HP OpenView Select Identity
CA-ACF2 LDAP Bridge

For the z/OS® Operating System

LDAP Bridge Version: 3.3.1

Installation and Configuration Guide

Document Release Date:November 2006
Software Release Date: November 2006

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/). Portions Copyright (c) 1999-2003 The Apache Software Foundation. All rights reserved.

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

Trademark Notices

Support

Please visit the HP OpenView support web site at:

http://www.hp.com/managementsoftware/support

This web site provides contact information and details about the products, services, and support that HP OpenView offers.

HP OpenView online software support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support site to:

- Search for knowledge documents of interest
- Submit enhancement requests online
- Download software patches
- Submit and track progress on support cases
- Manage a support contract
- Look up HP support contacts
- Review information about available services
- Enter 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 log in. Many also require a support contract.

To find more information about access levels, go to:

http://www.hp.com/managementsoftware/access_level

To register for an HP Passport ID, go to:

http://www.managementsoftware.hp.com/passport-registration.html

1 Introduction

The HP OpenView Select Identity CA-ACF2 LDAP Bridge (LDAP Bridge) is an LDAP gateway that provides access to the CA-ACF2 database (ACF2). By enabling you to access mainframe security data with LDAP, the LDAP Bridge allows you to extend mainframe authentication, authorization, administration, and provisioning to HP Select Identity.

Audience

This guide is intended for security administrators and system programmers. These personnel must be 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
- ACF2 concepts such as password verification and resource authorization

These personnel must have authority to:

- Edit mainframe files, create data sets, and submit jobs
- Access USS, enter UNIX commands, and create HFS files
- Create data sets and HFS files in ACF2

Overview of the LDAP Bridge

The LDAP Bridge provides an LDAP interface to ACF2 that transforms the mainframe security repositories into LDAP directories. The LDAP Bridge makes this data available to your environment through LDAP. Using the LDAP Bridge, you can use ACF2 information to authenticate users and authorize access to resources. The LDAP Bridge consists of the following components:

- An LDAP server that publishes a copy of the ACF2 database
- A plug-in for the LDAP server that provides ACF2 administrative and other functions
- A daemon that keeps the LDAP directory and ACF2 database in synch.

LDAP Server

The LDAP server publishes a copy of the ACF2 database.The database copy that is published is a real-time image of the entire ACF2 database as it resides on the host z/OS system.

LDAP Command Translator

The LDAP command translator modifies ACF2 to reflect the changes that were initiated within the LDAP Bridge. Whenever users make a change to the ACF2 database, the LDAP command translator transforms the LDAP modify command into an equivalent ACF2 command so that the ACF2 database is modified accordingly. When the change has been made to the ACF2 database, the synchronization daemon processes and reflects the change in the mirror database.

Synchronization Daemon

The synchronization daemon updates the database copy to reflect the current status of the ACF2 database. Whenever a change is made to the ACF2 database, the LDAP command translator intercepts the audit record that is generated by the ACF2 command. If the LDAP Bridge is stopped, ACF2 changes accumulate until it is restarted.

2 Installing and Configuring the LDAP Bridge

System Requirements

The following requirements are necessary to install and use the LDAP Bridge.

Software Requirements

The LDAP Bridge requires the following elements:

- IBM z/OS 1.4 or later
- CA-ACF2
- Functional Requirements

The LDAP Bridge runs under UNIX System Services (USS), and uses TCP/IP to communicate with remote clients.

Before You Begin

Selecting the Install 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 and so on, must be valid on the second machine.

Multi System Installation

If you plan to share file systems between two or more z/OS systems on which the LDAP Bridge is installed, you must perform multi-system installation. The multi system installation ensures that naming conflicts will not occur when sharing a file system. organizes configuration, data, and log files into separate subdirectories identified by the run-time system name, in order to avoid naming conflicts between the files used when running on different systems.

A difference from the single-system install is that the directories conf, logs, sbin, and data will each have a subdirectory with the system name entered during the configuration. These system-named subdirectories will hold the configuration files, logs, and data that will be used by that system (sbin will have the binaries, sbin's subdirectory will be for user customized/developed binaries.) The MVS data sets will also bear the system name in the high level qualifier in addition the high level qualifier supplied at the prompt during the configuration. The members of the MVS data set will be tailored with the system name where appplicable.The configure script can then be run again specifying the values that will be valid on a second system. This will result in a second set of subdirectories in conf, logs, sbin, and data with the second system name and a second set of MVS datasets also tailored with the second system name. The configure script is then run once for each system accumulating a subdirectory in conf, logs, sbin, and data for each system. The HFS mount points for conf, logs, sbin, and data can then be mounted to the systems specified during the configuration and the MVS datasets can be transferred to the appropriate systems. When the JCL to start the job is submitted on any of these systems, the LDAP Bridge content in the corresponding subdirectories of conf, logs, sbin, and data will be used.The multi-system configuration allows maintaining a single installation of the binaries for ease of applying maintenance. It also allows segregation with respect to storage of the conf, data, logs and sbin in whatever combination is desired.

In the multi-system configuration the LDAP Bridge is initially installed on a single system. If you choose to perform a multi system installation you must determine what the LDAP Bridge install directory will be. The LDAP Bridge install directory must be valid for all systems in the multi install. 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/hp1 on SYS1
>
> A second run of the multi-system install is desired to set up for a remote system SYS2 where the sdir will be /usr/lpp/hp2.
>
> Prior to the second run of the configure script on SYS1 from /usr/lpp/hp1, a link is created on SYS1 with the following command:

```
ln -s  /usr/lpp/hp1  /usr/lpp/hp2
```

> The second run of the configure script will prompt with the actual directory /usr/lpp/hp1
>
> The user can enter the name of the remote directory /usr/lpp/hp2.

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.

