cn=tssManager,o=%company%` acf2.conf when working with the acf22ldap plug-in. The default value is `cn=acf2Manager,o=%company%`
Cachesize	To optimize performance, set this parameter to the total number of entries on your system. For example, if you have 20000 users and 5000 groups, set the cachesize to 25000 or greater. Setting the cachesize to a value too small impedes system performance, while a cachesize too large wastes system memory. Adjusting the cachesize can require adjusting the heap parameter in the `installDirectory/conf/systemName/stdenv.slapd` file.
Index	Specifies attributes to be indexed during the database process. If your LDAP clients frequently search based on certain attributes, such as cn or sn, you can add additional index statements as described in the section below. At minimum, it is recommended that you index the uid and member attributes.
tracelog-enabled	Specifies that a trace log will be created with incoming LDAP commands in plaintext readable format. The default setting for this parameter is false. Because of the high volume of output the tracelog should only be used for short periods of time under the direction of technical support.

If your LDAP clients frequently request searches based on attributes other than uid, member or objectClass, you can create additional index files to improve online performance.

Creating additional index files

To create additional index files, edit either the

`installDirectory/conf/systemName/slapd.racf.conf` file, the `installDirectory/conf/systemName/slapd.tss.conf` file, or the `installDirectory/conf/systemName/slapd.acf2.conf` file depending on the plug-in that you are working with.

To add an index for the cn (common name) attribute, use the following example:

```
index uid eq

index member eq

index cn pres,eq,sub,approx
```

Where the last line represents the required change. Any attribute can be indexed using the following values in the index statement:

- pres - creates a presence index
- eq - creates an equality index
- sub - creates a substring index
- approx - creates an approximate (phonetic) index.

STDENV: UNIX environment variables

The stdenv files in `installDirectory/conf/systemName` contain UNIX environment variables that affect batch and online processing:

stdenv.slapd - affects online processing (RQXSTART).

stdenv.slapadd - affects database load processing (RQXCONVT, RQXCONVR, RQXCONVA)

stdenv.racf2ldap, stdenv.tss2ldap, stdenv.acf22ldap - affects online processing (RQXSTT2L)

stdenv - affects all other processing (RQXSTOP, etc)

As delivered, these files are optimized for the various components that they affect. The following table describes the parameters defined in these files:

Table 6: stdenv: UNIX environment variables

Parameter	Description
_BPX_BATCH_SPAWN	Controls whether z/OS uses the spawn or fork/exec service to start UNIX processes. To optimize performance, set this parameter to **Yes**.
_BPX_SHAREAS	Controls whether spawned processes run in the same address space as the parent UNIX process. To minimize resource usage, set this parameter to **Yes**.
_BPX_SPAWN_SCRIPT	Controls whether UNIX treats spawned processes as shell scripts. To improve script performance, set this parameter to **Yes**.
_CEE_RUNOPTS:RPTS	Determines whether a storage report is generated. To generate a storage report, set this parameter to **RPTS (ON)**. To optimize performance, set this parameter to **RPTS(OFF)**.
_CEE_RUNOPTS:RPTO	Determines whether a CEE runtime option is generated. To generate a CEE runtime option report, set this parameter to **RPTO(ON)**. To optimize performance, set this parameter to **RPTO(OFF)**.

Parameter	Description
_CEE_RUNOPTS:STACK	Controls the size of the stack, that is used to spawn processes and threads. These parameters are delivered optimized for the LDAP Bridge.
_CEE_RUNOPTS: H	Controls the size of the overall storage heap in UNIX. This parameter is delivered optimized for the LDAP Bridge.
_CEE_RUNOPTS:ANYHEAP	Controls the size of the storage heap in UNIX allocated mainly above the 32M addressing line. This parameter is delivered optimized for the LDAP Bridge.
CEE RUNOPTS:HEAPPOOLS	Controls the size of the pre-allocated storage pools in the storage heap. These is delivered optimized for the LDAP Bridge.

LDAP security configuration file

The LDAP Bridge uses Access Control Lists (ACLs) to determine who can access the LDAP database and what actions they can perform. This section describes how to enable group-based access control, explains how ACLs are used in the LDAP Bridge, and provides example scenarios to help create ACLs that meet your site's requirements.

ACLs are defined in the `installDirectory/conf/systemName/slapd.acl.conf` file. To customize or create an ACL definition, simply add your ACL statement and save the file. Once any change is made to the file, you must recycle the LDAP Bridge for the new definition to take effect.

The scenarios presented here represent the most commonly used protection schemes for LDAP environments. If you find that your site has ACL requirements that are not discussed in this section, refer to the general ACL specification, that is available at the following location:

http://www.openldap.org/software/

General ACL format

The general format for an ACL statement is shown below:

```
access to db entries ldap attr by user/group permitted action
```

where `db entries`, `ldap attr`, `user/group`, and `permitted action` are all site-specific values that each have their own syntax requirements. You can specify several ACL definitions concurrently. However, you must give careful consideration to the order in which the definitions appear. The LDAP Bridge processes ACLs by selecting the first ACL definition in slapd.acl.conf that applies to the specified <db_entries>. Once found, the LDAP Bridge applies the access granted or denied by the ACL definition. Any subsequent ACLs defined for the same <db entries> are not evaluated. As such, if you choose to define several ACLs

for the same entry or entries, more specific ACL definitions should appear in the file before more general ACL definitions.

LDAP Bridge default settings

As delivered, the LDAP Bridge is configured to permit write database access to any authenticated user, and no database access to unauthenticated users. Only the directory administrator that is defined in the slapd.conf file is permitted write access.

