HP OpenView Select Identity
CA Top Secret LDAP Bridge

For the z/OS® Operating System

CA Top Secret LDAP Bridge Version: 3.3

Installation and Configuration Guide

Document Release Date: June 2006
Software Release Date: June 2006

invent

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

This product includes software developed by Teodor Danciu http://jasperreports.sourceforge.net). Portions Copyright (C) 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 (c) 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

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

Contents

1 Introduction

The CA Top Secret LDAP Bridge is an LDAP gateway that provides access to the CA-Top Secret database. By enabling you to access mainframe security data with LDAP, the CA Top Secret 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 should be experienced in and have access to the following:

- Basic LDAP concepts such as directory schema and LDAP operations.
- Mainframe concepts such as JCL, partitioned datasets, and job submission.
- Have authority to edit mainframe files, create data sets, and submit jobs.
- 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.
- CA-Top Secret concepts such as password verification and resource authorization.
- Have CA-Top Secret authority to create data sets and HFS files.

About the CA Top Secret LDAP Bridge

Mainframe Directories

Large organizations typically employ the mainframe as a central repository for corporate information. Most critical information within this type of environment resides in secure directories such as CA-Top Secret.

As corporations move to improve means of information exchange, there is a need to extend the mainframe directory data to other applications that provide enhanced access to this information.

To integrate this mainframe data into your application infrastructure, you need the CA Top Secret LDAP Bridge.

CA Top Secret LDAP Bridge

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

- Mirror database
- tss2ldap

- ldap2tss
- Configuration database

Mirror database

The mirror database represents a real-time image of the entire CA-Top Secret database as it resides on the host z/OS system. The CA-Top Secret database and mirror database are automatically updated with the tss2ldap and ldap2tss synchronization processes.

tss2ldap

tss2ldap updates the mirror database to reflect the current status of the CA-Top Secret database.

Whenever a change is made to the CA-Top Secret database, tss2ldap intercepts the Audit record generated by the CA-Top Secret command. The CA-Top Secret command is then translated into an equivalent LDAP modify command that updates the mirror database accordingly.

ldap2tss

ldap2tss modifies CA-Top Secret to reflect changes initiated within the CA Top Secret LDAP Bridge.

Whenever users make a change to the CA-Top Secret database, ldap2tss translates the LDAP modify command into an equivalent CA-Top Secret command to update the CA-Top Secret database accordingly. Once the change has been made to the CA-Top Secret database, tss2ldap processes and reflects the change within the mirror database.

Configuration database

The configuration database acts as a central repository of configuration data for all components of the CA Top Secret LDAP Bridge.

2 Installing and Configuring the CA Top Secret LDAP Bridge

The CA Top Secret LDAP Bridge enables HP OpenView Select Identity to perform tasks on CA-Top Secret databases.

The following tasks are supported by the CA Top Secret LDAP Bridge:

- Add, update, and remove users
- Retrieve user attributes
- Enable and disable users
- Verify a user's existence
- Change user passwords
- Reset user passwords
- Retrieve all entitlements
- Retrieve a list of supported user attributes
- Assign and unassign entitlements to and from users, including the addition of users to multiple OUs

System Requirements

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

z/OS Requirements

The CA Top Secret LDAP Bridge requires the following elements:

- z/OS V1R2M0 or later
- CA-Top Secret releases included with above versions of z/OS
- UNIX System Services
- TCP/IP for z/OS
- LE runtime libraries, with C-language support

TCP/IP Requirements

The CA Top Secret LDAP Bridge can communicate using any port. The default ports for LDAP communications are 389 for unencrypted data, and 636 for SSL-encrypted data.

Before You Begin

You must prepare these elements of your environment before installing the CA Top Secret LDAP Bridge.

Configuring UNIX System Services

The CA Top Secret LDAP Bridge runs on the mainframe under UNIX System Services (USS):

- Make sure that you can access USS using either ISHELL, OMVS, or telnet, and are authorized to browse directories and issue UNIX commands.
- Allocate an HFS directory for the CA Top Secret LDAP Bridge. The amount of disk space required for the directory can be determined using the following formula:

  Disk Space = 200MB + (size of CA-Top Secret database x 3.2)
- For the CA Top Secret LDAP Bridge to function correctly, the parent directories of the CA Top Secret LDAP Bridge must have execute access permission for OTHER. If, for example, the parent directory for the product is /usr/lpp, you should make sure that the both /usr and /usr/lpp have execute permission for OTHER. To view the permissions of this directory, you would 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
- For the CA Top Secret LDAP Bridge to function correctly, the directory for the CA Top Secret LDAP Bridge itself must have appropriate permissions:
  - OWNER: read/write/execute
  - GROUP: read/write/execute
  - OTHER: execute

  If, for example, you are installing hpv33t into the /usr/lpp/hpv33t directory, assign the appropriate permissions by issuing the following commands:

  chmod 0771 /usr/lpp/hpv33t
- The group owner of the hpv33t directory must be a CA-Top Secret group of which the user-ID associated with the CA Top Secret LDAP Bridge started task is a member. If, for example, the hpv33t directory is /usr/lpp/hpv33t, and you plan to run the CA Top Secret LDAP Bridge under a user-ID that is a member of the ADMIN group, then the group owner of the /usr/lpp/hpv33t should be ADMIN. To see the group owner of /usr/lpp/hpv33t, issue the following command:

  ls -ld /usr/lpp/hpv33t

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

  chgrp ADMIN /usr/lpp/hpv33t

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

Configuring Your z/OS TCP/IP and Firewall

The CA Top Secret LDAP Bridge communicates using TCP/IP. You must enable the following ports for TCP/IP access:

- Port 389

- Port 636 (if you plan to use SSL encryption)
- Port 623, or the appropriate port used at your site for OMVS telnet access

If users from outside your firewall will be accessing the CA Top Secret LDAP Bridge, you must modify your firewall to enable access ports 389 or 636.

If any ports other than the defaults are used, ensure that the PORT parameter is set accordingly in the START JCL.

Assuring Sufficient Region Size

CA Top Secret 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 CA Top Secret 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.

You should verify with the system programmer that the job class and user-id under which you plan to run the CA Top Secret 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 CA-Top Secret Access to Control and Authorize FACILITY Class Resources

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

- BPX.FILEATTR.PROGCTL in the IBMFAC Resource Class
- BPX.FILEATTR.APF in the IBMFAC Resource Class

Add TSSCFILE to the TSO Authorized Command List

The TSSCFILE must be added to the TSO Authorized Command List that is stored in the SYS1.PARMLIB(IKJTSOxx) member.

To define TSSCFILE as a TSO authorized command, edit the IKJTSOxx member, and add the following definition in the appropriate location:

```
AUTHCMD NAMES(        /* AUTHORIZED COMMANDS     */     + TSSCFILE                /*
REQUIRED HP TSS CONNECTOR  */ +
```

Verify CA-Top Secret access to BPX.DAEMON

This section applies only if you have defined BPX.DAEMON to CA-Top Secret in the IBMFAC Resource Class.

The user ID under which the CA Top Secret LDAP Bridge LDAP executable is run must have CA-Top Secret READ access to BPX.DAEMON in the IBMFAC Resource Class. Verify that the appropriate access has been granted. For further information concerning this procedure, refer to Chapter 25 of the *IBM UNIX System Services Planning Guide*.

Installing from CD or Via FTP

The CD or downloaded version of the CA Top Secret LDAP Bridge release media contains the compressed file hpv33t.pax.Z, which is used to install the CA Top Secret LDAP Bridge onto an HFS file system.

Expanding the CA Top Secret LDAP Bridge PAX Archive

Insert the CD in a machine that has FTP transfer abilities to your z/OS system, then transfer the hpv33t.pax.Z file using FTP to your HFS directory. During the transfer, be sure to specify binary mode.

