HP Select Identity LDAP Bridge for ACF2, RACF, and Top Secret

Version: 3.50

Installation and Configuration Guide

Document Release Date: September 2007
Software Release Date: September 2007

Legal Notices

Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notices

This product includes software developed by the Apache Software Foundation (http://www.apache.org/).

Select Identity uses software from the Apache Jakarta Project including:

- Commons-beanutils
- Commons-collections
- Commons-logging
- Commons-digester
- Commons-httpclient
- Element Construction Set (ecs)
- Jakarta-poi
- Jakarta-regexp
- Logging Services (log4j)

Additional third party software used by Select Identity includes:

- JasperReports developed by SourceForge
- iText (for JasperReports) developed by SourceForge
- BeanShell
- Xalan from the Apache XML Project
- Xerces from the Apache XML Project
- Java API for XML Processing from the Apache XML Project
- SOAP developed by the Apache Software Foundation
- JavaMail from SUN Reference Implementation
- Java Secure Socket Extension (JSSE) from SUN Reference Implementation
- Java Cryptography Extension (JCE) from SUN Reference Implementation
- JavaBeans Activation Framework (JAF) from SUN Reference Implementation

- OpenSPML Toolkit from OpenSPML.org
- JGraph developed by JGraph
- Hibernate from Hibernate.org
- BouncyCastle engine for keystore management, bouncycastle.org

This product includes software developed by Teodor Danciu (http://jasperreports.sourceforge.net). Portions Copyright © 2001-2004 Teodor Danciu (teodord@users.sourceforge.net). All rights reserved.

Portions Copyright 1994-2004 Sun Microsystems, Inc. All Rights Reserved.

This product includes software developed by the Waveset Technologies, Inc. (www.waveset.com). Portions Copyright © 2003 Waveset Technologies, Inc. 6034 West Courtyard Drive, Suite 210, Austin, Texas 78730. All rights reserved.

Portions Copyright © 2001-2004, Gaudenz Alder. All rights reserved.

Trademark Notices

AMD and the AMD logo are trademarks of Advanced Micro Devices, Inc.

Intel and Pentium are trademarks or registered trademarks of Intel Corporation in the United States, other countries, or both.

JAVA™ is a US trademark of Sun Microsystems, Inc.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

Oracle® is a registered US trademark of Oracle Corporation, Redwood City, California

UNIX® is a registered trademark of The Open Group.

Support

You can visit the HP software support web site at:

http://support.openview.hp.com/support.jsp

HP Software online support provides an efficient way to access interactive technical support tools. As a valued support customer, you can benefit by using the support site to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract.

To find more information about access levels and HP Passport, go to:

http://support.openview.hp.com/new_access_levels.jsp

Contents

1 Introduction

The HP Select Identity LDAP Bridge (LDAP Bridge) is an LDAP gateway that provides access to IBM Resource Access Control Facility (RACF), CA Top Secret (Top Secret) or CA ACF2 (ACF2) depending on which version you purchase. It enables you to access mainframe security-based data with LDAP, the LDAP Bridge extends mainframe authentication and authorization to your environment.About this Guide

Audience

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.
- RACF concepts such as password verification and resource authorization.
- Top Secret concepts such as password verification and resource authorization
- ACF2 concepts such as password verification and resource authorization.

These personnel must have authority to:

- Edit mainframe data sets, submit jobs, and install exits
- Access USS, run UNIX commands, and create HFS files

Conventions

Throughout this document, the following conventions are used:

- The use if *italics* indicates a variable.
- `Monospace` font indicates a program listing.
- The following variables refer to values specific to your site:
  - *sdir* - refers to the root directory in Unix System Services that you choose for this product. The default value is /usr/lpp/hpv35.
  - *SQUAL* - 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 mainframe security database:
    - racf - RACF

  - tss - CA Top Secret
  - acf2 - CA ACF2

- The variable *RC** is used to represent the High Level Qualifier corresponding to the version of the LDAP Bridge that you purchased.
  - HP Select Identity CA ACF2 LDAP Bridge - the high level qualifier for this product is RCY
  - HP Select Identity RACF LDAP Bridge - the high level qualifier for this product is RCX
  - HP Select Identity CA Top Secret LDAP Bridge - the high level qualifier for this product is RCZ
- Examples of directory paths apply to a multi-system installation. Refer to "*Directories Created During Installation and Configuration*" in Chapter 2 for the equivalent directories for a single-system installation.

Overview of the LDAP Bridge

There are three different versions of the LDAP Bridge. Each version works with a different mainframe security database.

- HP Select Identity CA ACF2 LDAP Bridge
- HP Select Identity RACF LDAP Bridge
- HP Select Identity CA Top Secret LDAP Bridge

The LDAP Bridge is composed of the following components:

LDAP Server Component

The LDAP server publishes a copy of the RACF, Top Secret, or ACF2 (mainframe security) database (depending on the version of the product that you have purchased). The database copy that is published is a real-time image of the entire 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 make a change to the LDAP database, 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.

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 the mainframe security database in response to the command. The mainframe security database

command is then translated into an equivalent LDAP command that updates the LDAP database copy accordingly. If the LDAP Bridge is stopped, mainframe security database changes accumulate in the Synchronization Daemon directory until it is restarted so that no changes are lost.

Reverse Password Synchronization Feature

The Reverse Password Synchronization is an option in the LDAP Bridge that allows you to extract user passwords from the mainframe security database for synchronization with other HP Select Identity identity stores.

In this and previous versions of the LDAP Bridge, passwords are pushed into your mainframe security database using the LDAP attribute userPassword. The values that are pushed with userPassword are not stored in the LDAP Bridge LDAP database and cannot be queried with LDAP searches.

The Reverse Password Synchronization feature uses new attributes racfPassword, acf2Password, or tssPassword depending on the mainframe security database that you are working with. Passwords can be both pushed to the mainframe security database and read from it using these new attributes.

The Reverse Password Synchronization feature

- allows you to retrieve passwords from the mainframe security database in response to on-demand LDAP searches on the database
- automatically notifies the LDAP database that the password has been changed at the mainframe security database so that HP Select Identity can pick up the change at a scheduled reconciliation

The way that the Reverse Password Synchronization feature works and the details of the set up process vary depending on the version of the LDAP Bridge that you are working with.

- **When working with HP Select Identity CA ACF2 LDAP Bridge**, the password values are encrypted and stored in the LDAP database. If all configuration and authorization conditions are met, the LDAP Bridge will decrypt the password and return it to HP Select Identity using SSL encryption. The ACF2 LDAP Directory Services (LDS) will be used to notify the LDAP Bridge when passwords in the target user population change. LDS sends both the notification of the change and the new password value itself. The password will be encrypted using context and 256-bit AES operational keys and stored in the LDAP database.
- **When working with HP Select Identity RACF LDAP Bridge**, the password values are never stored in the LDAP database. Instead a placeholder value is stored in the LDAP database to signal to HP Select Identity when the password changes. If all configuration and authorization conditions are met, the LDAP Bridge uses the RACF r_admin callable service to retrieve a PKCS #7 envelope containing the password and returns it to HP Select Identity using SSL encryption. After initially enabling the feature, RACF creates password envelopes only when passwords are changed. **Note:** RACF does not provide any support for enveloping (or otherwise retrieving) existing passwords. Password values will only be available through the LDAP Bridge when they have been changed at least one time after password enveloping was enabled in RACF.
- **When working with HP Select Identity CA Top Secret LDAP Bridge**, the password values are never stored in the LDAP database. Instead a placeholder value is stored in the LDAP database to signal to HP Select Identity when the password changes. If all configuration and authorization conditions are met, the LDAP Bridge uses the TSSCFILE to issue a LIST(acid) DATA(PASSWORD) command and returns it to HP Select Identity using SSL encryption.