Example 1

The LDAP Bridge uses the following default ACL definition:

```
access to *
by anonymous auth
by users write
```

Where:

Table 7:

ACL variable	Syntax	Meaning
db entries	*	Wildcard character that represents all database entries.
ldap attr	none	
user/group	anonymous	Anonymous represents unauthenticated users.
	users	Users represents authenticated users.
permitted action	auth	Auth allows users to authenticate.
	write	Write allows users to read the specified database entries.

The purpose of this ACL definition is to require users to authenticate if they want to view database entries. If an anonymous user attempts to access a database entry, they will be required to authenticate. Authenticated users are granted write access to the database.

Example 2

The LDAP Bridge uses the following default ACL definition:

```
access * attrs=%secs%Password
      by * none
```

Where:

Table 8:

ACL variable	Syntax	Meaning
db entries	*	Wildcard character that represents all database entries.
ldap attr	attrs=%secs%Password	%secs%Password represents the user passwords entry attribute.
user/group	none	none represents that no one is authorized to access the password value.
permitted action	none	no access or action is allowed.

By default, no users can access password values. In order to read password values the pwdsync plug-in must be installed and authorized users must be added to the access list. To add a user to the list, add lines like:

```
by dn.exact="uid=pwdadmin,ou=people,o=%company%" read
```

before the

```
"by * none"
```

line.

Allowing all users and groups read access to the entire database

To allow all users, authenticated or otherwise, to view all entries in the database, use an ACL definition similar to the following:

```
access to * by * read
```

Where:

Table 9:

ACL variable	Syntax	Meaning
db entries	*	Wildcard character that represents all database entries.
ldap attr	none	
user/group	*	Wildcard character that represents all users or groups.
permitted action	none	allows users to read the specified database entries.

The purpose of this ACL definition is to remove the authentication requirement from the viewing database entries.

Limiting entire database access to specific users

You can choose to permit only certain users read access to the entire database to protect sensitive information in the database. These ACL definition protection schemes are intended to work with another, more general, ACL definition that restricts other authenticated users to viewing only defined entries or entry attributes.

Example 1

To restrict read access of the entire database to a number of specific user IDs, use an ACL definition similar to the following:

```
access to *
by dn.exact="uid=USERID1,ou=people,o=company" read
by dn.exact="uid=USERID2,ou=people,o=company" read
```

Where:

Table 10:

ACL variable	Syntax	Meaning
db entries	*	Wildcard character that represents all database entries.
ldap attr	none	
user/group	dn.exact="uid=USERID1,	dn.exact represents an exact user ID entry within

ACL variable	Syntax	Meaning
	ou=people,o=company"	the database. USERID1 or USERID2 represents the user IDs of the authorized users. company represents the root dn you specified for the LDAP Bridge.
permitted action	read	Allows users to read the specified database entries.

Example 2

To restrict read access of the entire database based upon a user ID filter, use an ACL definition similar to the following:

```
access to *
by dn.regex="uid=*.*,ou=people,o=company" read
```

Where:

Table 11:

ACL variable	Syntax	Meaning
db entries	*	Wildcard character that represents all database entries.
ldap attr	none	
user/group	dn.regex=”uid=*.*, ou=people,o=company″	dn.regex represents user IDs that match the specified characteristics. *.* is a regular expression used to filter user entries. For example, M.* would permit all user IDs beginning with M. company represents the root dn that you specified

ACL variable	Syntax	Meaning
		for the LDAP Bridge.
permitted action	read	Allows users to read the specified database entries.

Limiting entire database access to specific groups

You can choose to permit only certain groups read access to the entire database to protect sensitive information in the database by limiting who can view all the entries. These ACL definition protection schemes are intended to work with another, more general, ACL definition that restricts other authenticated users to viewing only defined entries or entry attributes.

Example 1

To restrict read access of the entire database to a number of specific groups, use an ACL definition similar to the following:

```
access to *
by group/tssGroup/member.exact=”cn=GROUP1,ou=groups,o=company" read
```

Where:

Table 12:

ACL variable	Syntax	Meaning
db entries	*	Wildcard character that represents all database entries.
ldap attr	none	
user/group	group/tssGroup/ member.exact= ”cn=GROUP1,ou =groups, o=company"	group/tssGroup/member.exact represents an exact group ID entry within the database.GROUP1 and GROUP2 represents the group ID of the authorized groups. company represents the root dn that you specified for the LDAP Bridge.
permitted action	read	Allows users to read the specified database entries.

Example 2

To restrict read access of the entire database based upon a group ID filter, use an ACL definition similar to the following:

```
access to *
by group/tssGroup/member.regex="cn=*.*,ou=groups,o=company" read
```

Where:

Table 13:

ACL variable	Syntax	Meaning
db entries	*	Wildcard character that represents all database entries.
ldap attr	none	
user/group	group/tssGroup/member.regex = ”cn=*.*,ou=groups,o=company”	group/tssGroup/member.regex represents group IDs that match the specified characteristics. *.* is a regular expression used to filter user entries. For example, M.* would permit all group IDs beginning with M. company represents the root dn you specified for the LDAP Bridge.
permitted action	read	Allows users to read the specified database entries.

Limiting entire database access to a specific IP address

You can choose to permit only requests from a specific IP address read access to the entire database. The purpose of this ACL definition is to protect sensitive information within the database by limiting who can view all the entries. This protection scheme is intended to work with another, more general, ACL definition that restricts other authenticated users to viewing only defined entries or entry attributes.