To expand the PAX file, enter OMVS from TSO, and issue the following commands:

cd ***sdir***

pax -rv -px -f hpv33t.pax.Z

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

Customizing the CA Top Secret LDAP Bridge

After the CA Top Secret LDAP Bridge has been installed, you must complete the following post-installation steps.

The recommended installation directory for the CA Top Secret LDAP Bridge is the default location of */usr/lpp/hpv33t*. You can, however, install the CA Top Secret LDAP Bridge in any location you choose.

Running the Install Script

Execute the installation script to install the CA Top Secret LDAP Bridge. The script performs the following tasks:

- Assigns the site-specific variables defined within the site.variables file, or prompts the user for their values.
- Customizes the JCL and configuration files with the dosed subscript.
- Allocates the SRCLIB, LOADLIB, JCLLIB, and ATTR files under z/OS with the doalloc subscript.
- Moves the source, load, JCL, and attributes from UNIX System Services to z/OS with the doget subscript.
- Frees the file allocations for SRCLIB, LOADLIB, JCLLIB, and ATTR with the dofree subscript.
- Installs the CA Top Secret LDAP Bridge along with the tss2ldap and ldap2tss components.

Executing the Install Script

The first time that the install 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 install script can be run as many times as necessary. Whenever the install script is run again, the script deletes the previous files and creates new ones based on the initial information provided.

The install script is located in *sdir*, where *sdir* is the HFS directory you created for the CA Top Secret LDAP Bridge. If you are using the default installation directory, the install script is located in */usr/lpp/hpv33t*. To run the installation script, enter OMVS from TSO, then issue the following commands:

cd sdir

sh install

When the install script is executed, you will need to supply the following site-specific information:

SQUAL: Dataset Names

Default:

SQUAL='HPV33T'

The SQUAL variable represents the high-level qualifier for the CA Top Secret LDAP Bridge datasets. Enter a value that conforms to your site standards. It is recommended that you preserve the second-level qualifier as HPV33T.

PDUNIT: Permanent Storage Device Name

Default:

PDUNIT='***3390***'

The PDUNIT variable represents the unit designation for your site's permanent storage devices. Other common values for this variable are "DISK" and "DASD".

TDUNIT: Temporary Storage Device Name

Default:

TDUNIT='***SYSDA***'

The TDUNIT variable represents the unit designation for your site's temporary storage device. Another common value for this variable is "VIO".

SDIR: HFS Root Directory

There is no default value for SDIR. The SDIR variable represents path to the directory from which the install script is being run. The default directory is usr/lpp/hpv33t. You can, however, install the CA Top Secret LDAP Bridge into another location.

COMPANY: LDAP Root RDN Value

Default:

company='***your domain name***'

The company variable represents the LDAP root. You must define this value later in order to access your directory.

Hostname: DNS Host Name of IP Address

Default:

hostname='***hostname***'

The hostname variable represents the name of host system for the CA Top Secret LDAP Bridge. Change the hostname operand to match the Domain Name or TCP/IP address of the host for the CA Top Secret LDAP Bridge. The default value that initially appears is the hostname defined for the z/OS system where the install script is running.

Hostport DNS host port

Default:

hostport='***389***'

The hostport variable represents the port number used for unencrypted communications by the host system. Change the 389 operand to match the TCP/IP port used by the CA Top Secret LDAP Bridge for unencrypted communications. The default port is 389. If you change this default port, you must use an unreserved port that is available on the host running the CA Top Secret LDAP Bridge. Available ports are usually above 1023.

Install Script Error Messages

The first time the installation script is run, four warnings will be produced that are related to the DELETE functions issued for the SRCLIB, JCLLIB, LOADLIB, and ATTR files. These warnings can be safely ignored.

At the end of the script, you might also see a message similar to:

IGD103I SMS ALLOCATED TO DDNAME SYS00024
THE RECORD SIZE IN THE OUTPUT DATA SET IS SMALLER THAN A LINE IN THE INPUT FILE
SOME RECORDS HAVE BEEN TRUNCATED.

This is also a normal message, which you can safely ignore.

Manual Editing of the site.variables File

The site.variables file sets filenames and basic configuration options for your site as described in the preceding section. The file is located in *sdir*/samples/site.variables, where *sdir* is the HFS directory you created for the CA Top Secret LDAP Bridge. If you wish to change any of the configuration options specified during the initial execution of the install script, simply edit this file and re-run the install script. The install script can be re-run as many times as necessary.

Edit the Prototype Job Card

Locate the JOBCARD member of the ***SQUAL***.JCLLIB data set, where ***SQUAL*** is the high-level qualifier you selected for your data sets. The JOBCARD member contains a prototype job card that can be customized and copied to any JCL that is used to test and operate the CA Top Secret LDAP Bridge.

Configure and Start the CA Top Secret LDAP Bridge

Configure and start the CA Top Secret LDAP Bridge by performing the tasks detailed in the sections below.

Verify Sufficient z/OS Resource Allocations

The CA Top Secret LDAP Bridge is a high-volume online process that requires resources similar to those required for a high-volume CICS region. Verify that sufficient CPU, disk, and memory resources have been allocated to the CA Top Secret LDAP Bridge environment.

Verify Module File Attributes

Several of the CA Top Secret LDAP Bridge LDAP modules must be APF-Authorized and several must be both APF-Authorized and Program-Controlled in order to perform authentications against CA-Top Secret.

To verify that these modules have the necessary authorization, run the following command for each file:

extattr ***sdir***/sbin/***module***

where:

- ***sdir*** is the name of the CA Top Secret LDAP Bridge HFS directory
- ***module*** is the name of an LDAP module.

LDAP modules

The following LDAP modules require that the APF-Authorized parameter is set to "YES":

- tss2ldap
- slconvt
- slconvtp

The following LDAP modules require that both the APF-Authorized parameter and the Program-Controlled parameter are set to "YES":

- slapd
- ldap2tss.so
- pwdbind.so
- authtso
- libshr.so

To authorize a module that does not have the required authorizations, enter OMVS from TSO and issue one of the following commands:

extattr +ap ***sdir***/sbin/***module***

or

extattr +a ***sdir***/sbin/***module*** as the case dictates

where:

- ***sdir*** is the name of the CA Top Secret LDAP Bridge HFS directory
- ***module*** is the name of an LDAP module.

Starting the CA Top Secret LDAP Bridge

You can run the CA Top Secret LDAP Bridge as either a started task, a background or foreground job running under UNIX System Services, or as a submitted job.

Regardless of whether you run the CA Top Secret LDAP Bridge as a started task or a submitted job, the CA-Top Secret user-ID of the CA Top Secret LDAP Bridge must be a member of the CA-Top Secret group that owns the HFS directories for the CA Top Secret LDAP Bridge. To see the group owner, from OMVS, issue the following command:

ls -ld *dir*

where dir is the root directory of the product.

Submitted Jobs

For testing purposes, it is recommended to start the CA Top Secret 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 CA Top Secret LDAP Bridge.

Started Tasks

To create started tasks that start and stop the CA Top Secret LDAP Bridge, customize the appropriate JCL provided within the *SQUAL*.JCLLIB dataset, where:

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

Testing the CA Top Secret LDAP Bridge

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

Running the dotestserver Script

To test the CA Top Secret LDAP Bridge, follow the series of steps below:

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

   cd /sdir/samples

   dotestserver
3. At the prompts, enter your CA-Top Secret user ID and password. This test should return information on your CA-Top Secret user ID as stored in the LDAP repository.

Stopping the CA Top Secret LDAP Bridge

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

Examining Dump Information

The CA Top Secret LDAP Bridge writes dump information to facilitate fault diagnosis if any troubleshooting is required. The dump information is located in the *sdir*/samples directory, and CA Top Secret LDAP Bridge records are identified by the prefix CEE. You can delete older dump information as required in order to optimize your installation environment.