Preparing Your Environment

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

User IDs

The user ID that is used to install, configure, and run and the LDAP Bridge must have the appropriate authorities. You can use one or multiple user IDs to install, configure and run the LDAP bridge. The function and required permissions for the user ID that is used to preform that function is listed below. If company policies allow all or any of the tasks to be performed by the same id the install process can be simplified

- The user ID that is used to install and configure the LDAP Bridge (Install ID)- This user must be able to login to USS and create directories
- The user ID that is used to submit the JCL to build the LDAP database (ACF2 Admin ID). This user ID must:
  - be authorized to read an ACF2 logonid databse backup file to extract the records from ACF2
  - be authorized to write to the USS directories that are created by the install user
  - have an OMVS Segment
  - a member of the same group as group owner of the USS directories
- 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 and write to directories in USS
- The User ID that is used by OVSI to connect to the LDAP database and administer the ACF2 data (OVSI Admin ID). This user must have an OMVS segment and the TSS authority to run the set of commands that are needed by OVSI.

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:

  Disk Space = 200MB + (size of ACF2 backup file x 3.2)
- ensure that the parent directories of the LDAP Bridge have execute access permission for OTHE - For example, 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 this directory, ld 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 must have appropriate permissions:
  - OWNER: read/write/execute
  - GROUP: read/write/execute
  - OTHER: execute

  If, for example, you are installing the LDAP Bridge (hpv33a) into the /usr/lpp/hpv33a directory, assign the appropriate permissions by issuing the following commands:

  ```
  chmod 0771 /usr/lpp/hpv33a
  ```

- The group owner of the hpv33a directory must be a ACF2 group that the user ID that is associated with the LDAP Bridge started task is a member. If, for example, the hpv33a directory is /usr/lpp/hpv33a, and you plan to run the LDAP Bridge under a user ID that is a member of the ADMIN group, then the group owner of the /usr/lpp/hpv33a must be ADMIN. To see the group owner of /usr/lpp/hpv33a, issue the following command:

  ```
  ls -ld /usr/lpp/hpv33a
  ```

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

  ```
  chgrp ADMIN /usr/lpp/hpv33a
  ```

  The person installing the product must also be a member of this group.

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.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.
- Port 623, or the appropriate port 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 200 megabytes of memory.

The default REGION parameter coded in the JCL is 0M, which usually indicates no memory limitations. However, at your site, there could be specific limitations that apply regardless of the REGION=0M parameter. These limitations, usually coded in an IEFUSI user-exit, could 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 200 megabytes or more. If a process fails to allocate memory, it will exit with a return code 9. This indicates that the region size is too small and needs to be adjusted upwards.

Verify ACF2 Privileges

The LDAP Bridge LDAP executable must be APF-Authorized and Program-Controlled to perform authentications against ACF2. In order to create the required permissions, you must first ensure that you have ACF2 access to the following:

- BPX.FILEATTR.PROGCTL Facility Class
- BPX.FILEATTR.APF Facility Class

Installation Overview

The CD or downloaded version of the LDAP Bridge release media contains the compressed file hpv33a.pax.Z, that is used to install the LDAP Bridge onto an HFS file system. After the initial 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 and then makes customized versions of the files from the install directory using the values input at the prompts. These customized files along with the binaries are placed in the conf, data, and sbin directories. A log of the install process is placed in logs. During the configure script, a set of MVS data sets are created. JCL, LOAD and Source members from the install directory that have also been customized using the values input at the prompts are copied into the data sets.Among values input at the prompts are directory paths, port numbers, and system names that will be specific to the installation machine.

The multi system configuration path names are referenced in this document. For example reference to the slapd.conf file would be sdir/conf/slapd.conf for a single-system configuration but sdir/conf/system/slapd.conf for a multi-system configuration. In this case this document would use sdir/conf/system/slapd.conf.

Installing the LDAP Bridge

1. Transfer the product media to the machine where you want to install the LDAP Bridge. Transfer the hpv33a.pax.Z file using FTP to your HFS directory. During the transfer, be sure to specify binary mode.
2. Expand the PAX file. Enter OMVS from TSO, and issue the following commands:

```
cd sdir
pax -rv -px -f hpv33a.pax.Z
```

where ***sdir*** is the name of the HFS directory you created for the LDAP Bridge.

Configuring the LDAP Bridge

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

- Prompts the user 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 files under z/OS
- Moves the source, load, JCL, and attributes file from UNIX System Services to z/OS
- Frees the file allocations for EXITLIB, SRCLIB, LOADLIB, JCLLIB, and ATTR
- Installs the LDAP Server and configuration data base along with the synchronization daemon and LDAP command translator.

Running the Configuration Script

The first time that the configuration script is run, you are queried for site-specific information that is used to create the file. Exiting the script before providing any information will create a file that uses default values for all of the variables listed below. 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.

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.

The configuration script is located in *sdir*, where *sdir* is the HFS directory you created for the LDAP Bridge. If you are using the default installation directory, the configuration script is located in */usr/lpp/hpv33a*. To run the installation script, enter OMVS from TSO, then issue the following commands:

```
cd sdir

sh configure
```

During the configuration you will need to supply the following site-specific information:

- Do 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. Enter the name of the system to you want to configure (default = <system name>). Where <system name> is the system name that was discovered by LDAP Bridge configuration script. This question only appears when you are doing a multi-system configuration.
- (1 of 9) Enter the (case sensitive) name of the UNIX HFS directory for this product (default = 'the directory from which the configure script is being run'):

  Specify the directory path for the LDAP Bridge install directory. The default value is discovered by the configure script. You can use another path name as long as it resolves to the same directory from which the configure script is being run. The directory path will be used for the sdir variable in various files.

- (2 of 9) Enter the high level qualifier(s) for the MVS data sets for this product (default = USERID.HPV33A):

  The high level qualifier will be used for the SQUAL variable in various files, scripts and data set members and will be the high-level qualifier for the LDAP Bridge data sets. Enter a value that conforms to your site standards. It is recommended that you preserve the second-level qualifier as HPV33A.

  In a Multi-system configuration the system name will automatically be appended to the HLQ supplied here. Accepting the default in a multi-system configuration will result in an HLQ of USERID.HPV33A.SYSTEM where USERID is the logon id of the user running the script and SYSTEM is value supplied for System name.