Example 1

To restrict read access of the entire database to a specific IP address, use an ACL definition similar to the following:

```
access to *
by peername.ip=IPADDRESS read
```

Where:

Table 14:

ACL variable	Syntax	Meaning
db entries	*	Wildcard character that represents all database entries.
ldap attr	none	
user/group	peername.ip=IPADDRESS	peername.ip represents an exact IP address making an LDAP request. IPADDRESS represents the IP address of the authorized request.
permitted action	read	Allows users to read the specified database entries.

Tuning the LDAP database

The LDAP server uses the open source BDB (Berkeley database) as the location where the LDAP data is stored. The there are several configuration files that govern the behavior of BDB.

DB_CONFIG: database variables

The DB_CONFIG files in `installDirectory/conf/systemName` contain database settings that affect batch and online processing:

- DB_CONFIG.slpad - affects online connector processing (RQXSTART).
- DB_CONFIG.slapadd - affects database load processing (RQXCONVT, RQXCONVA, RQXCONVR)

As delivered, these files are optimized for the processes they affect. The following table describes the parameters defined in these files:

Table 15: Database variables

Parameter	Description
set_cachsize	Controls the size of the cache. The format is: `set_cachesize gigabytes, bytes number_of_caches`

Parameter	Description
	gigabytes must be set to 0. bytes must be the size of installDirectory/data/plugin/secs/ldif2entry.bdb + 20%. number_of_caches must be set to 1. Where plugin is the name of the LDAP Bridge plug-in that you are working with. To tune this parameter, given an ldif2entry.bdb size of 50,000,000, the setting would be: `set_cachesize 0 60000000 1`
set_flags	DB_TXN_NOSYNC controls whether the database flushes changed data to the log and the database. Speeds up database loads. DB_TXN_NOT_DURABLE controls whether the database logs changes for recovery. Speeds up database loads.

Tuning the LDAP Bridge for data recoverability and durability

The LDAP Bridge can be tuned to suit the recoverability and durability of your data. You can specify:

- whether a recovery should be run at every startup. By default, recovery is not run at every startup.

  To change this setting, open the following file in a text editor:

  `/sbin/doslapd`

  In doslapd the following lines are commented out:

```
: rm -f $bdbdir/$secs/__db.* 1>/dev/null 2>&1
     if whence db_recover 1>/dev/null 2>&1; then
          echo "$pgname: running db_recover"
          db_recover -v -h $bdbdir/$secs
          xcode="$?"
if Ý "$xcode" != 0 ¨; then
echo "$pgname: db_recover exited with status: $xcode"
```

```
break
            fi
      fi
```

In order to run a recovery at every startup, these lines must not be commented out.

- the frequency of checkpoints. At each the setting of the DB_TXN_NOT_DURABLE and DB_TXN_NOSYNC parameters determines what happens at each checkpoint. By default, the checkpoint parameter is set follows for the first database definition:

  ```
  checkpoint 10 10
  ```

  This forces a checkpoint to occur every 10 KB or every 10 minutes. One checkpoint per minute is the maximum allowed frequency. This will ensure that the database is updated every minute or every one KB, however, it will also increase disk and resource usage. You can increase either of these parameters, at the expense of recovery granularity.

  To modify the checkpoint parameter:

  In a text editor, open the file from the list below that corresponds to the plug-in that you are working with:

  ```
  ./conf/systemName/slapd.racf.conf
  ```

  ```
  ./conf/systemName/slapd.tss.conf
  ```

  ```
  ./conf/systemName/slapd.acf2.conf
  ```

- whether data is written to a file and the database at every checkpoint using the DB_TXN_NOT_DURABLE and DB_TXN_NOSYNC flags. By default these are set so that database and file will be updated at every checkpoint. This provides recoverability and integrity if the server goes down, through any process other than a normal shutdown, but can result in a performance cost. You can modify the frequency of database updates by removing the comment character inferno of the DB_TXN_NOSYNC and DB_TXN_NOT_DURABLE flags.

  To change the setting of these flags:

  Open the following file in a text editor:

  ```
  ./conf/systemName/DB_CONFIG.slapd
  ```

  and remove the comment characters from following flags:

  ```
  #set_flags DB_TXN_NOSYNC
  ```

  ```
  #set_flags DB_TXN_NOT_DURABLE
  ```

Tuning the synchronization daemon

Most customization of the synchronization daemon occurs in the plug-in.conf configuration file, where the plug-in is the LDAP Bridge plug-in that you are working with. The sections below describe the various parameters in this file and presents step-by-step instructions for performing various common customization tasks.

Synchronization daemon configuration settings are stored in `installDirectory/conf/systemName/plug-in.conf`. As delivered, this file enables the synchronization daemon to synchronize your security database (IBM RACF® CA Top Secret®, or CA ACF2™) with the LDAP Bridge.

Synchronization daemon general definitions

The following parameters control the global functioning of the synchronization daemon:

Table 16: Synchronization daemon parameters

Parameter	Default Value	Description
LOGDIR	%logdir%	Configured at run time from site.variables, used by synchronization daemon for location to write synchronization daemon logs.
DATADIR	%datadir%	Configured at run time from site.variables, used by synchronization daemon to find the audit records.
REPLOG	%datadir%/replog.dat	Configured at run time from site.variables, used by Synchronization Daemon to write LDAP server change logs.
POLL	2	Polling rate in seconds for Synchronization Daemon to look for audit records.
RETRY	100	Specifies the number of retry attempts for a non-responsive LDAP server.
LOGLEVEL	4	Log level for event

Parameter	Default Value	Description
		details in LOGDIR/tss2ldap.log Range from 0 to 5, 0=minimal information logged, 5=maximum information logged. Recommended 4 for proof of concept and 0 for normal operations.
CONVERTLOGLEVEL	0	Logging for the database build process
RETAIN	30	Specifies how records are to be written to plug-in/old, where plug-in is the name of the LDAP Bridge plug-in that you are working with. Values are: 0 = SMF records are written to plug-in/old, and are not deleted. For example records for the tss2ldap plug-in are stored in tss2ldap/old. -1 = SMF records are deleted once they are processed and are not written to plug-in/old.