Insufficient memory error condition

If the CA Top Secret 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*/samples/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*/samples/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 Assuring Sufficient Region Size.

3 Installing and Configuring tss2ldap

This chapter describes how to install and configure the tss2ldap portion of the CA Top Secret LDAP Bridge. Tss2ldap synchronizes CA-Top Secret changes with the CA Top Secret LDAP Bridge.

Installing tss2ldap

Tss2ldap runs as a stand-alone UNIX daemon in a separate address space from the CA Top Secret LDAP Bridge. It reads the audit records that are generated whenever CA-Top Secret changes are made, and propagates the changes to the CA Top Secret LDAP Bridge using LDAP. The audit records are written to the ***sdir*/tss2ldap/new** directory by the TSSINST program that runs as a TSS user exit point.

To use tss2ldap, you must activate the TSS user exit TSSINST described below.

Installing the CA-Top Secret Site Installation Exit TSSINSTX

In order to synchronize updates made to the CA-Top Secret repository outside of the HP provisioning application, an exit must be installed into CA-Top Secret. The install script assembles and link-edits this exit into SQUAL.LOADLIB(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)

You can also temporarily activate the exit by issuing the following command from the TSO command line:

TSS MODIFY(EXIT(ON))

After initial testing, to permanently activate the exit, edit SYS1.PARMLIB(TSSPARM0) and add the following statement.

EXIT(ON)

Integration with an Existing Version of TSSINSTX

TSSINSTX processes site specific code for a myriad of Top Secret functions. If you are running a site-specific, customized version of TSSINSTX, then the HP CA Top Secret LDAP Bridge version of TSSINSTX will need to be integrated into your version.

In order to do this, you need to make the following additions to your version of the TSSINSTX source code:

1 In the active exit matrix, the COMMAND Exit must be activated as shown below:

```
MATRIX   DC    CL8'MATRIX'      DO NOT INSERT OR DELETE STATEMENTS HERE
*
#####YES EQU   255              FUNCTION IS ACTIVE
######NO EQU   000              FUNCTION IS NOT ACTIVE
*
         DC    AL1(######NO) 00 PRE-INIT
         DC    AL1(######NO) 04 VOLUME ACCESS VALIDATION
         DC    AL1(######NO) 08 DATASET VALIDATION
         DC    AL1(######NO) 12 RESOURCE VALIDATION
         DC    AL1(#####YES) 16 TSS COMMAND
         DC    AL1(######NO) 20 ADDRESS SPACE TERMINATION
         DC    AL1(######NO) 24 POST-INIT
         DC    AL1(######NO) 28 UNDEFINED ACID
         DC    AL1(######NO) 32 NEW PASSWORD VERIFICATION
         DC    AL1(######NO) 36 TERMINAL I/O
```

2 The command exit handler must include the code as shown below:

```
COMMAND  DS    0H
* USER CODE GOES HERE TO INTERPRET TSS COMMAND USAGE
* TXACTGTS CAN BE USED TO DETERMINE THE MACHINE OF ORIGIN WHEN V4L3M000
* CPF IS IN USE.  TXACTGTS = 0 INDICATES THE COMMAND           V4L3M000
* ORIGINATED ON THE LOCAL MACHINE.  IF TXACTGTS > 0, THEN IT   V4L3M000
* POINTS TO THE 8-CHAR NODE NAME OF THE ORIGINATING MACHINE.   V4L3M000
         L     R15,=V(SLAPU83T)
         LR    R1,R9                 Restore parameter list
         BALR  R14,R15
         B     EXIT0
         EJECT
```

Note that the call to SLAPU83T always returns 0.

Assemble and link-edit TSSINSTX. If you have a site-specific version, then you probably will have existing JCL to do this. The linkage-edit step of this JCL will require customization to include the CA Top Secret LDAP Bridge modules, as shown below:

```
//LKED     EXEC  PGM=HEWL,COND=(4,LT,ASMEVT),
//  PARM='AC=1,LIST,XREF,MAP,NCAL,AMODE=31,RMODE=ANY'
//SYSLIB   DD DISP=SHR,DSN=C56.HPV33T.LOADLIB
//OBJECT   DD DISP=SHR,DSN=C56.HPV33T.OBJLIB
//SYSLMOD  DD DISP=SHR,DSN=C56.HPV33T.LOADLIB(TSSINSTX)
//SYSPRINT DD SYSOUT=*
//SYSUT1   DD UNIT=SYSDA,SPACE=(CYL,(5,3))
//SYSLIN   DD *
   INCLUDE OBJECT(TSSINSTX)
   INCLUDE OBJECT(SLAPU83T)
   INCLUDE OBJECT(SLATEVTT)
   ORDER TSSINSTX
   ENTRY TSSINSTX
   NAME TSSINSTX(R)
/*
```

Please note that all modules that comprise TSSINSTX must be assembled and linked as AMODE(31) RMODE(ANY). The ENTRY TSSINSTX statement is required, or the exit will fail when loaded. The exit needs to be installed in a link-listed library and the link list refreshed.

Populating the LDAP Directory

The CA Top Secret LDAP Bridge uses a directory database that is populated with data from your CA-Top Secret repositories. Once the CA Top Secret LDAP Bridge database is initially loaded, the tss2ldap and ldap2tss components of the CA Top Secret LDAP Bridge keep all databases synchronized.

Directory Load Conversion Jobs

To populate your CA Top Secret LDAP Bridge database, you must run the SLCONVT job found within *SQUAL*.JCLLIB, where *SQUAL* is the high-level qualifier you selected for your data sets. SLVCONVT writes the CA-Top Secret database to a flat file, converts the flat file to LDIF statements, then loads the LDIF statements to the CA Top Secret LDAP Bridge database.

The CA-Top Secret user-ID under which you run the SLVCONVT job must be a member of the CA-Top Secret group that owns of the HFS directories for the CA Top Secret LDAP Bridge. To see the group owner, from OMVS, issue the following command:

ls -ld *dir*

where *dir* is the root directory of the product.

When the flat file is converted to LDIF statements, the statements are written to the ***sdir*/samples/tss.ldif** file, where ***sdir*** is the HFS directory you created for the CA Top Secret LDAP Bridge.

Once you have made the customization changes to the conversion job, submit the JCL. All steps in the conversion jobs return a condition code of 04 or less.

Notes:

- HFS File System Full Condition: The CA Top Secret LDAP Bridge requires approximately 200 MB of space in addition to the size of your CA-Top Secret primary database. Allocating insufficient space can result in a SOC6 Abend or other UNIX error. If this error condition occurs, you must unmount your HFS file system, create a backup, reallocate a new file system with sufficient space, then restore the new file system.
- Temporary File System Full Condition: An error code of 256 indicates that the temporary file system is full, and fails to stop the CA Top Secret LDAP Bridge. If this error condition exists, you must allocate a larger temporary file system. For more information concerning this procedure, refer to Chapter 24 of the *IBM UNIX Systems Services Panning Guide*.
- Insufficient SORTWK Space Allocation Condition: A return code of 16 along with the following error messages:
  - ICE046A 0 SORT CAPACITY EXCEEDED
  - CR07: SORT FAILED, SEE RETURN CODE AND SORT OUTPUT

  indicates that sorting in SLVCONVT has failed due to insufficient SORTWK dataset space allocations.
- An 0C4 abend for module IGZCEV5 in the CONV step indicates that your default Language Environment Libraries are not compatible with the versions used by the CA Top Secret LDAP Bridge. The CA Top Secret LDAP Bridge requires Language Environment version 1.4 or above. This is almost certainly already installed on your system, but you will need to explicitly point the CA Top Secret LDAP Bridge at these libraries. Identify the library name for the LE 1.4 SCEERUN library, and add it to the STEPLIB concatenation of the CONV step for SLVCONVT.