The Reverse Password Synchronization feature is configured using the LDAP Bridge configure script during the installation of the LDAP Bridge.

Note: To change the state of the product from Password Synch enabled to Password Sync disabled, the configure script must be re-run. As a result, the USS files and MVS data sets that were created when the configure script was originally run will be overwritten and any customizations that had been made to the product will need to be reapplied.

2 Installing and Configuring the LDAP Bridge

System Requirements

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

Software Requirements

HP Select Identity CA ACF2 LDAP Bridge

- IBM z/OS 1.7 or later
- *eTrust* CA ACF2 r8 or r9 - If you plan to use the Reverse Password Synchronization Feature, ACF2 r8 APAR# QO77056 is not supported.

HP Select Identity RACF LDAP Bridge

IBM z/OS 1.7 or later

HP Select Identity CA Top Secret LDAP Bridge

- IBM z/OS 1.7 or later
- *eTrust* CA Top Secret r8 or r9 - If you plan to use the Reverse Password Synchronization Feature, Top Secret r8 APAR# QO78840 is not supported.

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 runtime libraries, with C-language support. The LDAP Bridge uses the R_admin interface to communicate with RACF, ACF2 and Top Secret.

Configuration Requirements for the Reverse Password Synchronization Feature

When working with the Reverse Password Synchronization feature you must configure your environment according to the following requirements:

HP Select Identity CA ACF2 LDAP Bridge

- The HP Select Identity connection must use an SSL connection.
- Ensure that the:

- LDAPBRIDGE.KEY.MASTER profile in the FACILITY class has READ and ALTER authority
- LDAPBRIDGE.KEY.OPER profile in the FACILITY class has READ and ALTER authority
- LDAPBRIDGE.KEY.REKEY profile in the FACILITY class has ALTER authority

- The HP Select Identity user ID that is making the password request must be ACL authorized at the LDAP server. Grant access to the appropriate users to access the synchronized password values. By default, no users can access synchronized password values. To do this, follow the instructions in `slapd.acl.conf`.
- Configure the LDAP Directory Services (LDS) according to the information in the RCYLDSA member. The LDS will be used to send notification to the LDAP Bridge when passwords in the target user population change. This is notification only, not the actual passwords. For more information on this configuration, see the LDAP Directory Services (LDS) chapter in the *eTrust CA-ACF2. Security for z/OS Administrator Guide.*

  Information: The default setting in the RCYLDSA job of BROADCST results enabling Password Synchronization for all users that are within scope for the user ID that is associated with the HPSI ACF2 LDAP Bridge job or started task. To limit the users enabled for password synchronization, remove BROADCST from the RCYLDSA job and issue the following command for each user:

  ```
  SET LID
  CHANGE logonid LDS
  ```

- Ensure that in ACF2, only the user IDs under which the LDAP Bridge runs have access to the 256-bit AES master key file, and the operational key files that are used to encrypt and decrypt the passwords. It is recommended that only one ID be so authorized, and that all of these processes run only under that ID. Also, it is recommended that the master key and operational keys each be stored in an MVS data set rather than an HFS file to ensure that UNIX superusers are not able to bypass file system security and access the master key. Sample JCL is provided in the JCLLIB member RCYKGPA to define the profiles used to protect access to key management functions and to permit access to them.
- Ensure that master key file and the operational key files are backed up in a secure fashion. If their contents are lost, all encrypted password data is unrecoverable.

HP Select Identity RACF LDAP Bridge

- The HP Select Identity connection must use an SSL connection.
- RACF's built-in PKCS #7 password enveloping feature must be configured and enabled for the target user population. For more information on this, see the chapter on Password Enveloping in the *IBM z/OS Security Server RACF Security Administrator's Guide*.
  - Setup the RACF keyring. This requires CA and server certificates. Sample certificates are supplied with the LDAP Bridge for testing purposes, but you must provide the certificates for production use.
  - Specify the certificate and private key to be used with password enveloping. Ensure that both the overlay ldifsync and overlay pwdsrch have directives specifying the same certificate and key that were imported into RACF for the Password Envelope configuration. You can do this using directives in either the PEM format or the PKCS #12 format as follows:
    PEM format:
    `pwdsrch-cert-file and pwdsrch-key-file`

```
ldifsync-cert-file and ldifsync-key-file
```

or

PKCS #12 format:

```
pwdsrch-pkcs12-file and pwdsrch-pkcs12-pwd
ldifsync-pkcs12-file and ldifsync-pkcs12-pwd
```

In this PEM format example the *server_cert* and *server_key* are located in the USS directory for the LDAP Bridge in the conf subdirectory for certs. The security of certificate and key depend on the USS file/directory security rules:

```
overlay pwdsrch
pwdsrch-sec-mgr %secs%
pwdsrch-cert-file %confdir%/certs/server_cert.pem
pwdsrch-key-file %confdir%/certs/server_key.pem
pwdsrch-min-ssf 112

overlay ldifsync
ldifsync-replog %datadir%/replog.dat
ldifsync-sec-mgr %secs%
ldifsync-cert-file %confdir%/certs/server_cert.pem
ldifsync-key-file %confdir%/certs/server_key.pem
ldifsync-min-ssf 112
```

- Make the certificates that are used in setting up the keyring, available outside of RACF to the LDAP Bridge.
- The ID under which the LDAP Bridge job or started task runs, must have the authority to issue the ADMN_XTR_PWENV call via r_admin. This requires READ access to the IRR.RADMIN.EXTRACT.PWENV profile in the FACILITY class. **Note:** Due to the sensitivity of the PWENV authority, the user ID used to run slapd must be highly restricted.
- Grant access to the appropriate users to access the synchronized password values. By default, no users can access synchronized password values. To do this, follow the instructions in `slapd.acl.conf`.

- Users whose passwords will be manipulated must
  - have OMVS segments.
  - be enabled for password enveloping.
- The HP Select Identity connection must use an SSL connection.
- The HP Select Identity user that is making the password request must be ACL authorized at the LDAP server
- Activate the RACF installation exit ICHPWX01 by adding the module to the LPALST. This exit is provided in the LDAP Bridge LOADLIB data set. This exit sends notification of password changes, not password values, to the LDAP Bridge. The LDAP Bridge populates the replog and the LDAP database with the password placeholder to indicate a change. New RACF installation exits, such as ICHPWX01, can only be properly activated with an IPL of the MVS system. **Note**: Without this exit in place, there will be no replog (synchronization) notification when password values have changed.

HP Select Identity CA Top Secret LDAP Bridge

- The HP Select Identity connection must use an SSL connection.
- The HP Select Identity user ID that is making the password request must be ACL authorized at the LDAP server.
- The PWVIEW control option must be set to YES in the Top Secret parameter file or when starting Top Secret.
- The user ID associated with the LDAP Bridge job or started task must have the authority to issue the TSS LIST(acid) DATA(PASSWORD) command. Due to the sensitivity of the PWVIEW authority, the ACID used to run slapd must be highly restricted.
- Configure the LDAP Directory Services (LDS) according to the information in the RCZLDST member. The LDS will be used to send notification to the LDAP Bridge when passwords in the target user population change. This is notification only, not the actual passwords. For more information on this configuration, see the LDAP Directory Services (LDS) chapter in the *eTrust CA-Top Secret Security for z/OS User Guide* and the Using the NDT Record chapter in the *eTrust CA-Top Secret Security for z/OS Command Functions Guide.*
- Users whose passwords need to be manipulated must be enabled for LDS notification on a user by user basis unless LDS is set to broadcast to all users. There is no option to enable groups of users by profile.
- Grant access to the appropriate users to access the synchronized password values. By default, no users can access synchronized password values. To do this, follow the instructions in `slapd.acl.conf`.