Parameter	Default Value	Description
		nn = SMF records are written to plug-in/old and records older than nn days are deleted where nn is a number between 0 and 999.
RETAINPOLL	86400	Poll rate in seconds for the cleanup of secs2ldap/old by function of the RETAIN parameter.
NOTIFY	%sec%manager@%company%CONSOLEoperations@%company%	Specifies the e-mail addresses of personnel to notify in case of errors equal to or greater than the NOTIFYLEVEL, below.
NOTIFYLEVEL	SERIOUS	Specifies the level of messages to trigger a notification e-mail to the personnel listed in NOTIFY, above. Values are: WARNING - Informational SERIOUS - Config. error must be fixed SEVERE - Possible data loss FATAL - Error resulting in termination.
HOST	%hostname%	Configured at run time from site.variables, tells Synchronization Daemon where to find the LDAP server.

Parameter	Default Value	Description
PORT	%hostport%	Configured at run time from site.variables, tells Synchronization Daemon the port to use at the LDAP server.
LDAPVERSION	3	Specifies the supported LDAP version. Do not change this parameter.
ORGDN	High-level qualifier.	Configured at run time from site.variables. LDAP Root.
AMANAGERDN	High-level qualifier.	Specifies the LDAP Distinguished Name used to perform LDAP updates.
SSL	N	Specifies whether SSL is to be used for communication to the connector. This is usually not necessary for local communications with the LDAP Bridge.
HLQ	High-level qualifier.	Specifies the high-level qualifier(s) for your z/OS data sets for this product.
TSSCOMMAND	TSS LIST(%s)DATA(ALL)	Specifies the TSS command used to synchronize audit record content. This must be kept in sync with HLQ.JCLLIB (TSSCFILE) This applies only when working with CA Top Secret®.

Tuning the MVS data sets

ATTR file

The HLQ.ATTR file determines the fields and profile types that are exposed in your LDAP Bridge installation. You can modify this file to add, remove, or modify fields, depending on the needs of your client LDAP applications.

In the ATTR file, the value in column 1 under 'USED' for each record specifies whether the field will be loaded into the LDAP directory and synchronized during online transactions. To enable the record, specify **Y** in column 1 under **USED**. To ensure that the column is not loaded into the LDAP directory or synchronized, specify a value other than **Y**. Each record's original value is shown in column 3 under **USED**. The values in column 3 are informational only and have no impact on the behavior of the LDAP Bridge.

NOTE: Some columns have values other than Y or N. Records that have an asterisk (*) in column 1 are comment records only. They should never be enabled by changing the asterisk to a Y. Other records have characters other than Y or N to signal the configuration script to enable them in particular circumstances. These records are disabled. Only records with the column 1 value set to Y are enabled.

NOTE: The LDAP Bridge cannot access or convert encrypted fields, and verifies all user ID and password combinations by making API calls to your mainframe security database.

Installation exit

The HLQ.SCRLIB MVS file contains several sample installation exit source programs. The initial comments that are contained in all installation exit programs contain programming information. To compile an installation exit, use RQXCMPLK in the JCLLIB. The following table summarizes the delivered sample programs:

Table 17: Installation exit sample programs

Members	Language	Description
RQXCONVAF	COBOL	Filter installation exit called by RQXCONVA, the CA ACF2™ conversion process. Filters the ACF2 profiles loaded into the LDAP directory. By default, RQXCONVA loads all profile types defined in the ATTR file. If you need to load only certain profiles, such as all users beginning with the letter A, then code this installation exit. This installation exit is controlled by the FILTER flag

Members	Language	Description
		in HLQ.JCLLIB(ACF2CONV), that must be set to YES for it to be enabled. This member is only used when you are working with the acf22ldap plug-in.
RQXCONVAU	COBOL	Rule installation exit called by RQXCONVA, the ACF2 conversion process. Contains additional data manipulation rules that are not delivered as part of the product. To define a new rule that, for example, converts names into special e-mail address, then code this installation exit. You must modify the ATTR file to specify the new rules for the attributes to which it applies. This member is only used when you are working with the acf22ldap plug-in.
RQXCONVRF	COBOL	Filter installation exit called by RQXCONVR, the IBM RACF® conversion process. Filters the RACF profiles loaded into the LDAP directory. By default, RQXCONVR loads all profile types defined in the ATTR file. If you need to load only certain profiles, such as all users beginning with the letter A, then code this installation exit. This installation exit is controlled by the FILTER flag in HLQ.JCLLIB(RACFCONV), that must be set to YES for it to be enabled. This member is only used when you are working with the racf2ldap plug-in.
RQXCONVRU	COBOL	Rule installation exit called by RQXCONVR, the RACF conversion process. Contains additional data manipulation rules that are not delivered as part of the product. To define a new rule that, for example, converts names into special e-mail address, then code this installation exit. You must modify the ATTR file to specify the new rules for the attributes to which it applies. This member is only used when you are working with the racf2ldap plug-in.
RQXCONVTF	COBOL	Filter installation exit called by RQXCONVT, the CA Top Secret® conversion process. Filters the Top Secret profiles loaded into the LDAP directory. By default, RQXCONVT loads all profile types defined in the ATTR file. If you need

Members	Language	Description
		to load only certain profiles, such as all users beginning with the letter A, then code this installation exit. This installation exit is controlled by the FILTER flag in HLQ.JCLLIB (TSSCONV), that must be set to YES for it to be enabled. This member is only used when you are working with the tss2ldap plug-in.
RQXCONVTU	COBOL	Rule installation exit called by RQXCONVT, the Top Secret conversion process. Contains additional data manipulation rules that are not delivered as part of the product. To define a new rule that, for example, converts names into special e-mail address, then code this installation exit. You must modify the ATTR file to specify the new rules for the attributes to which it applies. This member is only used when you are working with the tss2ldap plug-in.