- If this job exits the DOLDIF step with a return code 9 (usually without any messages), this indicates that it could not allocate sufficient memory. As delivered, the CA Top Secret LDAP Bridge requires approximately 200 megabytes of processor memory. You will have to consult your systems programmer, and adjust the user-ID, job class or REGION parameter in the DOLDIF step to assure that this job can allocate sufficient memory.
- If the job exits the DOLDIF step with a non-zero return code and a message of "cannot execute", this indicates an authority issue. The user ID of the SLVCONVT job does not have sufficient authority to run *sdir*/samples/doldif or *sdir*/sbin/slapd. You must change the user or group of the job to one that has execute authority.

Starting tss2ldap

Tss2ldap starts automatically via the same START JCL that is used to start the CA Top Secret LDAP Bridge. Whenever you start the CA Top Secret LDAP Bridge, tss2ldap is also active.

If tss2ldap ever needs to be started separately from the CA Top Secret LDAP Bridge, submit the STARTT2L member of the SQUAL.JCLLIB.

Testing tss2ldap

To test the tss2ldap daemon by running the dotestt2l script, follow the series of steps below:

1. Verify that the TSSINST program is enabled and start the CA Top Secret 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 CA-Top Secret user ID.
3. Wait briefly, enter OMVS from TSO.
4. Enter the following commands:

   cd /sdir/samples

   dotestt2l
5. At the prompts, enter your CA-Top Secret user ID and password along with *testuserID*. This test should return the distinguished name of the entry along with the following text:

   cn: TSS2LDAP TEST

   If you do not receive this result, consult sdir/tss2ldap.log to determine the cause of the error.

Stopping tss2ldap

Tss2ldap stops automatically via the same STOP JCL that is used to stop the CA Top Secret LDAP Bridge. Whenever you stop the CA Top Secret LDAP Bridge, tss2ldap also stops.

If tss2ldap ever needs to be stopped separately from the CA Top Secret LDAP Bridge, submit the STOPT2L member of the SQUAL.JCLLIB.

Recovering Data After Restarting tss2ldap

As long as TSSINST is active, all CA-Top Secret changes are captured to the ***sdir*/tss2ldap/new** directory. If tss2ldap is stopped, CA-Top Secret changes accumulate in the directory so none are lost when it is restarted.

Similarly, if the CA Top Secret LDAP Bridge is stopped, tss2ldap accumulates any CA-Top Secret changes so none are lost when the CA Top Secret LDAP Bridge is restarted.

If the TSSINST user exit is disabled, CA-Top Secret changes cannot be captured or propagated, and are therefore lost. The CA Top Secret LDAP Bridge cache must be rebuilt using the SLVCONVT job.

Archiving CA-Top Secret Changes

Once a CA-Top Secret change has been processed, tss2ldap moves the SMF record from the ***sdir*/tss2ldap/new** directory to the ***sdir*/tss2ldap/old** or ***sdir*/tss2ldap/error** directories, where:

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

Managing Archived CA-Top Secret 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 and schedule the T2LCLEAN job to run regularly.

Setting the RETAIN parameter

The tss2ldap.conf configuration file contains the parameters that control the operation of tss2ldap. Within tss2ldap.conf, the RETAIN parameter determines how SMF records are to be archived by tss2ldap.

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

1. Open the tss2ldap.conf file located in ***sdir*/samples/**.
2. Set the RETAIN parameter to the appropriate setting:
   - -1 = SMF records are deleted once they are processed and are not written to tss2ldap/old. Note that if you choose this setting, you will not have to schedule the T2LCLEAN job as no records are written to the archive.
   - 0 = SMF records are written to tss2ldap/old and are not deleted.
   - *nn* = SMF records are written to tss2ldap/old and records older than *nn* (0-999) days are deleted once the T2LCLEAN job is run.

Running T2LCLEAN

The T2LCLEAN job is used to purge the archive of audit records that exceed the age limit specified by the RETAIN parameter. The T2LCLEAN job is located ***SQUAL***.JCLLIB and must be scheduled to run on a regular basis (either daily or weekly).

Note: You do not need to schedule the T2LCLEAN job if RETAIN=-1, as no audit records are written to the archive.

4 Installing and Configuring Ldap2tss

Ldap2tss propagates changes made via the CA Top Secret LDAP Bridge to the CA-Top Secret database. The CA Top Secret LDAP Bridge also installed ldap2tss as part of its installation procedure. As ldap2tss runs in the same address space as the CA Top Secret LDAP Bridge, you do not need to run ldaptss separately from the CA Top Secret LDAP Bridge. Whenever you start the CA Top Secret LDAP Bridge, ldap2tss is also active.

Testing Ldap2tss

To test Ldap2tss by running the dotestl2t script, follow the series of steps below:

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

   cd /sdir/samples

   dotestl2t
4. At the prompts, enter your CA-Top Secret user ID and password along with a *newuserID* that will be created on your CA-Top Secret database using ldap2tss.

   This command can take up to one minute to complete. It should return the LDAP information for the new CA-Top Secret user ID created by this script. If you receive a CA-Top Secret error message, your own CA-Top Secret user ID may lack the authority required to create new user IDs. The CA-Top Secret message should contain information on why the command failed. If you are unable to correct the problem in CA-Top Secret, please contact technical support.
5. After this command completes, issue the following CA-Top Secret command from TSO:

   LU *newuserID*

   You should see output similar to the following for this new CA-Top Secret user:

   ACCESSORID = *newuserID* NAME=ROCKET USER1
   TYPE = DIV C/A SIZE = 1280 BYTES
   DIV ACID = ROCDIV DIVISION = ROCKET DIVISION
   ZONE ACID = ROCZONE ZONE = ROCKET ZONE
   CREATED = 03/08/05 LAST MOD = 07/14/05 12:39
   PROFILES = BSCPROF ROCPROF
   GROUPS = ROCGROUP FTPGRP
   LAST USED = 07/14/05 12:39 CPU(DDIZ) FAC(TSO) COUNT(00911)
   DLFTGRP = FTPGRP

   TSS0300I LIST FUNCTION SUCCESSFUL

   Where:

— ***newuserID*** is the user-ID you selected in step 4.

If CA-Top Secret produces and error message for this command, please refer to *sdir*/samples/slapd.err for detailed error data on why CA-Top Secret did not create this user, and then contact technical support.

Testing Ldifsync

To test Ldifsync by running the dotestls script, follow the series of steps below:

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

   cd /sdir/samples

   dotestls
4. Respond to the prompts for your user-ID and password.
5. The script should return an attribute, replog, that contains the changes made to the server as part of the previous tests, in LDIF format.

5 Operating and Tuning Tss2ldap

This chapter describes how to operate and tune tss2ldap within your environment.

Customizing Tss2ldap

Almost all customization of tss2ldap occurs in the tss2ldap.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.

tss2ldap configuration settings are stored in ***sdir*/samples/tss2ldap.conf**, where *sdir* is the install directory of the CA Top Secret LDAP Bridge. As delivered, this file enables tss2ldap to synchronize CA-Top Secret with the CA Top Secret LDAP Bridge.

Tss2ldap General Definitions

The following parameters control the global functioning of the tss2ldap daemon, including which connectors to synchronize, how to handle error conditions, etc.