Before You Begin

Before you install the LDAP Bridge you must make decisions about your installation and prepare your environment for installation.

Reverse Password Synchronization

Before you install the LDAP Bridge, you must determine whether you want to use the Reverse Password Synchronization Feature. It is recommended that you make this decision before running the configure script because in order to change the state of the product from Password Synch enabled to Password Sync disabled, the configure script must be re-run. As a result, the USS files and MVS data sets that were created when the configure script was originally run will be overwritten and any customizations that had been made to the product will need to be reapplied.

If you choose to use the Reverse Password Synchronization Feature, ensure that you have met all of the requirements that are specified in the Configuration Requirements for the Reverse Password Synchronization Feature section.

Installation Type

Before you install the LDAP Bridge, you must determine the type of install 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 on running 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, see "*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 must be able to login to USS, create directories, and allocate MVS data sets.
- The user ID that is used to submit the JCL to build the LDAP database (**Database Admin ID**) - This user ID must be:
  - be authorized to write to the USS directories that are created by the install user
  - have an OMVS Segment
  - be a member of the same group as group owner of the USS directories
  - be authorized to run the RACF IRRDBU00 utility to extract from RACF (when working with the HP Select Identity RACF LDAP Bridge)

- be authorized to run TSSCFILE to extract from Top Secret (when working with the HP Select Identity CA Top Secret LDAP Bridge)
- be authorized to run the ACF2 BACKUP command or have access to a recent backup (when working with the HP Select Identity CA ACF2 LDAP Bridge)

- The user ID that is used to submit the JCL to start the LDAP Bridge (**LDAP Bridge Admin ID**). This user ID must be able to run the scripts, write to directories in USS, and submit commands through the R_admin interface to SAF.
- The user ID that is used by your HP Select Identity client to connect to the LDAP database and administer the mainframe security database (**HPSI client Admin ID**). This user must have an OMVS segment and the mainframe security database authority to run the set of commands that are needed by your HP Select Identity 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 ISHELL, OMVS, or telnet.
- be authorized to browse directories and issue UNIX commands in USS.
- 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/hpv35` directory, assign the appropriate permissions by issuing the following command:

  ```
  chmod 0771 /usr/lpp/hpv35
  ```
- 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/hpv35`, 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.

Tip: To see the owner and group owner of the `/usr/lpp/hpv35` directory, for example, issue the following command:

ls -ld /usr/lpp/hpv35

To change the owner of this directory to SLAPD, that requires superuser authority, issue the following command:

chown SLAPD /usr/lpp/hpv35

To change the group owner of this directory to ADMIN, issue the following command:

chgrp ADMIN /usr/lpp/hpv35

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 port this port.
- If you plan to use SSL access for all or part of the application, enable a port for SSL access. If you want to use the Reverse Password Synchronization feature, you must use 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 port this port.
- If you plan to use telnet access, use port 623, or the appropriate port that is used at your site for OMVS telnet access

Ensuring Sufficient Region Size

LDAP Bridge processes run as a 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 approximately 400 megabytes of memory.

The default REGION parameter that is coded in the JCL is 0M. It usually indicates no memory limitations. 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.

Verify Privileges

The LDAP Bridge executables must be APF-authorized and defined to program control in order to perform authentications against the mainframe security database. The Install ID must have certain permissions in order to set these attributes.

- If you are using RACF, you must have READ access to the BPX.FILEATTR.APF and BPX.FILEATTR.PROGCTL profiles in the FACILITY class.
- If you are using Top Secret, you must be permitted READ access to the BPX.FILEATTR.APF and BPX.FILEATTR.PROGCTL resources in the IBMFAC resource class.

- If you are using ACF2, you must be permitted to the BPX.FILEATTR.APF and BPX.FILEATTR.PROGCTL resources of the FAC type.

Install Directory

Before installing the LDAP Bridge you must select the directory where you want the product to be installed. When working with a multi-system installation, the LDAP Bridge install directory must be valid for all systems in the installation.

Tip: You can create a link on local system that matches the LDAP Bridge install directory for the remote system but resolves to the actual directory that is being used for the configuration.

For example: The initial install is in *sdir*= `/usr/lpp/hp351 on SYS1`

A second run of the multi-system install is necessary to set up for a remote system SYS2 where the *sdir* will be `/usr/lpp/hp352`.

Before you run the configuration script again on SYS1 from /usr/lpp, create a link on SYS1 with the following command:

ln -s /usr/lpp/`hp351` /usr/lpp/`hp352`

The second run of the configuration script will prompt with the actual directory `/usr/lpp/hp351`

The user can enter the name of the remote directory `/usr/lpp/hp352`.

The link will allow the configure script to proceed and find all the required files on the local system but it will use the link name in all files and data set members so that when the resulting configuration is mounted at the remote SYS2 it will be correctly configured for that system.

Installation Overview

The CD or downloaded version of the LDAP Bridge release media contains a compressed files that is used to install the LDAP Bridge into an HFS file system.

- HP Select Identity CA ACF2 LDAP Bridge includes the following compressed files:
  - `hpv35a.pax.Z`
- HP Select Identity RACF LDAP Bridge includes the following compressed files:
  - `hpv35r.pax.Z`
- HP Select Identity CA Top Secret LDAP Bridge includes the following compressed files:
  - `hpv35t.pax.Z`

After the initial `hpv35*.pax.Z` archive is expanded, the install directory contains five subdirectories and the configure script. The subdirectories are:

- install
- conf
- logs
- sbin
- data

The configure script prompts for certain variable values, then makes customized versions of the files from the install directory using the values that are input at the prompts. These customized files, along with the binaries, are placed in the `conf`, `data`, and `sbin` directories. A log of the configuration is placed in logs. During the configuration, a set of MVS data sets are created and populated with customized content. Among values that are input at the prompts, are directory paths, port numbers, and system names that will be specific to the installation machine.

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. The multi-system configuration path names are referenced in this document.

In a multi-system installation, the directories that the LDAP Bridge creates during the installation (conf, logs, sbin, 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.)