- (3 of 9) Enter the name of a permanent disk unit for the MVS data sets for this product (default = 3390):

  The permanent disk unit value will be used for the PDUNIT variable during the configuration process.

- (4 of 9) Enter the name of temporary disk unit for temporary MVS data sets created during the operation of this product (default = SYSALLDA):

  The temporary disk unit value will be used for the TDUNIT variable during the running of the LDAP Bridge.

- (5 of 9) Enter the LDAP root (default =):

  The LDAP root will be used in the LDAP commands from OVSI. This value is case sensitive, and is often the Internet domain name of your organization. The value used is at the customer's discretion.

- (6 of 9) Enter the host name or IP address for the LDAP server (default = 'host name discovered by script'):

  This hostname is used by the synchronization daemon acf22ldap to connect to the LDAP server that it is synchronizing. On a multi-system configuration it will be necessary to supply an appropriate address for the system being configured. In general if acf22ldap and the LDAP server are to be running on the same machine then the loopback address of 127.0.0.1 can be used.

- (7 of 9) Enter the port number for unencrypted connections (default = 389):

  Specify the port number for unencrypted connections the LDAP server. You can enter "0" for either the SSL or unencrypted port to disable the of connection. The standard LDAP default port for unencrypted connections is 389, but you can enter any unused port number. You must enter different values for each port.

- (8 of 9) Enter the port number for SSL connections (default = 636):

  Specify the port number for SSL-encrypted connections the LDAP server. You can enter "0" for either the SSL or unencrypted port to disable the of connection. The standard LDAP default port for SSL-encrypted connections is 636, but you can enter any unused port number. You must enter different values for each port.

- (9 of 9) Enter the name of the locale used for the security database (default = 'discovered by script, if none discovered then default to en_US.IBM-1047'):

  You must specify the name of the locale that describes the way that data is stored in the security database. The locale specifies the language and code page of the data. In general, the locale that you choose should match the locale used in your 3270 terminal settings when viewing or editing security data (for example, when using TSO commands or ISPF panels to perform security database functions).

Information: If you want to change any of the configuration options that were specified during the initial execution of the configuration script, re-run the configuration script. The configuration script can be re-run as many times as necessary.

Installing the ACF2 Installation Exit

Acf22ldap runs as a stand-alone UNIX daemon in a separate address space from the CA-ACF2 LDAP Bridge. It reads the SMF records generated whenever CA-ACF2 changes are made, and propagates the changes to the CA-ACF2 LDAP Server using LDAP. The SMF records are written to the sdir/data/system/acf22ldap/new directory by the SLAPU83AA program that runs in the user exit points SYSSTC.IEFU83, and either SYSTSO.IEFU83, SYSJES2.IEFU83, or SYS.IEFU83.

To use acf22ldap, you must activate the SMF user exits decribed below..

Enabling the IEFU83 User Exit Points

To install this exit, perform the following steps:

Before implementing the acf22ldap IEFU83 user-exit program, you must verify that user-exit points are enabled on your system for the following environments:

- Started Tasks - SYSSTC.IEFU83 user-exit point
- SYSTSO.IEFU83, SYSJES2.IEFU83, or SYS.IEFU83 user-exit point

LDAP BridgeWhether the CA-ACF2 LDAP Bridge requires the SYSTSO.IEFU83, SYSJES2.IEFU83, or SYS.IEFU83 user-exit point depends on your system configuration:

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

The sections below explain how to determine which SMF subsystems are defined in your environment. 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 ExitsManual.

This information is available from IBM online at the following location:

http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/IEA2E410/2.28?FS=TRUE&SHELF=IEA2BK11&DT=20010627160030

To To enable the required exit points, follow the series of steps below:

1 Edit the SMFPRMnn member of the SYS1.PARMLIB data set, where nn is the SMF

parameter member currently active on your system.

26 Chapter 3

2 Verify that IEFU83 is specified in the EXITS clause of the SUBSYS(STC) parameters. For

example:

SUBSYS(STC,EXITS(IEFU83,xxx))

where xxx represents other keywords and parameters used in your environment.

3 If TSO is defined as a separate SMF subsystem, then this member contains a statement

starting with "SUBSYS(TSO)". In this case, verify that IEFU83 is specified in the EXITS

clause parameters. For example:

SUBSYS(TSO,EXITS(IEFU83,xxx))

where xxx represents other keywords and parameters used in your environment.

4 If JES2 is defined as a separate SMF subsystem, then this member contains a statement

starting with "SUBSYS(JES2)". In this case, verify that IEFU83 is specified in the EXITS

clause parameters. For example:

```
SUBSYS(JES2,EXITS(IEFU83,xxx))
```

where xxx represents other keywords and parameters used in your environment.

5 If neither TSO nor JES2 are defined as separate SMF subsystems, verify that IEFU83 is

specified in the EXITS clause parameters for the SYS statement. For example:

```
SYS(xxx,EXITS(IEFU83,xxx)xxx )
```

where xxx represents other keywords and parameters used in your environment.

Activating the IEFU83 Dynamic User-exit Program

The procedure for activating a dynamic IEFU83 user-exit program is described in the IEFU83 section of the IBM z/OS MVS Installation Exits Manual. This information is available fromIBM online at the following location:

http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/ IEA2E410/
2.28?FS=TRUE&SHELF=IEA2BK11&DT=20010627160030

Activating SLAPU83A Dynamically

The SLAPU83A 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 SMFsubsystems in your SMF parameter file:

If TSO is defined as a separate SMF subsystem, use the command that referencesSYSTSO.IEFU83.

If JES2 is defined as a separate SMF subsystem, use the command that referencesSYSJES2.IEFU83.

If neither TSO nor JES2 are defined as separate SMF subsystems, use the command that references SYS.IEFU83. Activating user-exit points via 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 PROGnn member of the SYS1.PARMLIB data set, where nn is the programparameter member currently active on your system.