MVS data set security

You must protect the following files so that access is available only to key personnel and the protected user ID defined for the RQXSTART, RQXSTOP, EQXCNVA, RQXCNVR and RQXCNVT jobs:

- HLQ.JCLLIB
- HLQ.SRCLIB
- HLQ.LOADLIB
- HLQ.ATTR

DEBUGL parameter

The DEBUGL parameter in the RACFCONV, TSSCONV and ACF2CONV jobs controls the amount of output generated during the database load and refresh jobs. To optimize performance, this parameter is set to 000 by default. It can be set to 256 to produce full trace debugging output.

5

LDAP Schema

The LDAP Bridge interacts with your mainframe security database using a mapping file that is provided by your mainframe security database and a schema provided by the LDAP Bridge.

The `installDirectory/conf/systemName/schema` directory contains the LDAP schema files that are used by the LDAP Bridge: acf2.schema, racf.schema, and tss.schema.

By default, these files contain all necessary attributes and objectclasses to support the definitions in the corresponding ATTR files, whether or not these definitions are enabled there. During the running of the configuration script, the appropriate ATTR for the mainframe security database that you are working with is copied to the MVS data set HLQ.ATTR. It is this data set that is used by the LDAP Bridge, any ATTR changes must be made to this data set. Modifications to this file are necessary if you:

- need to change an attribute name
- need to create a new attribute
- want to load a custom field that is not defined by default.

See the Chapter template for explanations about the various paragraph, character and other tags.

Attribute definitions

Attribute definitions are located at the top of the schema file. To change an attribute name, locate that attribute and modify the name. To create a new one, find a similar attribute definition and copy it. Here is a typical attribute definition:

```
attributetype (1.3.6.1.4.1.12471.1.1.1.27
NAME 'racfData'
DESC 'racfData'
EQUALITY caseIgnoreMatch
SUBSTR caseIgnoreSubstringsMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE)
```

Attribute definitions support the following statements:

Table 18: Attribute definitions supported statements

Statement	Description
attributetype	Constant identifying this as an attribute definition. Must be followed by attribute definitions enclosed in parentheses.
OID number	Object Identifier. Do not change for existing attributes. For new attributes, use 1.3.1.4.1.12471.1.1.*xxx*, where *xxx* is a number greater than 500. OIDs must be unique.
NAME	The name of this attribute, enclosed in single quotes.
DESC	An optional description, enclosed in single quotes.
SYNTAX	The data type of this attribute. The product uses these syntaxes: **SYNTAX / Meaning** 1.3.5.1.4.1.1466.114.121.1.15 String, case ignored 1.3.6.1.4.1.1466.115.121.1.7 Boolean (TRUE/FALSE) 1.3.6.1.4.1.1466.115.121.1.12 LDAP Distinguished Name
EQUALITY	The equality matching rule. This depends on the syntax: **SYNTAX / Equality** 1.3.5.1.4.1.1466.114.121.1.15 caseIgnoreMatch 1.3.6.1.4.1.1466.115.121.1.7 booleanMatch 1.3.6.1.4.1.1466.115.121.1.12 distinguishedNameMatch
SUBSTR	The substring matching rule. This also depends on the syntax: **SYNTAX / Equality** 1.3.5.1.4.1.1466.114.121.1.15

Statement	Description
	caseIgnoreSubstringsMatch 1.3.6.1.4.1.1466.115.121.1.7 not applicable 1.3.6.1.4.1.1466.115.121.1.12 not applicable
SINGLE-VALUE	If present, indicates that this attribute can only have one value.

ObjectClass definitions

If you define a new attribute, in addition to the attribute definition described above, you will have to associate that attribute with one or more objectclasses. These objectclasses are also contained in the `installDirectory/conf/tss.schema` file, near the bottom. Here is a typical objectclass definition:

```
objectclass (1.3.6.1.4.1.12471.1.1.2.6
NAME 'racfUser'
DESC 'racfUser Class for RACF Connector'
SUP inetOrgPerson
STRUCTURAL
MAY (
racfUserpwdint $ racfUaudit $ racfUacc $
racfSpecial $ racfSeclevel $ racfSeclabel $ racfRevokeDate $
racfRevokecount $ racfRevoke $ racfResume $ racfRestricted $
racfPasswordInterval $ racfPasswordgen $ racfPassworddate $
racfPassword
racfOwner $ racfOperations $ racfOidcard $ racfModel $
racfLastjobtime $ racfLastjobdate $ racfGrpacc $ racfExpired $
racfDfltgrp $ racfData $ racfCustom $ racfCreatedate $
racfClauth $ racfCategory $ racfAuthority $ racfAuditor $
racfAdsp $ ou $ o $ cn $
sn $ givenName $ mail )
MUST (
uid
```

Objectclass definitions support the following statements:

Table 19: ObjectClass definitions supported statements

Statement	Description
objectclass	Constant identifying this as an attribute definition. Must be followed by attribute definitions enclosed in parentheses.
OID number	Object Identifier. Do not change for existing attributes. For new attributes, use 1.3.1.4.1.12471.1.1.*xxx,* where *xxx* is a number greater than 500. OIDs must be unique.
NAME	The name of this attribute, enclosed in single quotes.
DESC	An optional description, enclosed in single quotes.
SUP	The superior objectclass, in the objectclass inheritance tree. Entries defined in this objectclass inherit all attributes for the superior objectclasses.
STRUCTURAL	Indicates that this is a structural objectclass, and thus subject to inheritance rules.
MAY	A list of optional attributes that may be present for this entry, enclosed in parentheses and delimited by "$ ". If you add, modify or delete any attributes names, you should make corresponding changes to this list.
MUST	A list of required attributes that may be present for this entry, enclosed in parentheses and delimited by "$ ". If you add, modify or delete any attribute name, you should make corresponding changes to this list if that attribute appears here.