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	*sdir*/backup/	*sdir*/backup/
All of the configuration files for the system.	*sdir*/conf/	*sdir*/conf/*systemName*/
The sample certificates that are supplied for SSL and Reverse Password Synchronization testing.	*sdir*/conf/certs/	*sdir*/conf/*systemName*/certs/
A listing of all installed plugins.	*sdir*/conf/plugins/	*sdir*/conf/plugins/
The LDAP schema files.	*sdir*/conf/schema/	*sdir*/conf/*systemName*/schema/
The customized versions of the configuration files, that are tailored at run-time.	*sdir*/conf/tmp/	*sdir*/conf/*systemName*/tmp/

Contents	Single-system installation	Multi-system installation
All data files for the system.	*sdir*/data/	*sdir*/data/*systemName*/
All of the bdb databases that are used by the LDAP server	*sdir*/data/bdb/	*sdir*/data/*systemName*/bdb/
The bdb database containing the plug-in configuration information.	*sdir*/data/bdb/config/	*sdir*/data/*systemName*/bdb/config/
The bdb database containing the access log database	*sdir*/data/bdb/log/	*sdir*/data/*systemName*/bdb/log/
The bdb database that holds the product version information.	*sdir*/data/bdb/sb/	*sdir*/data/*systemName*/bdb/sb/
The main bdb database that holds the RACF security database information.	*sdir*/data/bdb/racf/	*sdir*/data/*systemName*/bdb/racf/
The LDIF files that are used to load the corresponding bdb databases.	*sdir*/data/ldif/	*sdir*/data/*systemName*/ldif/
The directories that are used by the Synchronization Daemon when working with RACF.	*sdir*/data/racf2ldap/	*sdir*/data/*systemName*/racf2ldap/
The new files that are written by the security exit when working with RACF.	*sdir*/data/racf2ldap/new/	*sdir*/data/*systemName*/racf2ldap/new/
Files that have been successfully processed by the Synchronization Daemon when working with RACF.	*sdir*/data/racf2ldap/old/	*sdir*/data/*systemName*/racf2ldap/old/
Files that have been unsuccessfully processed by the Synchronization Daemon when working with RACF.	*sdir*/data/racf2ldap/error/	*sdir*/data/*systemName*/racf2ldap/error/

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

Contents	Single-system installation	Multi-system installation
Files that have been unsuccessfully processed by the Synchronization Daemon when working with ACF2.	*sdir*/data/acf2ldap/error/	*sdir*/data/*systemName*/acf2ldap/error/
Installation and configuration materials for the LDAP Bridge. Nothing in this directory should be modified.	*sdir*/install/	*sdir*/install/
All of the LDAP Bridge log files.	*sdir*/logs/	*sdir*/logs/*systemName*/
Executable files	*sdir*/sbin	*sdir*/sbin
Any executable files that have been customized for the specific system. (This directory is typically empty.)	N/A	*sdir*/sbin/*systemName*/

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

Installing the LDAP Bridge

1. Transfer the product media to the machine where you want to install the LDAP Bridge. Transfer the `hpv35*.pax.Z` file (where `hpv35*.pax.Z` is the compressed installation file that was supplied with your version of the LDAP Bridge) to your HFS directory using FTP. Specify binary mode for the FTP transfer.
2. From an OMVS or telnet command prompt, issue the following commands:

```
cd sdir
pax -rv -px -f hpv35*.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

- Moves the source, load, JCL, and attributes files from the HFS to the MVS data sets
- Installs the LDAP Server and configuration database along with the Synchronization Daemon and 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 queried 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 run again, the script deletes the previous files and creates new ones based on the initial information provided.

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 displays that indicates the successful completion of the installation script.

1 Gather the following site-specific information:

   — Whether you want to perform single-system (s) or multi-system (m) configuration (default = multi-system)

      – If you choose to perform a multi-system installation, you can perform configuration for the current system or a different system. You will need to know the name of the system that you want to configure. 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) (*SQUAL*)

   — Permanent data set unit

   — Temporary data set unit

   — LDAP root

   — LDAP server clear text port

   — LDAP server SSL port

   — Security database locale

   — Whether you want to enable Reverse Password Synchronization. For more information on the Reverse Password Synchronization features and its installation, see Reverse Password Synchronization Feature on page 11.
   Note: To change the state of the product from Password Synch enabled to Password Sync disabled, the configure script must be re-run. As a result, the USS files and MVS data sets that were created when the configure script was originally run will be overwritten and any customizations that had been made to the product will need to be reapplied.

   — Whether you want to enable access logging - this option must be set to N.

2 From an OMVS or telnet command prompt, issue the following commands:

```
cd sdir
sh configure
```

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

Configuring the SMF Installation Exits When Working With RACF

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 RACF changes are made, and propagates the changes to the LDAP Bridge using LDAP. The SMF records are written to the *sdir*/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:

- 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 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. 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 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=SYSSTC.IEFU83,MODNAME=SLAPU83,
DSNAME=SQUAL.LOADLIB
```

and one of the following:

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

or:

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

or:

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

The command that you use 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 SLAPU83 Permanently

To install the SLAPU83 program permanently, follow the series of steps below:

1. Edit the PROG*nn* 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(SQUAL.LOADLIB)
```

   and one of the following:

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

   or:

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

   or:

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

   Alternatively, you can move SLAPU83 from *SQUAL*.LOADLIB to the LPALIB, in which 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.
- If JES2 is defined as a separate SMF subsystem, use the statements that reference SYSJES2.IEFU83.
- If neither TSO nor JES2 are defined as separate SMF subsystems, use the statements that reference SYS.IEFU83.

Once the PROG*nn* member has been edited in SYS1.PARMLIB, it may have to be activated by editing the COMMND*nn* member to include the following statement:

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

where *nn* corresponds to the suffix for the PROG*nn* member.

Setting the RACF System Options (SETROPTS)

To ensure that the LDAP Bridge database is always synchronized with 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.
- The *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.
- The *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 datasets.

Configuring the Top Secret Installation Exits

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

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 *sdir*`/data/`*systemName*`/tss2ldap/new` directory by the TSSINSTX program that runs as a Top Secret installation exit. TSSINSTX processes site specific code for a myriad of Top Secret functions. If a customized version of TSSINSTX is in use at your site, then you must integrate the LDAP Bridge version of TSSINSTX with it.

Installing the Top Secret Site Installation Exit TSSINSTX

To install this exit, perform the following steps:

1. Move SQUAL.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. To permanently activate the exit, edit SYS1.PARMLIB(TSSPARM0) and add the following statement.

EXIT(ON)

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 will need to be integrated into your version. See the SQUAL.SRCLIB(TSSINSTX) member for an example of the source modifications.

Configuring the SMF Installation Exits When Working With 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 ACF2 changes are made, and propagates the changes to the LDAP Bridge using LDAP. The SMF records are written to the *sdir*/data/*systemName*/acf22ldap/new directory 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.

Installing the IEFU83 Installation Exit

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 SMFPRM*nn* 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 the IEFU83 Dynamic installation exit Program

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=SQUAL.LOADLIB
```

and either:

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

or:

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

or:

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

where *SQUAL* is the high-level qualifier you created for the CA ACF2 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, follow the series of steps below:

1 Edit the PROG*nn* 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(SQUAL.LOADLIB)
```

and one of the following:

```
EXIT ADD
EXITNAME(SYSTSO.IEFU83)
MODNAME(SLAPU83A)
STATE(ACTIVE)
DSNAME(SQUAL.LOADLIB)
```

or:

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

or:

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

Where *SQUAL* is the high-level qualifier you created for the LDAP bridge. Alternatively, you can move SLAPU83A from *SQUAL*.LOADLIB to the LPALIB, in which 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.
- If JES2 is defined as a separate SMF subsystem, use the statements that reference SYSJES2.IEFU83.

- If neither TSO nor JES2 are defined as separate SMF subsystems, use the statements that reference SYS.IEFU83.

Once the PROG*nn* member has been edited in SYS1.PARMLIB, it activate it by editing the COMMND*nn* member to include the following statement:

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

where *nn* corresponds to the suffix for the PROG*nn* member.

Loading the LDAP Directory

The LDAP Bridge uses a directory database that is populated with data from your mainframe security repositories. When the LDAP Bridge database is initially loaded, the LDAP Bridge Synchronization Daemon keeps all databases synchronized.

Populating the LDAP Bridge Database

There are two ways populate the LDAP Bridge database. You can create the LDAP Bridge database using one job or, you can use a set of jobs where each job performs par of the population process. Populating the LDAP Bridge database using multiple steps is convenient if, in your environment, a single user does not have the authority to perform all three steps or if you anticipate the need to interrupt the database population process and resume it at a later time. Select the method that fits your environment.

The steps vary slightly depending on the mainframe security database that you are working with. Refer to the table in each step for the information specific to the mainframe security database that you are using.

Populating the LDAP Bridge Using a Single Job

1 Gather the following information:

Mainframe Security Database	**Required Information**
RACF	the name of the RACF database file
ACF2	the name of the ACF2 logonid database backup file

2 If you are working with ACF2, run the ACF2 BACKUP command to create a backup file for your ACF2 database. This step is not required when working with Top Secret or RACF.

3 Edit the following job found in *SQUAL*.JCLLIB:

Mainframe Security Database	Job
RACF	RCXCNVR
Top Secret	RCZCNVT
ACF2	RCYCNVA

Edit the JCL for this job so that the JOBCARD and the parameters including the SECFILE and the cylinder allocations are appropriate for your environment.

4 Submit the following job using the Database Admin ID:

Mainframe Security Database	Job
RACF	RCXCNVR
Top Secret	RCZCNVT
ACF2	RCYCNVA

Populating the LDAP Bridge Using Multiple Jobs

1 Gather the following information:

Mainframe Security Database	Required Information
RACF	the name of the RACF database file
ACF2	the name of the ACF2 logonid database backup file

2 Edit the following job found in *SQUAL*.JCLLIB:

Mainframe Security Database	Job
RACF	RCXCNVR1
Top Secret	RCZCNVT1

Edit the JCL for this job so that the JOBCARD and the parameters including UNFILE, and SECFILE and the cylinder allocations are appropriate for your environment.
Note: the value for UNFILE must be the same in RCXCNVR1 and RCXCNVR2 (when working with RACF) and the same in RCZCNVT1, and RCZCNVT2 (when working with Top Secret). The UNFILE parameter is not required when working with ACF2.

3 Edit the following job found in *SQUAL*.JCLLIB:

Mainframe Security Database	Job
RACF	RCXCNVR2
Top Secret	RCZCNVT2
ACF2	RCYCNVA2

Edit the JCL for this job so that the JOBCARD and the parameters including UNFILE, and SECFILE and the cylinder allocations are appropriate for your environment.
Note: the value for UNFILE must be the same in RCXCNVR1 and RCXCNVR2 (when working with RACF) and the same in RCZCNVT1 and RCZCNVT2 (when working with Top Secret). The UNFILE parameter is not required when working with ACF2.

4 Edit the following job found in *SQUAL*.JCLLIB:

Mainframe Security Database	Job
RACF	RCXCNVR3
Top Secret	RCZCNVT3
ACF2	RCYCNVA3

Edit the JCL for this job so that the JOBCARD and the parameters including UNFILE, and SECFILE and the cylinder allocations are appropriate for your environment.
Note: the value for UNFILE must be the same in RCXCNVR1 and RCXCNVR2 (when working with RACF) and the same in RCZCNVT1 and RCZCNVT2 (when working with Top Secret). The UNFILE parameter is not required when working with ACF2.

5 Submit the following job to unload data from the security system database:

Mainframe Security Database	Job or Command
RACF	RCXCNVR1 job
Top Secret	RCZCNVT1 job
ACF2	ACF2 BACKUP command

6 Submit the following job to convert the unloaded data to an LDIF file:

Mainframe Security Database	Job
RACF	RCXCNVR2
Top Secret	RCZCNVT2
ACF2	RCYCNVA2

7 Submit the following job to load the LDIF file into the LDAP directory:

Mainframe Security Database	Job
RACF	RCXCNVR3
Top Secret	RCZCNVT3
ACF2	RCYCNVA3

8 As a result of running the RCXCNVR1 job, or the RCZCNVT1 job a permanent data set containing information from the mainframe security database is created. Secure or delete this data set.

3 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.

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 in user IDs see the "*User IDs"* section.

Submitted Jobs

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

Started Tasks

To create a started task that starts the LDAP Bridge, customize the *RC**TASK JCL that is provided in the *SQUAL*.JCLLIB data set.

Starting the Synchronization Daemon

The Synchronization Daemon starts automatically using the same *RC**START JCL that is used to start the LDAP Bridge. Whenever you start the LDAP Bridge, the Synchronization Daemon is also active.

The REGION Parameter

Setting the REGION parameter of the *RC**START 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. See *"Ensuring Sufficient Region Size"* for further information on allocating a sufficient region size.

The TIME parameter

Setting the TIME parameter of the *RC**START 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

To stop the LDAP Bridge:

- with the *RC**STOP member of the JCLLIB data set. Add job card information to the JCL, then submit the job. All condition codes return as zero.
- by issuing the `dostop` command from a UNIX command prompt.
- 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 RC*START,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

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

   ```
   cd /sdir/sbin
   sh 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 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 /sdir/sbin
   sh 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 *sdir*`/logs/`*systemName*`/racf2ldap.log` to determine the cause of the error.

Testing the Synchronization Daemon with 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 /sdir/sbin
   sh 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 *sdir*`/logs/`*systemName*`/tss2ldap.log` to determine the cause of the error.

Testing the Synchronization Daemon with ACF2

1. Verify that the 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 /sdir/sbin
   sh 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 *sdir*`/logs/`*systemName*`/acf22ldap.log` to determine the cause of the error.

Testing 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 /sdir/sbin
   sh 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 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 /sdir/sbin
   sh 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 ACF2 Administration from an LDAP Client

ACF2 administration cannot be tested from an LDAP client.

Testing the LDAP Change Log for Synchronization Logging

1. Verify that the LDAP Bridge is running.
2. Enter OMVS from TSO.
3. Enter the following commands:
   ```
   cd /sdir/sbin
   sh dotestls
   ```
4. Respond to the prompts for your user ID and password.
5. The script displays an attribute, replog, that contains the changes made to the server as part of the previous tests, in LDIF format.

Testing the TSO command forwarding

1. Verify that the LDAP Bridge is running.
2. Enter OMVS from TSO.

3 Enter the following commands:

```
cd /sdir/sbin
sh dotestl2tso
```

4 Respond to the prompts for your user ID and password.

5 The script displays the output from the TSO PROFILE command in BASE64 encoding.

4 Tuning the LDAP Bridge

This chapter contains information about tuning the LDAP Bridge. You can use the LDAP Bridge without tuning it. However, you can make changes to the default operations of the LDAP Bridge by tuning it.