2 Add the following statements:

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

and either:

```
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 CA-ACF2 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 statementsthat reference SYS.IEFU83.Once the PROGnn member has been edited in SYS1.PARMLIB, it may have to be activated by editing the COMMNDnn member to include the following statement:COM='SET PROG=nn'where nn corresponds to the suffix for the PROGnn member.

Loading the LDAP Directory

The LDAP Bridge uses a directory database that is populated with data from your ACF2 repositories. Once the LDAP Bridge database is initially loaded, the LDAP Bridge synchronization daemon keeps all databases synchronized.

Populating the LDAP Bridge Database

To populate your LDAP Bridge database, you must run the SLCONVA job found in SQUAL.JCLLIB, where SQUAL is the high-level qualifier that you selected for your data sets. This job:

- reads an ACF2 logoid database backup file produced ahead of time with the ACF2 BACKUP command.You will need to supply the name of this file for the SECFILE value in the JCL.
- converts the data inteh backup file to an ldif formatted HFS file in sdir/data/system/ldif directory by invoking the <SQUAL>.LOADLIB(SLCONVA)
  Note: You may need to adjust the default SORTWK SPACE of SPACE=(CYL,(5,5)) on this step to suit the size of your ACF2 database.
- loads the ldif data into the LDAP database using the doldif script and slapadd program from the HFS sdir/sbin directory

You must use the ACF2 Admin ID to run this job. For more information on the user IDs that are required to install and configure the LDAP Bridge, see the "*User IDs*" section.

After you have made the customization changes to the SLCONVA job, submit the JCL. All steps in the SLCONVA job should return a condition code of 04 or less.

3 Running the LDAP Bridge

This chapter describes how to start, stop and test the LDAP Bridge.

Running the LDAP Bridge

You can run the LDAP Bridge as a z/OS batch job or started task using BPXBATCH

Submitted Jobs

To run the LDAP Bridge as a batch job, submit *SQUAL*.JCLLIB(START), after customizing this JCL with a job card appropriate for your site.

To stop the LDAP Bridge, submit the STOP member of the *SQUAL*.JCLLIB data set.

Started Tasks

To create started tasks that start and stop the LDAP Bridge, customize the appropriate JCL provided within the *SQUAL*.JCLLIB data set, where:

- STARTST creates a started task that starts the LDAP Bridge.
- STOPST creates a started task that stops the LDAP Bridge.

Starting the LDAP Bridge

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 START member of *SQUAL*.JCLLIB data set, then submit the job. All condition codes return as zero. The START job runs until the STOP job is submitted to bring down the LDAP Bridge.

Started Tasks

To create started tasks that start and stop the LDAP Bridge, customize the appropriate JCL that is provided within the *SQUAL*.JCLLIB data set, where:

- STARTST creates a started task that starts the LDAP Bridge.
- STOPST creates a started task that stops the LDAP Bridge.

Starting the Synchronization Daemon

The synchronization daemon starts automatically using the same 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 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 200MB 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. Similarly, the DOLDIF portion of the SLVCONVA job also requires considerable storage. Setting REGION=0M is also recommended.

However, in both these jobs, 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 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

Successful completion of the tests described above indicates that the
LDAP Bridge is running properly on your system. To conclude testing, stop the LDAP Bridge with the STOP member of the JCLLIB data set. Add job card information to the JCL, then submit the job. All condition codes return as zero.

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

dotestserver
```

3. At the prompts, enter your ACF2 user ID and password. This test returns information on your ACF2 user ID as stored in the LDAP repository.

Testing the Synchronization Daemon

To test the acf22ldap daemon by running the dotestr2l script, follow the series of steps below:

To test the acf22ldap daemon by running the dotestr2l script, follow the series of steps below:

1 Verify that the SLAPU83A program is enabled and start the CA-ACF2 LDAP Bridge if it is not already running.

2 From TSO, issue the following command:

```
ALTUSER testuserID NAME('ACF22LDAP TEST')
```

where testuserID is any valid CA-ACF2 user ID.

3 Wait briefly, enter OMVS from TSO.

4 Enter the following commands:

```
cd /sdir/sbin

dotestr2l
```

5 At the prompts, enter your CA-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/system/acf22ldap.log to determine the cause of the error.

Testing the LDAP Change Log

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

```
cd /sdir/sbin

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 test, in LDIF format.

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

The LDAP Bridge generates logging information that is written to the sdir/logs/system/slapd.out file, and is printed at the termination of the START job.

Setting the LDAP Bridge Logging Level

The LDAP Bridge generates debugging information that is written to the ***sdir*/conf/system/slapd.out** file, and is printed at the termination of the START job. You can set the logging level using the DEBUG parameter that is found in the 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.

To use multiple debugging levels, add the two 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.

LDAP Server Configuration files

Managing Archived 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 acf22ldap.conf configuration file contains the parameters that control the operation of synchronization daemon. Within acf22ldap.conf, the RETAIN parameter determines how SMF records are to be archived by synchronization daemon.

To set the RETAIN parameter, follow the series of steps below:

1 Open the acf22ldap.conf file located in sdir/conf/system/.

2 Set the RETAIN parameter to the appropriate setting:

- -1 = SMF records are deleted once they are processed and are not written to acf22ldap/old.
- 0 = SMF records are written to acf22ldap/old and are not deleted.
- nn = SMF records are written to acf22ldap/old and records older than nn (0-999) days are deleted by acf22ldap.

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 *SQUAL*.JCLLIB(START). At the bottom, change the SSLPORT variable from 0 to the port used for encrypted communications. 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. Available ports are usually above 1023.