If you modify an attribute name, you must change that name in all objectclass MUST and MAY clauses in which at appears. If you add an attribute, you must list it in the appropriate MUST and MAY clauses for the objectclasses to which it applies. If you delete an attribute, you must remove it from all the MUST and MAY clauses in which it appears.

ATTR data set definitions

There is an entry in the schema for each entry in the ATTR data set. The following is a sample from the IBM RACF® ATTR data set corresponding to the attribute racfData (used in the example above). Refer to the RACF, CA ACF2™, or CA Top Secret® ATTR data set as appropriate for more detail on the attributes for your mainframe security database.

```
Used RACF Field LDAP Attribute Description Format RecType Offset
Length
```

```
X X USBD_INSTALL_DATA racfData Installation defined data Char 02000125
025500060008User
```

The **X** in column one indicates that this attribute is not enabled. Set it to **Y** to enable it.

CA ACF2™ considerations

In addition to the CA ACF2™ Logon ID records, the LDAP Bridge supports ACF2 access and resource rules. Support is provided for:

- modeling rules in the LDAP directory
- performing an initial load of existing rules from ACF2 to the LDAP directory
- maintaining synchronization, so that changes to rules will be replicated to LDAP in real-time
- performing rule maintenance (adding or deleting rules) using an LDAP client interface.

Rules are modeled in the LDAP directory with entries under ou=datasets and ou=resources, for example:

```
o=company.com
ou=datasets
acf2RuleKey=HLQ1
acf2RuleKey=HLQ2
...
ou=resources
acf2ResourceType=TY1
acf2RuleKey=QUAL1
acf2RuleKey=QUAL2
...
acf2ResourceType=TY2
acf2RuleKey=QUAL1
acf2RuleKey=QUAL2
...
...
```

Access rule entries are modeled with the acf2AccessRule objectclass, which will has the following attributes:

- acf2RuleKey - the $KEY attribute of the rule
- acf2RuleText - the text of the rule
- acf2RuleSize - the size (in bytes) of the rule

- acf2RuleUtilization - the utilization of the rule (the size with respect to the maximum size)
- acf2ModifierUser - the LID of the user that created the rule
- acf2ModifyDate, acf2ModifyTime - the date and time the rule was created

Resource rule entries are modeled with the acf2ResourceRule objectclass, identical to `acf2AccessRule`, with the addition of an `acf2ResourceType` attribute, modeling the type of resource.

When creating access rules through LDAP, add or modify operations the following attributes are required:

- acf2RuleKey
- acf2RuleText

When creating resource rules, the following attributes are required:

- acf2RuleKey
- acf2RuleText
- acf2ResourceType

The other attributes are ignore when creating rules, but they are populated when replicating from ACF2 to LDAP.

To enable these attributes, the HLQ.ATTR data set must be edited. At the bottom of the ATTR data set are the following records. By default the Resource and Access rules are not enabled. By changing the **N** in the first column to a **Y**, any particular attribute can be enabled.

Table 20:

N N	KEY	acf2RuleKey	access rule key
N N	TEXT	acf2RuleText	access rule text
N N	SIZE	acf2RuleSize	access rule size
N N	UTIL	acf2RuleUtilization	access rule utilization
N N	USER	acf2ModifierUser	access rule modifier
N N	DATE	acf2ModifyDate	access rule modification date
N N	TIME	acf2ModifyTime	access rule modification time
N N	TYPE	acf2ResourceType	resource rule type
N N	KEY	acf2RuleKey	resource rule key
N N	TEXT	acf2RuleText	resource rule text
N N	SIZE	acf2RuleSize	resource rule size

N N	UTIL	acf2RuleUtilization	resource rule utilization
N N	USER	acf2ModifierUser	resource rule modifier
N N	DATE	acf2ModifyDate	resource rule modification date
N N	TIME	acf2ModifyTime	resource rule modification time

For more information on the attributes for your mainframe security database, refer to the ACF2 ATTR data set.

IBM RACF® considerations

When working with IBM RACF®, the LDAP Bridge can be used as a security provider for Linux, through the pam_ldap pluggable security module. The LDAP Bridge supports the schema that is used by pam_ldap, primarily the posixAccount and posixGroup objectClasses.

To enable this support in the LDAP Bridge, edit the ATTR data set. Any records with the character **L** in the first position, must be edited changing the **L** to a **Y** in the first position. The records affected are the following:

```
L L posixGroup objectClass Group name as taken from the profile name
L L GPMEM_MEMBER_ID memberUid A user ID within the group
L L GPOMVS_GID gidNumber OMVS GID associated with the Group name
L L posixAccount objectClass objectclass for this entity
L L USBD_PROGRAMMER gecos The name associated with the user ID
L L USBD_DEFGRP_ID gidNumber The default group associated with the user
L L USOMVS_UID uidNumber OMVS UID associated with the user name
L L USOMVS_HOME homeDirectory OMVS HOME PATH associated with the UID
L L USOMVS_PROGRAM loginShell OMVS Default Program associated with the UID
```

For more information on the attributes for your mainframe security database, refer to the RACF ATTR data set.