Logging

There are multiple types of logging available with the LDAP Bridge:

Activity logging

Activity logging logs the activity on the LDAP server. The logging information is written to the *sdir*/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 RC*START 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:

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= 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 advice when changing the RC*START 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 new date embedded in the name.new log file is created each day with a name of `slapd.log.`*`date`*. To keep the logs directory from filling up the disk, a cleanup script is provided that is stared when the LDAP Bridge starts and runs until the LDAP Bridge is shut down. It is located in `sdir/sbin/dopurge` and it reads the number of days to keep a file and the file mask including the directory containing the file. These values are read from a config file `sdir/conf/systemName/dopurge.conf`

The default settings of `dopurge.conf` are:

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

This can 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 stirng 'log.' in the data/systemName /bdb/secs directory will be purged"

Synchronization logging

Synchronization logging is enabled by default for HP Select Identity reconciliation. Both the activation of the synchronization logging and location of the synchronization log (replog.dat) are controlled by the REPLOG setting in the racf2ldap.conf, tss2ldap.conf, or acf22ldap.conf file. By default the replog.dat file is located in *sdir*/data/*systemName*/replog.dat

Logs are stored in a file called racf2ldap.log, tss2ldap.log, or acf22ldap.log. Every day at midnight the logging is redirected to a new file with the new date embedded in the name.new log file is created each day with a name of racf2ldap.log.*date*, tss2ldap.log.*date*, or acf22ldap.log.*date*. To keep the logs directory from filling up the disk, a cleanup script is provided that is stared when the LDAP Bridge starts and runs until the LDAP Bridge is shut down. It is located in sdir/sbin/dopurge and it reads the number of days to keep a file and the file mask including the directory containing the file. These values are read from a config file sdir/conf/systemName/dopurge.conf

The default settings of dopurge.conf are:

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

This can means that:

"All files older than 30 days containing the stirng '.log.' in the logs/systemName directory will be purged"

"All files older than 30 days beginning with the stirng 'log.' in the data/systemName /bdb/secs directory will be purged"

LDAP Server Configuration files

Managing Archived RACF, Top Secret, and 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.

Setting the RETAIN parameter

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

- racf2ldap.conf controls how RACF SMF records are archived by the Synchronization Daemon
- tss2ldap.conf controls how Top Secret SMF records are archived by the Synchronization Daemon
- acf22ldap.conf controls how ACF2 SMF records are archived by the Synchronization Daemon

To set the RETAIN parameter:

1. Open the desired file, located in: *sdir*/conf/*systemName*/
2. Set the RETAIN parameter to the appropriate setting:
   - 0 = SMF records are written to acf22ldap/old, racf2ldap/old, or tss2ldap/old, and are not deleted, depending on the version of the LDAP Bridge that you are working with.
   - -1 = SMF records are deleted once they are processed and are not written to acf22ldap/old, racf2ldap/old, or tss2ldap/old.
   - nn = SMF records are written to acf22ldap/old, racf2ldap/old, or tss2ldap/old, and records older than *nn* days are deleted where *nn* is a number between 0 and 999.

Encryption (SSL/TLS)

The LDAP Bridge supports encrypted LDAP communications using the Secure Sockets Layer (SSL) and its successor, Transport Layer Security (TLS). Implementing SSL/TLS has a negative performance impact, that you must consider before deciding to use encryption.

Performance Implications

Encrypting all LDAP communications increases resource utilization and response times, often 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.

To test and implement encryption, refer to the sections below:

Select an Encrypted Port

Edit the *sdir*/conf/*systemName*/site.variables file and change the sslport parameter to the desired port. The customary 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 running the LDAP Bridge.

Import the Test Digital Certificate

As delivered, the LDAP Bridge has three certificate files that enable the LDAP Bridge to test encrypted communications with authorized clients. These certificates are meant only for testing purposes. To implement SSL/TLS in production, you will need to order your own LDAP Bridge certificate from a recognized certificate authority. To test, however, you can use the files delivered in the *sdir*/conf/*systemName*/certs directory:

- ca_cert.pem
- server_cert.pem
- server_key.pem

To establish an SSL/TLS session, the LDAP Bridge presents the client with its connector certificate. The client then validates that certificate based on its own store of trusted Certificate Authorities (CAs). To test SSL/TLS, you will have to import the "OmniDAP Development" CA certificate into this store, so that the client will trust the connector certificate. The *sdir*/conf/*systemName*/certs/ca_cert.pem contains this test CA certificate.

To import the test digital certificate:

1. Download *sdir*/conf/*systemName*/certs/ca_cert.pem to the client platform, specifying EBCDIC-ASCII translation.
2. The method of importing the certificate varies depending on the platform that you are working on.
   - If you are testing from the address book on Microsoft Windows, for example, you can open Microsoft Internet Explorer and select **Tools**--> **Internet Options** --> **Content** --> **Certificates** --> **Import Menu Options** to import ca_cert.pem into your trusted root certificate authorities store. After you import the certificate, you will see the "OmniDAP Development" certificate in this store. This will allow you to test SSL/TLS encrypted communications from your Windows address book.
   - Other platforms and applications can require you to import ca_cert.pem into the cert7.db file or some other certificate store. Reference the appropriate documentation for the client platform to determine how to import this CA certificate.
3. Once you have imported ca_cert.pem into the platform specific certificate store, make sure that the calling application is referencing this store. The LDAP tab of the Directory Setup dialog shows the name of the certificate store.

Ordering your Own Connector Certificate

To implement the LDAP Bridge in production, your LDAP Bridge must use its own site-specific certificate. To obtain a certificate, you can order it from a variety of certificate authorities, including www.thawte.com, www.verisign.com, and www.rsasecurity.com.

After you have obtained your certificate, you must store the certificate, its private key, and the CA certificate in a USS directory or in MVS data sets. You must update the sladp.conf file directives with the names and locations of the certificate, its private key, and the CA certificate

If you choose to store them in USS, you should store them in a directory outside of *sdir* so that they will be unaffected by any maintenance that is applied to the LDAP Bridge.

If you choose to store them in MVS data sets, the syntax for specifying the location in the slapd.conf is as follows

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

for example

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

These files must all be in base64 format (also sometimes referred to as PEM format):

- **ca_cert.pem** - The Certificate Authority (CA) certificate for the CA that issued the connector certificate. You can usually acquire this file directly from the CA web site.
- **server_cert.pem** - The connector certificate presented to clients during the SSL/TLS handshake to verify connector identity and establish trust. This certificate must be signed by the CA referred to by the CA certificate, above.
- **server_key.pem** - The connector private key used to establish the session key and encrypt communications with the client. This file is generated during the certificate request.

Security for SSL/TSL

To implement SSL/TLS in production, protection of *sdir*/conf/*systemName*/server_key.pem becomes very important. Unauthorized read access to this key could enable decryption of communication, impersonation of the connector or other security breaches. To allow only the user ID of the connector to have access to these files you can use the following commands:

```
cd sdir/conf/systemName/certs
chown userid ./server_key.pem
chmod 0400 ./server_key.pem
```

Where *userid* is the Database Admin ID for the LDAP Bridge.

SSL/TLS Parameters in Slapd.conf

The following parameters in *sdir*/conf/*systemName*/slapd.conf control SSL/TLS functionality. If you change the file names of any of the SSL/TLS-related files in *sdir*/conf/*systemName*, you must modify these parameters in slapd.conf correspondingly.

Parameter	Description
TLSRandFile	The path to the entropy seed used to generate encryption keys. This file (default: *sdir*/logs/*systemName*/entropy.rnd) is generated at start-up by the doslapd script.
TLSCACertificateFile	The path to the Certificate Authority Certificate, in base64 format. The delivered value is *sdir/conf/systemName/certs/*ca_cert.pem. If you want to use a CA other than the CA that comes with the LDAP Bridge for testing purposes, you can either append it to this file or place it in a new file. If you do the latter, you must modify this parameter to point to this new file.
TLSCertificateFile	The path to the Connector Certificate, in base64 format. The delivered value is *sdir/conf/systemName/certs/*server_cert.pem. If you order your own connector certificate, you can either replace server_cert.pem with the new connector certificate (in base64 format), or place the new connector certificate into a new file. If you do the latter, you must modify this parameter to point to this new file.