```
// SSLPORT='636',
```

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 *ssdir*/conf/system/certs directory: ca_cert.pem, server_cert.pem and server_key.pem.

In general, to establish an SSL/TLS session, the LDAP Bridge presents the client with its connector certificate. The client then validates that certificate baed 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/system/certs/ca_cert.pem contains this test CA certificate.

First, download *sdir*/conf/system/certs/ca_cert.pem to the client platform, specifying EBCDIC-ASCII translation. After that, the importation method varies depending on the platform. If you are testing from the address book on MS-Windows, for example, you can open MS-Internet Explorer (IE) and select the tools / internet options / content / certificates / import menu options to import ca_cert.pem into your trusted root certificate authorities store. After importation, you will see the “OmniDAP Development” certificate in this store. This will allow you to test SSL/TLS encrypted communications from your MS-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.

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 LDAP Bridge Certificate

To implement 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. You can also generate a connector certificate yourself from ACF2 using the EXPORT command. For more information on digital certificates and the EXPORT command, see the *e Trust CA-ACF2 Security for z/OS Command Functions Guide*.

After you have obtained your certificate, you must store the certificate, its private key, and the CA certificate in the *sdir*/conf/system/certs directory. 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/system/certs/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. Ideally, only the user-id of the connector must have access to this file. This can be implemented by the following commands:

```
cd sdir/conf/system/certs

chown userid ./server_key.pem

chmod 0400 ./server_key.pem
```

Where *userid* is the ACF2 userid for the LDAP Bridge.

SSL/TLS Parameters in Slapd.conf

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

Parameter	**Description**
TLSRandFile	The path the entropy seed used to generate encryption keys. This files generated at start-up by the doslapd script.
TLSCACertificateFile	The path the Certificate Authority Certificate, in base64 format. The delivered value is *sdir*/conf/system/certs/ca_cert.pem. If you wish to use a CA other than the delivered testing CA, 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 the LDAP Bridge Certificate, in base64 format. The delivered value is *sdir*/conf/system/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.
TLSCertificateKeyFile	The path the LDAP Bridge Certificate Private Key, in base64 format. The delivered value is *sdir*/conf/system/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 LDAP Bridge will accept. The delivered value allows the connector to accept high and medium strength ciphers, which is sufficient for most uses.
TLSVerifyClient	Determines 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/system directory, where sdir is the HFS directory you created for the LDAP Bridge, 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. Only the customer settings are documented here.

Slapd.conf Configuration File

In the ***sdir*/conf/system directory**, where ***sdir*** is the HFS directory that you created for the LDAP Bridge, 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 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, which will degrade performance. Our recommendation is 0 (timeout disabled).
Allow bind_v2	This enables back-level support for LDAP version 2 binds. This setting cannot be changed.

Slapd.acf2.conf Backend Configuration File

The slapd.acf2.conf file contains the following online configuration parameters specific to your ACF2 security system.

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

Parameter	Description
Directory	This parameter must be set to%datadir%/bdb/acf2.
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 acf22ldap.conf. Default value is cn=acf2Manager,o=%company%
rootpw	This is the password that goes with the rootdn. Default value is secret
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*/samples/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/system/slapd.acf2.conf** file. 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/system/ contain UNIX environment variables that affect batch and online processing:

- **stdenv.slapd** - Affects online connector processing (START).
- **stdenv.slapadd** - Affects database load processing (SLVCONVT)
- **stdenv.acf22ldap** - Affects online connector processing (STARTT2L)
- **stdenv** - Affects processing for all other processing (STOP, etc.)

As delivered, these files are optimized for the various components 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).”
_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, which 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.

Parameter	Description
_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.codepage*	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 default to code page 1047. This parameter must be added to both the stdenv.slapd and stdenv.acf22ldap 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 ACF2 database. For example: LDAPBRIDGE_LOACALE=Fr_FR.IBM-297

Slapd.acl.conf LDAP Security

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 within the LDAP Bridge, and provides example scenarios to help create ACLs that meet your site's requirements.

ACLs are defined within the *sdir*/conf/system/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 not discussed within this section, please refer to the general ACL specification, which 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 defined within 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 wish to view database entries. If an anonymous user attempts to access a database entry, they will be required to authenticate, while authenticated users are granted read access to the database.

Example 2

The LDAP Bridge uses the following default ACL definition:

```
access to dn.onelevel="ou=people,o=company" attrs=userPassword

by self write
```

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>	attrs=userPassword	userPassword represents the user passwords entry attribute.
<user/group>	self	Self represents the user's own user ID.
<permitted action>	write	Write allows users to overwrite the database entry.

The purpose of this ACL definition is to allow authenticated users to change their own password. This ACL definition is very restrictive. First, the user is only permitted to access user entries within the database. Second, of the user entries available, the user can only access the userPassword attribute. Finally, the user is only permitted to overwrite the user password entry for their own user profile.

Allowing All Users and Groups Read Access to Entire Database

To allow all users, authenticated or otherwise, to view all entries within 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	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

In some cases, you may wish to permit only certain users read access to the entire database. The purpose of these ACL definitions are to protect sensitive information within the database by limiting who can view all the entries. These 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 and USERID2 represents the user IDs of the authorized users. Company represents the root dn you specified for the LDAP Bridge.
<permitted action>	read	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 you specified for the LDAP Bridge.
<permitted action>	read	Read allows users to read the specified database entries.

Limiting Entire Database Access to Specific Groups

In some cases, you may wish to permit only certain groups read access to the entire database. The purpose of these ACL definitions are to protect sensitive information within the database by limiting who can view all the entries. These 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/acf2Group/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/acf2Group/ member.exact= ”cn=*GROUP1*,ou= groups, o=*company*"	group/acf2Group/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 you specified for the LDAP Bridge.
<permitted action>	read	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/acf2Group/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/acf2Group/ member.regex= "cn=*.*,ou=groups ,o=*company*"	group/acf2Group/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	Read allows users to read the specified database entries.

Limiting Entire Database Access to a Specific IP Address

In some cases, you may wish 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=*IPAD DRESS*	peername.ip represents an exact IP address making an LDAP request.*IPADDRESS* represents the IP address of the authorized request.
<permitted action>	read	Read allows users to read the specified database entries.

Limiting Database Access to Specific Entries or Attributes

In some cases, you may wish to restrict what users and groups can view within the database. The purpose of these ACL definitions are to protect sensitive information within the database by limiting 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	Users represents authenticated users.
<permitted action>	read	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	Users represents authenticated users.
<permitted action>	read	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	Users represents authenticated users.
<permitted action>	read	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/system contain database settings that affect batch and online processing:

- **DB_CONFIG.slapd** - Affects online connector processing (START).
- **DB_CONFIG.slapadd** - Affects database load processing (SLVCONVT)

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/bdb/secs/ldif2entry.bdb + 20%. *number_of_caches* must be set to 1. 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.

Setting DB_TXN_NOSYN and DB_TXN_NOT_DURABLE to suit your environment

By default DB_TXN_NOSYNC is set so that it does not immediately write database updates to disk. This improves performance but can result in lost data if the server goes down, through any process other than a normal shutdown, before the database has been updated with the recent changes. You can increase the frequency of database updates by changing the setting od the DB_TXN_NOSYNC parameter.

To have updates written to the database immediately:

1 Open the following file in a text editor:

```
./conf/system/slapd.acf2.conf
```

2 Set the checkpoint parameter as follows for the first database definition:

```
checkpoint 1 1
```

This forces a checkpoint to occur every 1 KB or every one minute. 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.

3 Open the following file in a text editor:

```
./conf/system/doslapd
```

4 In `doslapd` (the startup script of the LDAP server), place the following lines at the beginning of the script:

```
LIBPATH=$LIBPATH:sdir/sbin
sdir/sbin/db_recover -h sdir/bdb/acf2
```

Where sdir is the install directory of the LDAP Bridge. This causes the recovery process to run before the LDAP server starts.

5 Open the following file in a text editor:

```
./conf/system/DB_CONFIG.slapd
```

6 In `DB_CONFIG.slapd` comment out the following flags:

```
#set_flags DB_TXN_NOSYNC
#set_flags DB_TXN_NOT_DURABLE
```

7 Open the following file in a text editor:

```
./data/system/bdb/acf2/DB_CONFIG
```

8 In DB_CONFIG comment out the following flags:

```
#set_flags DB_TXN_NOSYNC
#set_flags DB_TXN_NOT_DURABLE
```

The `DB_TXN_NOSYNC` flag tells the server to synchronize updates to the log according to the checkpoint parameters above. The `DB_TXN_NOT_DURABLE` flag tells it to maintain recovery logs of all update transactions.

9 Stop the server.

10 Run the `SLCONVR` job from `SQUAL.JCLLIB` where `SQUAL` is the high level qualifier that you selected for the MVS data sets during the install.

11 Start the server.

Note: When this change is implemented the log files, (./*.err,*.out,*.log) grow at a much greater rate that the do with the default setting, therefore, it is recommended that you periodically run `SLCONVR` to clear out the log files.

Tuning The Synchronization Daemon

Almost all customization of the synchronization daemon occurs in the acf22ldap.conf configuration file. The sections below describe the various parameters in this file and present step-by-step instructions for performing various common customization tasks.

Synchronization daemon configuration settings are stored in ***sdir*/conf/systems/acf22ldap.conf**, where *sdir* is the install directory of the LDAP Bridge. As delivered, this file enables the synchronization daemon to synchronize ACF2 with the LDAP Bridge.

Synchronization Daemon General Definitions

The following parameters control the global functioning of the synchronization daemon, including which connectors to synchronize, how to handle error conditions, etc.

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	100	Specifies the number of retry attempts for a non-responsive LDAP Server
LOGLEVEL	4	Log level for event details in LOGDIR/acf22ldap.log Range from 0 to 5, 0=minimal information logged, 5=maximum information logged Recommended 4 for proof of concept and 0 for normal operations
CONVERTLOGLEVEL	0	Logging for the database build process
RETAIN	30	Specifies how records are to be written to acf22ldap/old. Values are: -1 = SMF records are deleted once they are processed and are not written to acf22ldap/old. 0 = SMF records are written to acf22ldap/old and are not deleted. *nn* = SMF records are written to acf22ldap/old and records older than *nn* days are deleted by acf22ldap.
NOTIFY	acf2manager@%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.

Parameter	**Default Value**	**Description**
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
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
lLDAPVERSION	3	Specifies the supported LDAP version. Do not change.
ORGDN	*o=%company%*	Configured at run time from site.variables LDAP Root.
MANAGERDN	cn=acf2Manager,	Specifies the LDAP Distinguished Name used to perform LDAP updates.
MANAGERPW	secret	Specifies the LDAP Distinguished Name used by synchronization daemon to perform LDAP Server 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.
SSLKEYFILE	/usr/lpp/hpv33a/	Specifies the path to the SSL keyfile.
SSLKEYPW	xyz.key	Specifies the password for the SSL key.
SQUAL	High-level qualifier.	Specifies the high-level qualifier(s) for your z/OS data sets for this product.

Acf22ldap.conf Error Definitions

This section of acf22ldap.conf describes how the acf22ldap daemon should handle various LDAP error conditions returned from the CA-ACF2 LDAP Bridge. When an LDAP add, modify ordelete request from acf22ldap fails on the target connector, the connector returns an LDAPerror code. You should not have to modify this section from the delivered options. 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 descriptivepurposes only.

• code- The standard LDAP error code returned from the connector.

• level - The acf22ldap severity level for this error code: WARNING, SERIOUS, SEVERE or FATAL. See NOTIFYLEVEL, above.

• action- The action acf22ldap should take in the event of this error.

— NONE - Take no action.

— ABEND - Terminate the acf22ldap task.

— SLEEP - Retry in 10 seconds.

— SEND - Email those identified in the NOTIFY statement.

— MOVE - Move the CA-ACF2 change to the error directory.ynchronization daemon names the entries it adds, modifies, or deletes. If you are using the synchronization daemon to synchronize a remote directory, you must add target statements defining the format of the distinguished names on that remote directory.

Sample ERROR Definitions

ERROR LDAP_SUCCESS 0 WARNING NONE

This rule tells acf22ldap to take no action on successful LDAP requests.

ERROR LDAP_OPERATIONS_ERROR 1 FATAL ABEND

This rule tells acf22ldap terminate in the event of an LDAP operations error (error code 1).

ERROR LDAP_SERVER_DOWN 81 WARNING SLEEP

This rule tells acf22ldap to wait and then try again in the event that the CA-ACF2 LDAP Bridge is down (error code 81)..

Acf22ldap.conf Rule Definitions

Rules come in two types: DATA and UPDATE. DATA rules manipulate the value provided byCA-ACF2 into a different format.

UPDATE rules control how acf22ldap processes add, modify, or delete operations with the CA-ACF2 LDAP Bridge. You may code your own DATA and UPDATE rules to implement custom processing for anygiven LDAP attribute. If you create your own rule, you should define it with a RULE definition in this section of the configuration file.