CA Top Secret® considerations

The LDAP Bridge uses LDAP attributes that map to specific fields within the Top Secret database. For more information on the attributes for your mainframe security database, refer to the Top Secret ATTR data set.

6

Internationalization

The Locale for the LDAP Bridge is set by the configure script. To change the locale, re-run the configure script. Ensure that you record any customizations that you made before re-running the configure script so that they can be re-applied.

Troubleshooting

Recovering data after restarting the synchronization daemon

After a mainframe security database change has been processed, the synchronization daemon moves the SMF record from the `installDirectory/data/systemName/plugin/new` directory to the `installDirectory/data/systemName/plugin/old` or `installDirectory/data/systemName/plugin/error` directories, where:

- `/old` acts as an archive of CA Top Secret® audit records that can be used for debugging purposes, or to rebuild the Top Secret database.
- `/error` acts as an holding area for Top Secret audit records that were not processed successfully. You should send any records in the /error directory to support to determine the cause of the problem. This directory should normally remain empty.
- `plugin` is the LDAP Bridge plug-in that you are working with.

If the LDAP Bridge is stopped, mainframe security database changes accumulate in the directory so that none are lost when it is restarted.

When working with Top Secret, if the TSSINSTX installation exit is disabled, Top Secret changes cannot be captured or propagated, and are therefore lost.

When working with IBM RACF® or CA ACF2™, if the IEFU83 exits or the SLAPU83 or SLAPU83A module are disabled, changes at the mainframe security database are not captured or propagated and the LDAP Bridge database must be rebuilt with the appropriate jobs from JCCLIB.

plug-in.conf error definitions

Part of the plug-in.conf (where plug-in is the name of the plug-in that you are working with) file describes how the synchronization daemon handles various LDAP error conditions that are returned from the LDAP Bridge when working with the plug-in. When an LDAP add, modify or delete request from synchronization daemon fails on the target connector, the LDAP Bridge returns an LDAP error code.

```
ERROR text code level action[,action, action, ...]
```

All parameters must be separated by one or more spaces.

- ERROR - static text identifying this as an ERROR statement
- text - text message associated with the LDAP_error_code, included for descriptive purposes only
- code- standard LDAP error code returned from the connector
- level - Synchronization Daemon severity level for this error code: WARNING, SERIOUS, SEVERE or FATAL. See NOTIFYLEVEL, above.
- action - action synchronization daemon should take in the event of this error:
  - NONE - take no action
  - ABEND - terminate the synchronization daemon task
  - SLEEP - retry in 10 seconds
  - SEND - e-mail those identified in the NOTIFY statement
  - MOVE - move the CA Top Secret® change to the error directory.

Sample ERROR definitions

```
ERROR LDAP_SUCCESS 0 WARNING NONE
```

This rule tells synchronization daemon to take no action on successful LDAP requests.

```
ERROR LDAP_OPERATIONS_ERROR 1 FATAL ABEND
```

This rule tells synchronization daemon terminate in the event of an LDAP operations error (error code 1).

```
ERROR LDAP_SERVER_DOWN 81 WARNING SLEEP
```

This rule tells synchronization daemon to wait and then try again in the event that the LDAP Bridge is down (error code 81).

Insufficient memory error condition

If the LDAP Bridge exits with a return code of 0768, or if the job output shows messages such as "failure to allocate nnn bytes", or "cannot reallocate nnn bytes," this indicates an inability to allocate enough processor memory for HEAP storage.

To remedy this condition

1. Edit `installDirectory/conf/systemName/stdenv` to enable the storage report. Ensure that the appropriate section of line 5 appears as follows:

   ```
   _CEE_RUNOPTS=RPTS(ON),RPTO(ON)....
   ```

2. Re-create the problem and examine the storage report in the SYSOUT to determine the suggested values for the HEAP parameter.
3. Re-edit `installDirectory/conf/systemName/stdenv`. Ensure that the appropriate section of line 6 appears as follows:

   ```
   _CEE_RUNOPTS=...H(xxx,5M,ANYWHERE,KEEP,8K,4K)
   ```

   Where xxx is the suggested value for the HEAP parameter from the storage report.
4. If you adjust the heap size upwards, you will also have to adjust the REGION parameter in the RQXSTART JCL, as described in Ensuring sufficient region size.

Collecting diagnostic information using the dodiag script

During support incidents, the support team could need various config files, logs, audit records. To simplify the collection of these files, a script called dodiag is provided in `installDirectory/sbin`. You can run this script using the following syntax:

```
dodiag archive.pax sys=systemName parameters
```

The parameters are:

- conf
- data
- secdata
- scrdata
- logs
- sbin
- mvs
- all

The following are examples of the usage of this script:

Example 1

```
dodiag test1.pax
```

This results in a file test1.pax.Z containing files for all systems defined. It contains:

- the MVS data sets:
  - ATTR
  - JCLLIB
  - LOADLIB
  - SRCLIB
  - EXITLIB
- the `installDirectory/conf` directory including all system specific subdirectories
- the `installDirectory/logs` directory including all system specific subdirectories
- a recursive listing of `installDirectory, installDirectory/conf, installDirectory/data, installDirectory/logs, installDirectory/sbin`

NOTE: If any sensitive keys or certs are in the certs directory under conf, they will be collected. It is advised that all production certs be located in a directory outside of the install directory.