Parameter	Default Value	Description
DEBUG	0	Specifies which type of debug messages will be recorded in the long. • 0=All messages • 1=Information and error messages • 2=Error messages only
DIR	CA Top Secret LDAP Bridge root directory	Specifies the base directory for LDAP processing. tss2ldap looks in *sdir*/tss2ldap/new for new audit files to process.
HOST	CA Top Secret LDAP Bridge host name	Specifies the target connector.
lLDAPVERSION	3	Specifies the supported LDAP version. Do not change.
LOGFILE	.<DIR>/ tss2ldap.log	Specifies the path to the logfile containing all messages.

Parameter	Default Value	Description
MANAGERDN	cn=tssManager,	Specifies the LDAP Distinguished Name used to perform LDAP updates.
MANAGERPW	secret	Specifies the password for managerdn. The password can be changed by you to a more secure value. If you change this value, however, you should also change the corresponding value in *sdir*/samples/slapd.conf.
NOTIFY	Email addresses	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 email to the personnel listed in NOTIFY, above. Values are: WARNING - Informational SERIOUS - Config. error should be fixed SEVERE - Possible data loss FATAL - Error resulting in termination
ORGDN	o=*company*	Specifies the root dn in the connector.
POLL	2	Specifies the propagation speed, in seconds.
PORT	CA Top Secret LDAP Bridge host port	Specifies the target port.
REPLOG	/<DIR>/ replog.ldif	Specifies the location of the replog.
RETAIN	0	Specifies how records are to be written to tss2ldap/old. Values are: -1 = SMF records are deleted once they are processed and are not written to tss2ldap/old. 0 = SMF records are written to tss2ldap/old and are not deleted. *nn* = SMF records are written to tss2ldap/old and records older than *nn* days are deleted once the T2LCLEAN job is run.

Parameter	Default Value	Description
RETRY	3	Specifies the number of retry attempts for a non-responsive connector.
SQUAL	High-level qualifier.	Specifies the high-level qualifier(s) for your z/OS datasets for this product.
SSL	N	Specifies if SSL is to be used for communication to the connector. This is usually not necessary for local communications with the CA Top Secret LDAP Bridge.
SSLKEYFILE	/usr/lpp/hpv33t/	Specifies the path to the SSL keyfile.
SSLKEYPW	xyz.key	Specifies the password for the SSL key.
TSSCOMMAND	TSS LIST(%s) DATA(ALL)	Specifies the TSS command. This must be kept in sync with SQUAL.JCLLIB(TSSCFILE)

Tss2ldap.conf Error Definitions

This section of tss2ldap.conf describes how the tss2ldap daemon should handle various LDAP error conditions returned from the CA Top Secret LDAP Bridge. When an LDAP add, modify or delete request from tss2ldap fails on the target connector, the CA Top Secret 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 tss2ldap severity level for this error code: WARNING, SERIOUS, SEVERE or FATAL. See NOTIFYLEVEL, above.
- **action**- The action tss2ldap should take in the event of this error.
  - NONE - Take no action.
  - ABEND - Terminate the tss2ldap task.
  - SLEEP - Retry in 10 seconds.
  - SEND - Email those identified in the NOTIFY statement.
  - MOVE - Move the CA-Top Secret change to the error directory.

Sample ERROR Definitions

ERROR LDAP_SUCCESS 0 WARNING NONE

This rule tells tss2ldap to take no action on successful LDAP requests.

ERROR LDAP_OPERATIONS_ERROR 1 FATAL ABEND

This rule tells tss2ldap terminate in the event of an LDAP operations error (error code 1).

ERROR LDAP_SERVER_DOWN 81 WARNING SLEEP

This rule tells tss2ldap to wait and then try again in the event that the CA Top Secret LDAP Bridge is down (error code 81).

Tss2ldap.conf Target Definitions

Targets define how tss2ldap names the entries it adds, modifies, or deletes. If you are using tss2ldap 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 tss2ldap uses to construct specific dns. Substitution variables are prefixed by &, indicating a mandatory substitution, or !, indicating optional substitution. Tss2ldap 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 tss2ldap uses when constructing new entries for this target.

Sample TARGET Definitions

TARGET Acid uid=&ACID,ou=people NO_PARENT tssAcid top person, organizationalperson, inetorgperson

This target definition, named ACID, defines the prototype dn for group entries. This prototype dn requires the Acid keyword. It also specifies that these entries have a fixed parent not defined in this file. Finally, it directs tss2ldap to create new acids that use the tssAcid, person, organizationalperson, inetorgperson, and top.

6 Operating and Tuning the CA Top Secret LDAP Bridge

This chapter describes how to operate and tune the CA Top Secret LDAP Bridge within your environment.

Running the CA Top Secret LDAP Bridge

You can run the CA Top Secret LDAP Bridge in the following environments:

- z/OS batch job or started task using BPXBATCH
- Under OMVS in either the foreground or background

Running as a Batch Job or Started Task

This is the preferred method, as it allows you easily to manage the CA Top Secret LDAP Bridge from MVS, using SDSF and the operator console.

Submitted Jobs

To run the CA Top Secret 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 CA Top Secret LDAP Bridge, submit the STOP member of the *SQUAL*.JCLLIB dataset.

Started Tasks

To create started tasks that start and stop the CA Top Secret LDAP Bridge, customize the appropriate JCL provided within the *SQUAL*.JCLLIB dataset, where:

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

Running in the Foreground (OMVS or Telnet)

To run the CA Top Secret LDAP Bridge in the foreground, issue the following command from your Unix shell (OMVS or telnet):

cd /*sdir*/samples

doslapd *port sslport debug*

Where *debug* is the optional debugging level that ranges from -1 to 65535 as described below, *port* is the TCP/IP port for clear-text communication and *sslport* is the TCP/IP port for encrypted communications. *port* and *sslport* are required parameters; specifying a *port* of 0 disables clear-text communications, while specifying an *sslport* of 0 disables encrypted communications. It is recommended that, for all testing, you specify an *sslport* of 0.

To stop the CA Top Secret LDAP Bridge, issue the following command:

cd /*sdir*/samples

dostop

Terminating the CA Top Secret LDAP Bridge by other means, such as using CTRL-C, or a terminal idle timeout condition, can corrupt the database. In this case, you will have to rebuild it using SLVCONVT.

Running in the Background (OMVS or Telnet)

To run the CA Top Secret LDAP Bridge in the background, issue the same command as for running in the foreground (described above), with ampersand (&) appended:

cd /sdir/samples

nohup doslapd port sslport debug &

To stop the CA Top Secret LDAP Bridge, issue the following command:

cd /*sdir*/samples

dostop

Terminating the CA Top Secret LDAP Bridge by other means, such as using CTRL-C, or a terminal idle timeout condition, can corrupt the database. In this case, you will have to rebuild it using SLVCONVT.

Setting the Debugging Level

The CA Top Secret LDAP Bridge generates debugging information that is written to the ***sdir*/slapd.out** file, and is printed at the termination of the START job.

You can set the debugging level using the **-d** parameter as described in the sections above, or set with the DEBUG parameter found in the START JCL. The debugging level cannot be changed once the CA Top Secret LDAP Bridge is started. To change the debugging level, stop the CA Top Secret LDAP Bridge, make the required changes, then restart the CA Top Secret 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.

Encryption (SSL/TLS)

The CA Top Secret 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, which you should 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, you should select an unreserved port that is available on the host running the CA Top Secret LDAP Bridge. Available ports are usually above 1023.

// SSLPORT='636',

Import the Test Digital Certificate

As delivered, the CA Top Secret LDAP Bridge has three certificate files that enable the CA Top Secret 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 CA Top Secret LDAP Bridge certificate from a recognized certificate authority. To test, however, you can use the files delivered in the *sdir*/samples directory: ca_cert.pem, server_cert.pem and server_key.pem.

In general, to establish an SSL/TLS session, the CA Top Secret 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*/samples/ca_cert.pem contains this test CA certificate.