RULE name type entry library

All parameters must be separated by one or more spaces.

• RULE - Static text identifying this as a RULE statement.

• name - The name of this rule, for use in subsequent KEYWORD statements.

• type - The type of rule:

— DATA - For reformatting attribute values.

— UPDATE - For updating the CA-ACF2 LDAP Bridge.

• entry - The entry point for this rule in the shared library, below.

• library - The name of the shared library (DLL) file containing this rule. The product delivers its default rules in sdir/sbin/default.dll. Any new shared libraries should reside in sdir/sbin.

Sample RULE Definitions

RULE VAL DATA VAL default.dll

This data rule, named VAL, is found at entry point VAL in sdir/sbin/default.dll.

RULE SetValue UPDATE SetValue default.dll

This update rule, named SetValue, is found at entry point SetValue in sdir/sbin/default.dll.

Delivered Rules in Default.dll

In default.dll, acf22ldap delivers the following rules:

Rule Type Description

VAL DATA Use the value from CA-ACF2 as-is.

VALS DATA Use multiple CA-ACF2 values as-is.

NOVAL DATA Do not populate a value.

BOOLTRUE DATA Set the value to TRUE.

BOOLFALSE DATA Set the value to FALSE.

GROUPDN DATA Create a group DN from the

value: cn=value,ou=groups

USERDN DATA Create a user DN from the

value: uid=value,ou=people

BOOL DATA If value is YES or ONE or

TRUE, set to TRUE.

Otherwise, set to FALSE.

RANGE DATA Transform numeric ranges into discreet numbers. Ranges are determined by the –character. For example,

change 1-3 into 1 2 3.

SetSuperGroup UPDATE Update the superior group of the target with the target dn in the acf2SubGroup attribute.

SetValue UPDATE Replace the attribute value. If the attribute does not already exist for the entry, add it. SetMultiValue UPDATE Replace the multivalued attribute values. If the attribute does not already exist for the entry, add it.

AddMultiValue UPDATE Add the values to those already existing for themultivalued attribute. If the attribute does not already exist for the entry, add it.

DelMultiValue UPDATE Delete the values from those already existing for the multivalued attribute.

RemoveAttr UPDATE Delete the attribute value. RemoveAttrs UPDATE Delete all attributes matching the wildcard specification in attribute on the KEYWORD statement.

CreateEntry UPDATE Create a new entry.

SetBoolValue UPDATE If the value is TRUE, set the value to the last 4 characters of the attribute name.

Copy UPDATE Create a resource dn based onthe attribute value, and copy the entry referenced by that dn to the target dn.

CopyPermit UPDATE Create a permit dn based on the attribute value, and copy the entry referenced by that dn to the target dn.

RemoveSubEntry UPDATE In addition to removing the attribute value, remove the dn referenced by the attribute value.

RemoveEntry UPDATE Remove the entry referenced by the target.

Reset UPDATE Remove a dataset or resource permission.

Acf22ldap.conf Target Definitions

Targets define how acf22ldap names the entries it adds, modifies, or deletes. If you are using acf22ldap to synchronize a remote directory, you should add target statements defining the format of the distinguished names on that remote directory.

TARGET name dn parent objectclass [objectclass ...]

All parameters must be separated by one or more spaces:

• TARGET - Static text identifying this as a TARGET statement.

• name - The name of this target, for use in subsequent configuration file directives.

• dn - The prototype distinguished name for this target. This consists of a model distinguished name, minus the suffix, with substitution variables that acf22ldap uses toconstruct specific dns. Substitution variables are prefixed by &, indicating a mandatory substitution, or !, indicating optional substitution. Acf22ldap will ignore clauses in the dn when an optional substitution variable is missing.

• parent- The name of the parent target, if any. If no parent target exists, should be set to static text: “NO_PARENT”. This means that the parent target is a fixed member of the directory tree (such as ou=people), and thus not defined in this configuration file.

• objectclass - One or more objectclasses that acf22ldap uses when constructing newentries for this target.

Sample TARGET Definitions

TARGET Group cn=&GROUP,ou=groups NO_PARENT acf2Group top groupOfNames

This target definition, named GROUP, defines the prototype dn for group entries. This prototype dn requires the GROUP keyword. It also specifies that these entries have a fixed parent not defined in this file. Finally, it directs acf22ldap to create new groups that use the acf2Group, top and groupOfNames objectclasses.

Tuning the MVS data sets

JCLLIB members

The *SQUAL*.JCLLIB MVS file, where *SQUAL* represents your high-level qualifier, contains several members you can customize, depending on your sites requirements. The following table describes the members available for customization:

Members	Statements	Description
CMPLKPGM	LEPREF COBPREF MEMBER	This member compiles various COBOL user-exits, as described below. If you use these exits, you will have to set the substitution variables at left, as described in the JCL.
JOBCARD	JOB	This is normally customized to your site's specifications during the normal installation process.
KEY	KEYVAL	Contains the product key.
LDIFCONV	o: *company*	Static LDIF statements defining the first two levels of the directory tree. Normally, you should not modify this file. However, if the *company* value you chose during the installation has two clauses (for example, o=*company*,c=us), then you must remove the second clause from attribute value for o in the first entry of this file, so that it reads: dn: o=*company* objectClass: top objectClass: organization o: *company*description: *company* z/OS repository
ACF2CONV	DEBUGL	The debugging level used for messages. The only valid values are 000 (no debugging) and 256 (product debugging messages).
	FILTER	Controls whether to call the filter user-exit (SLVCONVTF) as described below. Valid values are YES and NO.