NOTE: If any of the MVS data sets are empty, a warning message comes up that has no effect on the script. EXITLIB is typically empty and causes this message to be produced: `FSUMF145 error when traversing the PDS(E) //'HLQ.EXITLIB'`

Example 2

```
dodiag test2.pax sys=SYSA
```

This results in a file test2.pax.Z containing files for SYSA only. It contains:

- the MVS data sets:
  - ATTR
  - JCLLIB
  - LOADLIB
  - SRCLIB
  - EXITLIB
- the `installDirectory/conf/SYSA` directory including all system specific subdirectories
- the `installDirectory/logs/SYSA` directory including all system specific subdirectories
- a recursive listing of `installDirectory, installDirectory/conf, installDirectory/data, installDirectory/logs, installDirectory/sbin`

Example 3

```
dodiag test3.pax sys=SYSA conf data
```

This results in a file test3.pax.Z containing files for SYSA only. It contains:

- the `installDirectory/conf/SYSA` directory including all system specific subdirectories
- the `installDirectory/data/SYSA` directory including all system specific subdirectories

- a recursive listing of `installDirectory, installDirectory/conf, installDirectory/data, installDirectory/logs, installDirectory/sbin`

However, the data directory will not be complete. It will not have the ldif file from the database build and it will not have the replog.dat which has the synchronization data. These are omitted because of the potentially sensitive nature of some of the information.

Example 4

```
dodiag test4.pax sys=SYSA conf secdata
```

This results in a file test4.pax.Z containing files for SYSA only. It contains:

- the `installDirectory/conf/SYSA` directory including all system specific subdirectories
- the `installDirectory/data/SYSA` directory including all system specific subdirectories
- a recursive listing of `installDirectory, installDirectory/conf, installDirectory/data, installDirectory/logs, installDirectory/sbin`

However the data directory will include the database and replog.dat and the ldif file extracted from the mainframe security database.

Example 5

```
dodiag test5.pax sys=SYSA scrdata
```

This results in a file test5.pax.Z containing files for SYSA only. It contains:

- a version of the ldif file extracted from the mainframe security database with the number and types of entries intact but actual data values overwritten with random characters
- a recursive listing of `installDirectory, installDirectory/conf, installDirectory/data, installDirectory/logs, installDirectory/sbin`

Example 6

```
dodiag test6.pax sys=SYSA all
```

This results in a file test6.pax.Z containing:

- all the files from the install directory and the MVS data sets and a scrambled version of the ldif file extracted from the mainframe security database
- a recursive listing of `installDirectory, installDirectory/conf, installDirectory/data, installDirectory/logs, installDirectory/sbin`

Expanding the /tmp directory in USS

During the collection of the various files for the compressed pax, dodiag writes a number of files to /tmp. If /tmp is not big enough or for some other reason it is not desirable to write to /tmp, then a new temporary directory can be specified with an export command before running `dodiag`. You can do this as follows:

For /bin/sh

```
export TMPDIR=/some/directory
dodiag archive.pax sys=systemName parameters
```

For /bin/tcsh:

```
% setenv TMPDIR /some/directory
% dodiag archive.pax sys=systemName parameters
```

These will cause dodiag to write the temp files to */some/directory*

Uninstalling the LDAP Bridge

To uninstall the LDAP Bridge

1. Shut down the LDAP Bridge server using the RC*STOP member of HLQ.JCLLIB.
2. Uninstall the appropriate exit. When working with:

   IBM RACF®, uninstall the SLAPU83 exit.

   - Issue the following commands to undo the current activation as appropriate:

     ```
     setprog exit,delete,exitname=sys.iefu83,modname=slapu83
     ```

     ```
     setprog exit,delete,exitname=sysstc.iefu83,modname=slapu83
     ```

     ```
     setprog exit,delete,exitname=sysjes2.iefu83,modname=slapu83
     ```

   - Undo any changes made to the PROGnn member of the SYS1.PARMLIB related to SLAPU83

   Top Secret, uninstall the TSSINSTX exit:

   - For an installation with only the LDAP Bridge using TSSINSTX, remove the TSSINSTX data set from the link listed library and refresh the Top Secret exit with the following commands:

     ```
     F TSS,EXIT(OFF)
     ```

     ```
     F LLA,REFRESH
     ```

     ```
     F TSS,EXIT(ON)
     ```

   - Undo any changes made to SYS1.PARMLIB(TSSPARM0) related to the LDAP Bridge.

   ACF2, uninstall the IEFU83 exit:

   - Issue the following commands to undo the current activation as appropriate:

     ```
     setprog exit,delete,exitname=sys.iefu83,modname=slapu83a
     ```

     ```
     setprog exit,delete,exitname=sysstc.iefu83,modname=slapu83a
     ```

     ```
     setprog exit,delete,exitname=sysjes2.iefu83,modname=slapu83a
     ```

3. Undo any changes made to the PROGnn member of the SYS1.PARMLIB related to SLAPU83.

4. Delete the HLQ.JCLLIB, HLQ.LOADLIB, HLQ.SRCLIB, HLQ.EXITLIB and HLQ.ATTR data sets.
5. Remove the install directory in an OMVS session with the command:

```
rm -r installDirectory
```

Contacting us

For sales or other inquiries, visit https://www.oneidentity.com/company/contact-us.aspx or call +1-800-306-9329.

Technical support resources

Technical support is available to One Identity customers with a valid maintenance contract and customers who have trial versions. You can access the Support Portal at https://support.oneidentity.com/.

The Support Portal provides self-help tools you can use to solve problems quickly and independently, 24 hours a day, 365 days a year. The Support Portal enables you to:

- Submit and manage a Service Request
- View Knowledge Base articles
- Sign up for product notifications
- Download software and technical documentation
- View how-to-videos
- Engage in community discussions
- Chat with support engineers online
- View services to assist you with your product