Parameter	Description
TLSCertificateKeyFile	The path to the Connector Certificate Private Key, in base64 format. The delivered value is *sdir*/conf/*systemName*/certs/server_key.pem. If you order your own connector certificate, the certificate request generates a private key file. You can either replace the contents of server_key.pem with the new private key (in base64 format), or place the new private key into a new file. If you do the latter, you must modify this parameter to point to this new file.
TLSCipherSuite	The client ciphers that the connector will accept. The delivered value allows the connector to accept high and medium strength ciphers, this is sufficient for most uses.
TLSVerifyClient	Specifies whether the connector will require client certificate authentication. As delivered, this is set to never.

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 *sdir*/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 *sdir*/conf/*systemName* directory, the slapd.conf file contains the following online configuration parameters for your site.

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 RACF, Top Secret, or 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).
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:

- RACF security system - slapd.racf.conf
- Top Secret security system - slapd.tss.conf
- ACF2 security system - slapd.acf2.conf

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

Parameter	Description
Database	This parameter must always be set to “bdb”.
Suffix	The LDAP root entry for the LDAP Bridge. There must be one suffix parameter: `o=%company%`
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%` • 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 *sdir*`/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.

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 the *sdir*`/conf/`*systemName*`/slapd.racf.conf` file, the *sdir*`/conf/`*systemName*`/slapd.tss.conf` file, or the *sdir*`/conf/`*systemName*`/slapd.acf2.conf file`, depending on the version of the LDAP Bridge 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 *sdir*/conf/*systemName* contain UNIX environment variables that affect batch and online processing:

- **stdenv.slapd** - Affects online connector processing (RC*START).
- **stdenv.slapadd** - Affects database load processing (RCZCONVT, RCXCONVR, RCYCONVA)
- **stdenv.racf2ldap, stdenv.tss2ldap, stdenv.acf22ldap** - Affects online connector processing (RC*STT2L)
- **stdenv** - Affects all other processing (RC*STOP, etc.)

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

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)”.

Parameter	Description
_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)”.
_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.
LDAPBRIDGE_LOACALE=*locale.code page*	Specifies that characters from code pages other than IBM-1047 can be processed by the LDAP Bridge. By default stdenv.slapd does not have this parameter listed and will use code page 1047 by default. This parameter must be added to the stdenv.slapd, stdenv.racf2ldap (for the racf2ldap plug-in), stdenv.tss2ldap (for the tss2ldap plug-in), and stdenv.acf22ldap (for the acf22ldap plug-in) files to enable processing of characters from code pages other than IBM the 1047 codepage. You must specify a code page that is supported by the security database that you are working with (such as RACF Top Secret, or ACF2). For example: LDAPBRIDGE_LOACALE=Fr_FR.IBM-297

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 *sdir*/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/man.cgi?query=slapd.access&sektion=5&apropos=0&manpath=OpenLDAP+2.2-Release

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 read
```

Where:

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.
	`read`	Read 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 read access to the database.

Example 2

The LDAP Bridge uses the following default ACL definition:

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

Where:

ACL Variable	Syntax	Meaning
db entries	*	Represents all user entries that are contained in the database.
ldap attr	attrs=*%secs%Password*	%secs%Password represents racfPassword, acf2Password or tssPassword, depending on the mainframe security database that you are using.
user/group	*	everyone.
permitted action	none	By default, no one has any access

The purpose of this ACL is to restrict these attributes that are used in Password Synchronization. The administrators must make careful decisions as to who has access to this feature.

Allowing All Users and Groups Read Access to 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:

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	read	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:

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

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 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:

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:

ACL Variable	Syntax	Meaning
db entries	*	Wildcard character that represents all database entries.
ldap attr	none	
user/group	*group/ tssGroup/ member.regex=* "cn=*.*,ou=gr oups,o=*compan y*"	*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:

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.

Limiting Database Access to Specific Entries or Attributes

You can choose to restrict what users and groups can view in the database to protect sensitive information in the database. You can use an ACL definition that limits users and groups to specific entry types and entry attributes. These protection schemes are intended to work with another, more specific, ACL definition that allows administrative users to view the entire database.

Example 1

To limit authenticated users' read access to user entries, use an ACL definition similar to the following:

```
access to dn.onelevel="ou=people,o=company"
by users read
```

Where:

ACL Variable	Syntax	Meaning
db entries	dn.onelevel="ou=people,o=*company*"	Represents all user entries contained within the database. *company* represents the root dn you specified for the LDAP Bridge.
ldap attr	none	
user/group	users	Represents authenticated users.
permitted action	read	Allows users to read the specified database entries.

Example 2

To limit authenticated users read access to group entries, use an ACL definition similar to the following:

```
access to dn.onelevel="ou=groups,o=company"
by users read
```

Where:

ACL Variable	Syntax	Meaning
db entries	dn.onelevel="ou=groups,o=*company*"	Represents all group entries contained within the database. *company* represents the root dn you specified for the LDAP Bridge.
ldap attr	none	
user/group	users	Represents authenticated users.
permitted action	read	Allows users to read the specified database entries.

Example 3

To limit authenticated users read access to a specific entry attribute, use an ACL definition similar to the following:

```
access to dn.onelevel="ou=people,o=company" attrs userName,userPassword
by users read
```

Where:

ACL Variable	Syntax	Meaning
db entries	dn.onelevel="ou=people, o=*company*"	Represents all user entries contained within the database. *company* represents the root dn you specified for the LDAP Bridge.
ldap attr	*userName*	*userName* represents the user name entry attribute.
	userPassword	*userPassword* represents the user password entry attribute.
user/group	users	Represents authenticated users.
permitted action	read	Allows users to read the specified database entries.

Tuning the LDAP Database

The LDAP Server users the open source BDB (Berkeley DB) as the back end to store the LDAP data. The there are several configuration files that govern the behavior of slapd.

DB_CONFIG: database variables

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

- **DB_CONFIG.slapd** - Affects online connector processing (RC*START).
- **DB_CONFIG.slapadd** - Affects database load processing (RCZCONVT, RCACONVA, RCXCONVR)

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

Parameter	Description
set_cachesize	Controls the size of the cache. The format is: `set_cachesize` *`gigabytes, bytes number_of_caches`* *`gigabytes`* must be set to 0. *`bytes`* must be the size of *`sdir`*`/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.
	DB_CONFIG.slapadd has the following settings that address a locking issue during the LDAP database load when the following message is logged:"Lock table is out of available locks"
	The following flags control locking: larger values increase the locking capacity of the system, at the expense of greater memory usage:
	`set_lk_max_locks 60000`
	`set_lk_max_lockers 1000`
	`set_lk_max_objects 60000`

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 commenting out 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 comments 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 *secs*`2ldap.conf` configuration file, where *secs* is acf2, racf, tss depending on the version of the LDAP Bridge that you are using.

Synchronization Daemon General Definitions

The following parameters control the global functioning of the Synchronization Daemon.

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.ldif	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	3	Specifies the number of retry attempts for a non-responsive LDAP Server
LOGLEVEL	4	Log level for event details in LOGDIR/replog.dat.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
CONVERTLOG LEVEL	0	Logging for the database build process, when working with TopSecret or ACF2.

Parameter	Default Value	Description
RETAIN	30	Specifies how records are to be written to `acf22ldap/old`, `racf2ldap/old`, or `tss2ldap/old`, depending on the version of the LDAP Bridge that you are working with. Values are: • 0 = SMF records are written to `acf22ldap/old`, `racf2ldap/old`, or `tss2ldap/old`, and are not deleted. • -1 = SMF records are deleted once they are processed and are not written to `acf22ldap/old`, `racf2ldap/old`, or `tss2ldap/old`. • nn = SMF records are written to `acf22ldap/old`, `racf2ldap/old`, or `tss2ldap/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 *secs*2ldap/old by function of the RETAIN parameter.
NOTIFY	tssmanager@%company% CONSOLE operations@%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
SSLPORT	%sslport%	Configured at run time from site.variables, tells Synchronization Daemon the SSL port to use at the LDAP Server
LDAPVERSION	3	Specifies the supported LDAP version. Do not change this parameter.
ORGDN	*o=%company%*	Configured at run time from site.variables LDAP Root.
MANAGERDN	cn=*%secs%*Manager,	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.
SQUAL	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 SQUAL.JCLLIB(TSSCFILE)

Tuning the MVS data sets

The ATTR file

The *SQUAL*.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 *SQUAL*.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 RC*CMPLK in the JCLLIB. The following table summarizes the delivered sample programs:

Members	Language	Description
RCYCONVAF	COBOL	Filter installation exit called by RCYCONVA, the ACF2 conversion process. Filters the ACF2 profiles loaded into the LDAP directory. By default, RCYCONVA 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 *SQUAL*.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.
RCYCONVAU	COBOL	Rule installation exit called by RCYCONVA, 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.
RCXCONVRF	COBOL	Filter installation exit called by RCXCONVR, the RACF conversion process. Filters the RACF profiles loaded into the LDAP directory. By default, RCXCONVR 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 *SQUAL*.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.

Members	Language	Description
RCXCONVRU	COBOL	Rule installation exit called by RCXCONVR, 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.
RCZCONVTF	COBOL	Filter installation exit called by RCZCONVT, the Top Secret conversion process. Filters the Top Secret profiles loaded into the LDAP directory. By default, RCZCONVT 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 *SQUAL*.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.
RCZCONVTU	COBOL	Rule installation exit called by RCZCONVT, 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 RC*START, RC*STOP, RCYCNVA, RCXCNVR and RCZCNVT jobs:

- *SQUAL*.JCLLIB
- *SQUAL*.SRCLIB
- *SQUAL*.LOADLIB
- *SQUAL*.ATTR

The 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.

LDAP Database Refresh

An LDAP Database refresh can be accomplished in two ways.

- Run the LDAP database load jobs RCXCNVRX, RCYCNVAX, or RCZCNVTX.

  These jobs allow the LDAP Bridge to stay running while the database is refreshed. First, these jobs disable the Synchronization Daemon. The LDAP server will still accept process requests from HP Select Identity and make the changes at the mainframe security database, but the audit records will accumulate in the `../new` directory. Next these jobs unload the mainframe security database and convert the unloaded file to an ldif file, then the ldif file is loaded into a new LDAP database in a different directory than the production LDAP database that is still running. When this new staging database is built, the LDAP Bridge is momentarily shut down and brought up using the newly created LDAP database. This restarts the Synchronization Daemon which immediately begins to process the audit records that have accumulated during this process.

- Shut down LDAP Bridge, then resubmitted the LDAP database load job RCXCNVR, RCYCNVA, or RCZCNVT and restart the LDAP Bridge when the load is done

  Because running the LDAP database load job RCXCNVR, RCYCNVA, or RCZCNVT requires the LDAP Bridge to be shut down, there will be an outage for HP Select Identity transactions going to the mainframe. These transactions must be reissued after the build finishes and the LDAP Bridge is restarted. Transactions that come into the mainframe security system from other sources will be logged by the audit records and synchronized into the LDAP Bridge when it restarts.

A Internationalization

The locale for the LDAP Bridge is set by the configure script. To change the locale, re-run the configure script. Before re-running the configure script, make a note of any manual customizations you have performed so that you can re-apply them afterwards.

B Troubleshooting

This appendix contains troubleshooting information.

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 *sdir/data/systemName/plugin*/new directory to the *sdir/data/systemName/plugin*/old or *sdir/data/systemName/plugin*/error directories, where:

- **/old** acts as an archive of 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. The LDAP Bridge cache must be rebuilt using the RC*CONVT job.

tss2ldap.conf Error Definitions

This section of tss2ldap.conf describes how the Synchronization Daemon should handle various LDAP error conditions returned from the LDAP Bridge when working with the tss2ladp 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* - The text message associated with the LDAP_error_code, included for descriptive purposes only.
- *code*- The standard LDAP error code returned from the connector.
- *level* - The Synchronization Daemon severity level for this error code: WARNING, SERIOUS, SEVERE or FATAL. See NOTIFYLEVEL, above.
- *action*- The 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 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, follow the series of steps below:

1 Edit *sdir*/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 *sdir*/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.

If you adjust the heap size upwards, you will also have to adjust the REGION parameter in the RC*START JCL, as described in Ensuring Sufficient Region Size on page 19.

Collecting Diagnostic Information Using the dodiag Script

During support incidents, the support team could need any number of config files, logs, audit records. To simplify the collection of these files, a script called dodiag is provided in *sdir*/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 *sdir*/conf directory including all system specific subdirectories.
- the *sdir*/logs directory including all system specific subdirectories.
- a recursive listing of *sdir*, *sdir*/conf, *sdir*/data, *sdir*/logs, *sdir*/sbin

Notes:

- 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'
- 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 sdir directory.

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 *sdir*/conf/SYSA directory including all system specific subdirectories.
- the *sdir*/logs/SYSA directory including all system specific subdirectories.
- a recursive listing of *sdir*, *sdir*/conf, *sdir*/data, *sdir*/logs, *sdir*/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 *sdir*`/conf/SYSA` directory including all system specific subdirectories.
- the *sdir*`/data/SYSA` directory including all system specific subdirectories.
- a recursive listing of *sdir*`,` *sdir*`/conf,` *sdir*`/data,` *sdir*`/logs,` *sdir*`/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 HPSI reconciliation 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 *sdir*`/conf/SYSA` directory including all system specific subdirectories.
- the *sdir*`/data/SYSA` directory including all system specific subdirectories.
- a recursive listing of *sdir*`,` *sdir*`/conf,` *sdir*`/data,` *sdir*`/logs,` *sdir*`/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 *sdir*`,` *sdir*`/conf,` *sdir*`/data,` *sdir*`/logs,` *sdir*`/sbin`

Example 6

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

This results in a file test5.pax.Z containing:

- all the files from sdir and the MVS data sets and a scrambled version of the ldif file extracted from the mainframe security database.
- a recursive listing of *sdir*`,` *sdir*`/conf,` *sdir*`/data,` *sdir*`/logs,` *sdir*`/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*.

C Uninstalling the LDAP Bridge

You can uninstall the LDAP Bridge.

To uninstall the LDAP Bridge

1. Shut down the LDAP Bridge server using the RC*STOP member of *SQUAL*.JCLLIB.
2. Uninstall the appropriate exit. When working with:
   - 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 HPSI 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
       ```
     - Undo any changes made to the PROGnn member of the SYS1.PARMLIB related to SLAPU83A
3. Delete the *SQUAL*.JCLLIB, *SQUAL*.LOADLIB, *SQUAL*.SRCLIB, *SQUAL*.EXITLIB, and *SQUAL*.ATTR data sets.
4. Remove the *sdir* directory in an OMVS session with the command:
   ```
   rm -r  sdir
   ```

Index

S

T

U

Z