	SUFFIX	The root DN in the directory. You should not have to change this parameter.

User Exits

The *SQUAL*.SRCLIB MVS file, where *SQUAL* represents your high-level qualifier, contains several sample user-exit source programs. The initial comments contained in all user-exit programs present programming information. To compile a user exit, use CMPLKPGM in the JCLLIB as described above. The following table summarizes the delivered sample programs:

Members	**Language**	**Description**
SLVCONVAF	COBOL	Filter user-exit called by SLVCONVT, the ACF2 conversion process. Filters the ACF2 profiles loaded into the LDAP directory. By default, SLVCONVT 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 user-exit. This user exit is controlled by the FILTER flag in *SQUAL*.JCLLIB(TSSCONV), which must be set to YES for it to be enabled.
SLVCONVAU	COBOL	Rule user-exit called by SLVCONVT, the ACF2 conversion process. Contains additional data manipulation rules 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 user-exit. You will also have to modify the ATTR file to specify the new rules for the attributes to which it applies.

MVS Data Set Security

You must protect the following files so access is available only to key personnel and the protected user ID defined for the START, STOP, and SLCONVA jobs:

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

where ***SQUAL*** represents the high-level qualifier you used for your LDAP Bridge.

The DEBUGL Parameter in ACF2CONV

The DEBUGL parameter within the TSSCONV job controls the amount of output generated during the database load and refresh jobs. To optimize performance, this parameter is normally set to “000”, but can be set to “256” to produce full trace debugging output.

5 Appendix: Internationalization

By default, the LDAP Bridge uses the IBM-1047 code page. In order for the LDAP Bridge to store and handle characters that are from code pages other than IBM-1047 the following edits must be made.

- **Edit the *SQUAL*.JCLLIB.SLCONVA job**. Where *SQUAL* is the high level qualifier that was designated during the install process. This change enables the LDAP Bridge to support characters from code pages other than IBM-1047 during the initial database load.
- **Edit stdenv.slapd**. This change enables the LDAP Bridge to support characters from code pages other than IBM-1047 during transactions carried out by the LDAP Bridge.
- **Edit stdenv.acf22ldap**. This change enables the LDAP Bridge to support characters from code pages other than IBM-1047 during transactions carried out by the LDAP Bridge.

Note: The LDAP Bridge supports the use of characters that are supported by the ACF2 database. Characters that are not supported by ACF2, cannot be used.

Editing the *SQUAL*.JCLLIB.SLCONVA

1. Open the *SQUAL*.JCLLIB.SLCONVA job for editing.
2. FIND the CONV step. Edit the EXEC statement to include a keyword parameter PARM with a value in the following format:
   //CONV EXEC PGM=SLCONVA,REGION=0M,COND=(0,LT),TIME=NOLIMIT,//
   PARM='/POSIX(ON),ENVAR(LDAPBRIDGE_LOCALE=*locale.codepage*)'
   Where:

   — *locale* is the locale that you want to use

   — *codepage* is the code page that you want to use

   For example, to work in the Fr_FR locale, using the IBM-297 code page:
   //CONV EXEC PGM=SLCONVA,REGION=0M,COND=(0,LT),TIME=NOLIMIT,// /
   / PARM='/POSIX(ON),ENVAR(LDAPBRIDGE_LOCALE= Fr_FR.IBM-297)'

Editing stdenv.slapd

1. Open stdenv.slapd for editing. By default, stdenv.slapd is located in: *sdir*/conf/system, where *sdir* is the HFS directory that was created for the LDAP Bridge during installation.
2. In stdenv.slapd, add the following parameter:
   LDAPBRIDGE_LOACALE=*locale.codepage*
   Where:

   - *locale* is the locale that you want to use
   - *codepage* is the code page that you want to use

   For example, to work in the Fr_FR locale, using the IBM-297 code page:
   LDAPBRIDGE_LOCALE= Fr_FR.IBM-297'

Editing stdenv.acf22ldap

1. Open stdenv.acf22ldap for editing. By default, stdenv.acf22ldap is located in: *sdir*/conf/system, where *sdir* is the HFS directory that was created for the LDAP Bridge during installation.

2. In stdenv.acf22ldap, add the following parameter:
   LDAPBRIDGE_LOACALE=*locale*.*codepage*
   Where:

   – *locale* is the locale that you want to use

   – *codepage* is the code page that you want to use

   For example, to work in the Fr_FR locale, using the IBM-297 code page:
   LDAPBRIDGE_LOCALE= Fr_FR.IBM-297'

6 Appendix: Troubleshooting

This appendix contains troubleshooting information.

Recovering Data After Restarting Synchronization Daemon

Once a ACF2 change has been processed, synchronization daemon moves the SMF record from the ***sdir/data/system*/acf22ldap/new** directory to the ***sdir/data/system*/acf22ldap/old** or ***sdir/data/system*/acf22ldap/error** directories, where:

- **/old** acts as an archive of ACF2 audit records that can be used for debugging purposes, or to rebuild the ACF2 database.
- **/error** acts as an holding area for ACF2 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.

If the LDAP Bridge is stopped, ACF2 changes accumulate in the directory so that none are lost when it is restarted. If the TSSINSTX user exit is disabled, ACF2 changes cannot be captured or propagated, and are therefore lost. The LDAP Bridge cache must be rebuilt using the SLVCONVT job.

acf22ldap.conf Error Definitions

This section of acf22ldap.conf describes how the synchronization daemon should handle various LDAP error conditions returned from the LDAP Bridge. When an LDAP add, modify or delete request from synchronization daemon fails on the target connector, the LDAP Bridge returns an LDAP error code. You should not have to modify this section from the delivered options.

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 ACF2 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/system/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/system/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 START JCL, as described in "*Ensuring Sufficient Region Size.*"