First, you should download *sdir*/samples/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 should 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. You should 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, you should 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 in production, your CA Top Secret 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 CA-Top Secret using the EXPORT command. For more information on digital certificates and the EXPORT command, see the *e Trust CA-Top Secret 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*/samples 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*/samples/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 should have access to this file. This can be implemented by the following commands:

```
cd /sdir/samples
chown userid ./server_key.pem
chmod 0400 ./server_key.pem
```

Where *userid* is the CA-Top Secret userid for the CA Top Secret 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 you should modify these parameters in slapd.conf as well.

Parameter	Description
TLSRandFile	The path the entropy seed used to generate encryption keys. This file (default: *sdir*/entropy.rnd) is generated at start-up by the doslapd script.
TLSCACertificateFile	The path the Certificate Authority Certificate, in base64 format. The delivered value is *sdir*/samples/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 should modify this parameter to point to this new file.
TLSCertificateFile	The path the Connector Certificate, in base64 format. The delivered value is *sdir*/samples/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 should modify this parameter to point to this new file.
TLSCertificateKeyFile	The path the Connector Certificate Private Key, in base64 format. The delivered value is *sdir*/samples/server_key.pem. If you order your own connector certificate, the certificate request should generate 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 should 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, which should be sufficient for most uses.
TLSVerifyClient	Determines whether the connector will require client certificate authentication. As delivered, this is set to never.

Tuning the CA Top Secret LDAP Bridge

This section describes the tuning parameters, operational choices, and optional features that are available with the CA Top Secret LDAP Bridge.

Slapd.conf Configuration File

In the ***sdir*/samples** directory, where ***sdir*** is the HFS directory you created for the CA Top Secret LDAP Bridge, the **slapd.conf** file contains the following online configuration parameters for your site.

Parameter	Description
Include	slapd.conf includes other files from the *sdir*/ schema directory: • core.schema contains basic LDAP object definitions • inetorgperson.schema contains standard definitions of an internet user • hpv33t.ldif contains CA Top Secret LDAP Bridge LDAP definitions • slapd.acl.conf contains the access control rules for your CA Top Secret LDAP Bridge
Loglevel	Not used in this file, as it is overridden by the DEBUG parameter setting within the START job.
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 CA Top Secret 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 CA-Top Secret database.
Timelimit	Controls the maximum number of seconds that the CA Top Secret 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.

Slapd.tss.conf Backend Configuration File

The slapd.tss.conf file contains the following online configuration parameters specific to your CA-Top Secret security system.

Parameter	Description
Database	This parameter must always be set to “bdb.”
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.
Lastmod	Controls whether the CA Top Secret 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 CA Top Secret LDAP Bridge. There must be one suffix parameter: o=sdir
Directory	This parameter must be set to *sdir*/bdb.
Mode	Sets the UNIX file permissions for the directory files that are created in the *sdir*/bdb directory. Set this parameter to 0770 to allow read-write access to these files for users in the UNIX group belonging to the user ID that ran the SLVCONVT job.
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 may want to add additional index statements as described in the section below. At minimum, 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*/samples/slapd.tss.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.

The DEBUGL Parameter in TSSCONV

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.

STDENV: UNIX Environment Variables

The stdenv files in *sdir*/samples 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.tss2ldap** - 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
Path	Searches the shell executables (/bin), the CA Top Secret LDAP Bridge executables (sdir/sbin), and the CA Top Secret LDAP Bridge samples directory (sdir/samples).
_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 should be delivered optimized for the CA Top Secret LDAP Bridge.
_CEE_RUNOPTS:H	Controls the size of the overall storage heap in UNIX. This parameter should be delivered optimized for the CA Top Secret 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 should be delivered optimized for the CA Top Secret LDAP Bridge.
_CEE_RUNOPTS: HEAPPOOLS	Controls the size of the pre-allocated storage pools in the storage heap. These should be delivered optimized for the CA Top Secret LDAP Bridge.
LDAPBRIDGE_LOACALE=Xx_XX.IBM-nnn…	Allows the LDAP Bridge to handle and store non-IBM code page 1047 characters that are supported by the Top Secret database. For example: LDAPBRIDGE_LOACALE=Fr_FR.IBM-297 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.tss2ldap files to enable processing of characters from code pages other than IBM the 1047 codepage.

DB_CONFIG: database variables

The DB_CONFIG files in *sdir*/samples 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* should be set to 0. *bytes* should be the size of sdir/bdb/secs/ldif2entry.bdb + 20%. *number_of_caches* should 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 may 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:

   ```
   ./samples/slapd.tss.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:

   ```
   ./samples/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/tss
   ```

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:

```
./samples/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:

```
./bdb/tss/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 datasets 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.

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 CA Top Secret LDAP Bridge can acquire as much storage as it needs. As delivered, the CA Top Secret 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 CA Top Secret LDAP Bridge in an initiator that permits sufficient storage. Similarly, the DOLDIF portion of the SLVCONVT job also requires considerable storage. Setting REGION=0M is also recommended.

However, in both these jobs, specifying REGION=0M does not always guarantee enough memory. See Assuring 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 CA Top Secret LDAP Bridge can run. Without this parameter, the CA Top Secret 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 CA Top Secret LDAP Bridge is run in a class that permits no time restrictions.

The ATTR file

The *SQUAL*.ATTR file determines which CA-Top Secret fields and profile types are exposed in your CA Top Secret LDAP Bridge. You can modify this file to add, remove or modify fields, depending on the needs of your client LDAP applications. If your LDAP client applications require access to security fields other than those specified in the default ATTR file, use the following table as a guide for editing the ATTR file.

Column	Name	Description
001	Used	The following settings are valid: Y - Directs the CA Top Secret LDAP Bridge to expose this field to the LDAP directory. N - Directs the CA Top Secret LDAP Bridge to not expose this field to the LDAP directory.
006 - 025	Field Name	CA-Top Secret security field name. Do not change.
026 - 045	Attribute	LDAP attribute name. You can change this attribute name, but if you create a new one, you should make sure that it is defined at the top of *sdir*/schema/tss.schema and also present in the MAY clause for the appropriate objectclasses defined later in that file.
046 - 125	Description	Description of the current field. Do not change.
126 – 133	Format	The format of this field. For comment only.
134 – 141	Rectype	The type of security record to be unloaded.
146 - 149	Offset	The offset of the field to be unloaded.
154 – 157	Length	The length of the field to be unloaded.
158 – 161	ID1 Offset	The offset of the first (low-order) dn attribute value.
162 - 165	ID1 Length	The length of the first (low-order) dn attribute value.
166 – 185	Profile Type	The profile type.
186 – 205	Syntax Rule	The data manipulation rule, if any, applicable to this field. Delivered rules are described below.
206 – 209	ID2 Offset	The offset of the second dn attribute value, if any.
210 - 213	ID2 Length	The length of the second dn attribute value, if any.
214	Append flag	Directs SLCONVT to append the attribute value.
215 – 219	ID3 Offset	The offset of the second dn attribute value, if any.

Column	Name	Description
220 - 222	ID3 Length	The length of the second dn attribute value, if any.
223 – 227	ID4 Offset	The offset of the second dn attribute value, if any.
228 - 231	ID4 Length	The length of the second dn attribute value, if any.

Note: The CA Top Secret LDAP Bridge cannot access or convert encrypted fields, and verifies all user ID and password combinations by making API calls to CA-Top Secret. The CA Top Secret LDAP Bridge does not store passwords in any form.

By changing the values in the Used column, you can control which attributes are exposed. You can also change the way various attributes are converted, as described below:

Syntax Rules

The following table describes the syntax rules that you can specify.

Rule	Description
USEREXIT	Call the appropriate user-exit to perform this manipulation: SLVCONVTU. Described below.
LASTNAME	Extract the last string from the field value.
FIRSTNAME	Extract the first string from the field value.
EMAIL	Create an email address from the *first* and *last* strings in the value: *first*.*last*@company.com
DNUSER	Use the field value to create an LDAP distinguished name for a user entry: uid=*value*,ou=people,o=*company*.
DNGROUP	Use the field value to create an LDAP distinguished name for a group entry: cn=*value*,ou=groups,o=*company*.
SETTOP	Set the value to TOP. Used for objectclasses.
SETVAL	Set the value to Field Name, as defined in columns 6 – 25 of this record.
BOOLEAN	Transform YES to TRUE, anything else to FALSE.
BASE64	Use this field to force a conversion to BASE64 when transferring information from Top Secret to the LDAP database. For example, this might be used for preserving leading spaces in the Instdata attribute.

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
TSSCONV	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*.MEMBERS 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
SLVCONVTF	COBOL	Filter user-exit called by SLVCONVT, the CA-Top Secret conversion process. Filters the CA-Top Secret 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.
SLVCONVTU	COBOL	Rule user-exit called by SLVCONVT, the CA-Top Secret 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 email 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.

z/OS File 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 SLVCONVT jobs:

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

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

UNIX File Security

All UNIX files and directories have an owner ID, a group ID, and a three-byte number that represents file permissions. This section assumes familiarity with these components, as well as the chown, chgrp and chmod UNIX commands. If you are unfamiliar with these parameters, refer to Chapter 14 of the *IBM UNIX System Services Users Guide*.

The following settings are recommended for UNIX security as well as for the user IDs that run the various jobs in the JCLLIB data set.

Component	Parameter	Recommendation
sdir directories and files	owner-ID	The CA-Top Secret user ID of the person who performed the installation.
sdir directories and files	group-ID	The UNIX default group ID for the owner ID above, which is associated with the user ID by CA-Top Secret. The group must also contain the user IDs under which the START, and SLVCONVT jobs run.
sdir directories and files	File Permissions	A setting of 770 allows read-write-execute access to the protected directories for the owner of the directory and members of the owner's group.
User ID for the START job	User ID in JOBCARD	This CA-Top Secret user ID must be that of either the owner ID, or a member of the group ID.
User ID for the SLVCONVT job	User ID in JOBCARD	This CA-Top Secret user ID must be that of either the owner ID, or a member of the group ID.

Using the recommended configuration assures that the various mainframe jobs have sufficient file access permissions to the required UNIX files, while preventing access to other users. You must verify that only user IDs of those required to maintain and operate the CA Top Secret LDAP Bridge are members of the specified group ID.

LDAP Security

The CA Top Secret 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 CA Top Secret LDAP Bridge, and provides example scenarios to help create ACLs that meet your site's requirements.

ACLs are defined within the *sdir*/samples/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 CA Top Secret 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 CA Top Secret LDAP Bridge processes ACLs by selecting the first ACL definition in slapd.acl.conf that applies to the specified <*db entries*>. Once found, the CA Top Secret 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.

CA Top Secret LDAP Bridge Default Settings

As delivered, the CA Top Secret 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 CA Top Secret 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 CA Top Secret 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 CA Top Secret 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 CA Top Secret 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 CA Top Secret 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/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 you specified for the CA Top Secret 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/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=groups,o=*company*"	group/tssGroup/member.regex represents group IDs that match the specified characteristics. *.* is a regular expression used to filter user entries. For example, M.* would permit all group IDs beginning with M. *Company* represents the root dn you specified for the CA Top Secret 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.

ACL Variable	Syntax	Meaning
<ldap attr>	none	
<user/group>	peername.ip=*IP ADDRESS*	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 CA Top Secret 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 CA Top Secret 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 CA Top Secret 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.

A Appendix:The LDAP Schema File

The CA Top Secret LDAP Bridge interacts with OVSI using a mapping file (TSS.xml) that is provided by OVSI and a schema provided by the CA Top Secret LDAP Bridge. See the OVSI documentation for information on this mapping file. The schema file is described in this Appendix.

General Information

The *sdir*/schema contains the LDAP schema files used by the CA Top Secret LDAP Bridge. By default, these files contain all necessary attributes and objectclasses to support the definitions in the ATTR file, whether or not these definitions are enabled there. Because of this, you should only have to modify a schema file in the following cases:

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

The schema files contain definitions of this format:

Attribute Definitions

At the top of the schema file, you'll find attribute definitions. To change an attribute name, locate that attribute and modify the name. To create a new one, find a similar attribute definition and copy it. Here is a typical attribute definition:

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

Attribute definitions support the following statements:

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

ObjectClass Definitions

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

objectclass (1.3.6.1.4.1.12471.1.2.2.1
NAME 'tssAcid'
DESC 'Acid Class for CA-Top Secret Connector'
SUP inetOrgPerson
STRUCTURAL

MAY (

cn $ givenName $ mail $
member $ o $ ou $ sn $
telephoneNumber $ tssAcidSize $ tssAcidType $ tssAction $
tssAdminAcid $ tssAdminData $ tssAdminFacility $ tssAdminMisc1 $
tssAdminMisc2 $ tssAdminMisc3 $ tssAdminMisc8 $ tssAdminMisc9 $
tssAdminRdtResource $ tssAdminResource $ tssAdminScope $ tssAfter $
tssAttribute $ tssBefore $ tssBypass $ tssCalendar $
tssCommand $ tssCreateDate $ tssDefnode $ tssDefNodes $
tssDeptAcid $ tssDeptName $ tssDfltGrp $ tssDivAcid $
tssDivName $ tssExpireDate $ tssFacility $ tssFor $
tssGroup $ tssInstdata $ tssLanguage $ tssLastModifyAcid $
tssLastModifyDate $ tssLastModifySmfId $ tssLastModifyTime $ tssLastUsedCount $
tssLastUsedCpu $ tssLastUsedDate $ tssLastUsedFacility $ tssLastUsedTime $
tssLinuxNam $ tssLnxents $ tssLockTime $ tssMasterFacility $
tssMcsAltG $ tssMcsAuth $ tssMcsAuto $ tssMcsCmds $
tssMcsDom $ tssMcsKey $ tssMcsLevl $ tssMcsLogc $
tssMcsMFrm $ tssMcsMgId $ tssMcsMon $ tssMcsRout $
tssMcsStor $ tssMcsUd $ tssOmvsAsSize $ tssOmvsCpuTm $
tssOmvsDfltgrp $ tssOmvsFileP $ tssOmvsGid $ tssOmvsHome $
tssOmvsMmapArea $ tssOmvsProcUser $ tssOmvsProgram $ tssOmvsThreads $
tssOmvsUid $ tssOpclass $ tssOpident $ tssOpprty $
tssOwn $ tssPhyskey $ tssPriv $ tssProfile $
tssPswdExpireDate $ tssPswdInterval $ tssSctykey $ tssSitran $
tssSmsAppl $ tssSmsData $ tssSmsMgmt $ tssSmsStor $
tssSource $ tssSuspend $ tssSuspendDate $ tssTimeZone $
tssTsoCommand $ tssTsoDefPrfG $ tssTsoDest $ tssTsoHClass $
tssTsoJClass $ tssTsoLAcct $ tssTsoLProc $ tssTsoLSize $
tssTsoMClass $ tssTsoMSize $ tssTsoOption $ tssTsoSClass $
tssTsoUData $ tssTsoUnit $ tssUntil $ tssXauth $
tssXCommand $ tssZoneAcid $ tssZoneName $ uid $
userPassword)

Objectclass definitions support the following statements:

Statement	Description
objectclass	Constant identifying this as an attribute definition. Must be followed by attribute definitions enclosed in parentheses.
NAME	The name of this attribute, enclosed in single quotes.
DESC	An optional description, enclosed in single quotes.
SUP	The superior objectclass, in the objectclass inheritance tree. Entries defined in this objectclass inherit all attributes for the superior objectclasses.
STRUCTURAL	Indicates that this is a structural objectclass, and thus subject to inheritance rules.
MAY	A list of optional attributes that can be present for this entry, enclosed in parentheses and delimited by " $ ". If you add, modify or delete any attributes names, you should make corresponding changes to this list.
MUST	A list of required attributes that can be present for this entry, enclosed in parentheses and delimited by " $ ". If you add, modify or delete any attributes names, you should make corresponding changes to this list of that attribute appears here.

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

CA-Top Secret Mapping Information

The CA Top Secret LDAP Bridge uses LDAP attributes that map to specific fields within the CA-Top Secret database. The table below lists all CA-Top Secret fields and their corresponding LDAP attributes.

LDAP Attribute Name	Read Only	TSS Field/ Value	Description
cn		NAME	Required on LDAP add operations.
givenName		N/A	Read-only, derived from NAME.
mail		N/A	Read-only, derived from NAME.

LDAP Attribute Name	Read Only	TSS Field/ Value	Description
sn		N/A	Read-only, derived from NAME.
tssAcidSize		SIZE	Read-only.
tssAcidType		TYPE	Valid values include USER PROFILE GROUP DEPARTMENT DIVISION ZONE DCA VCA ZCA LSCA SCA. For testing, please use USER, PROFILE or GROUP.
tssAdminAcid		ADMIN ACID	Multi-valued. An ACID under administrative control of this ACID.
tssAdminData		ADMIN DATA	Multi-valued. Administrative authorities for DATA.
tssAdminFacility		ADMIN FACILITY	Multi-valued. Administrative authorities for facilities.
tssAdminMisc1		ADMIN MISC1	Multi-valued. Administrative authorities for MISC1.
tssAdminMisc2		ADMIN MISC2	Multi-valued. Administrative authorities for MISC2.
tssAdminMisc3		ADMIN MISC3	Multi-valued. Administrative authorities for MISC3.
tssAdminMisc8		ADMIN MISC8	Multi-valued. Administrative authorities for MISC8.
tssAdminMisc9		ADMIN MISC9	Multi-valued. Administrative authorities for MISC9.
tssAdminRdtResource		ADMIN resource	Multi-valued. Administrative authorities for a resource class defined in the RDT.
tssAdminResource		ADMIN RESOURCE	Multi-valued. Administrative authorities for RESOURCE.
tssAdminScope		ADMIN SCOPE	Multi-valued. Scope of administrative authorities.
tssAfter		AFTER	For use only with PROFILE (tssProfile) keyword.
tssAttribute		Various	Various TSS attributes, like DUFXTR, DUFUPD, etc.
tssBefore		BEFORE	For use only with PROFILE (tssProfile) keyword.
tssBypass		Various	Bypass attributes, such as NODSNCHK.
tssCalendar		CALENDAR	Assign a calendar to the SDT ACID.

LDAP Attribute Name	Read Only	TSS Field/ Value	Description
tssCommand		COMMAND	Used for the Limited Command Facility.
tssCreateDate		n/a	Read-only. The date the ACID was created.
tssDefNodes		DEFNODES	Multi-valued. Default remote node ID.
tssDeptAcid		DEPARTMENT	Department ACID, required for LDAP add operations.
tssDeptName		n/a	Read-only. The name of the department for this ACID.
tssDfltGrp		DFLTGRP	The default OMVS group.
tssDivAcid		DIVISION	Division ACID.
tssDivName		n/a	Read-only. The name of the division for this ACID.
tssExpireDate		EXPIRE	The date on which this ACID will expire. Only used on LDAP modify/delete operations. To set an expire date, use tssFor or tssUntil.
tssFacility		FACILITY	A facility assigned this Acid.
tssFor		FOR	An expiration interval, in days.
tssGroup		GROUP	A group added to this ACID. On LDAP add operations, either PROFILE or GROUP can be specified, but not both.
tssInstdata		INSTDATA	Site-defined, ACID-specific data.
tssLanguage		LANGUAGE	The one-byte language character of this ACID.
tssLastModifyAcid		n/a	Read-only. The last ACID to have modified this ACID.
tssLastModifyDate		n/a	Read-only. The last date this ACID was modified.
tssLastModifySmfId		n/a	Read-only. The SMF system-ID of the last modification to this ACID.
tssLastModifyTime		n/a	Read-only. The last time this ACID was modified.
tssLastUsedCount		n/a	Read-only. The number of times this ACID signed-on.
tssLastUsedCpu		n/a	Read-only. The CPU ID of the last ACID sign-on.

LDAP Attribute Name	Read Only	TSS Field/ Value	Description
tssLastUsedDate		n/a	Read-only. The date of the last ACID sign-on.
tssLastUsedFacility		n/a	Read-only. The facility of the last ACID sign-on.
tssLastUsedTime		n/a	Read-only. The time of the last ACID sign-on.
tssLinuxNam		LINUXNAM	The Linux user-name assigned this ACID.
tssLnxents		LNXENTS	The Linux sign-on attributes, including: facility, UID, home, group_ACID. Comma-separated list.
tssLockTime		LTIME	The maximum idle time, in minutes, before a session locks.
tssMasterFacility		MASTFAC	The master facility assigned this ACID.
tssMcsAltG		MCSALTG	The master console alternate group.
tssMcsAuth		MCSAUTH	Multi-valued: master console authorizations.
tssMcsAuto		MCSAUTO	The master console AUTO attribute: YES or NO.
tssMcsCmds		MCSCMDS	Multi-valued: master console commands.
tssMcsDom		MCSDON	The master console should receive delete operator messages. YES or NO.
tssMcsKey		MCSKEY	The master console key.
tssMcsLevl		MCSLEVL	Mutli-valued: master console levels.
tssMcsLogc		MCSLOGC	Master console hardcopy logging.
tssMcsMFrm		MCSMFRM	Master console message format.
tssMcsMgId		MCSMGID	Master console migration ID.
tssMcsMon		MCSMON	Master console monitoring.
tssMcsRout		MCSROUT	Master console routing codes.
tssMcsStor		MCSSTOR	Master console storage in megabytes.
tssMcsUd		MCSUD	Master console undelete operator messages.

LDAP Attribute Name	Read Only	TSS Field/ Value	Description
tssOmvsAsSize		ASSIZE	OMVS maximum address space size.
tssOmvsCpuTm		OECPUTM	OMVS maximum CPU time.
tssOmvsDfltgrp		DFLTGRP	OMVS default group.
tssOmvsFileP		OEFILEP	OMVS maximum open files.
tssOmvsGid		GID	OMVS Group ID.
tssOmvsHome		HOME	OMVS home directory.
tssOmvsMmapArea		MMAPAREA	OMVS maximum MMAP area size.
tssOmvsProcUser		PROCUSER	OMVS maximum users.
tssOmvsProgram		OMVSPGM	OMVS initial program.
tssOmvsThreads		THREADS	OMVS maximum threads.
tssOmvsUid		UID	OMVS UID.
tssOpclass		OPCLASS	Multi-valued: CICS operator class.
tssOpident		OPIDENT	CICS operator identity.
tssOpprty		OPPRTY	CICS operator priority.
tssOwn		A resource class from the RDT	Owned resource. Format class(resource). Accepts other keywords such as UNDERCUT, NOPERMIT, etc.
tssPhyskey		PHYSKEY	Key for external authentication devices.
tssProfile		PROFILE	Profile assigned to this ACID.
tssPswdExpireDate		n/a	Date on which the password expires.
tssPswdInterval		PASSWORD	Password expire interval.
tssSctykey		SCTYKEY	CICS security key.
tssSitran		SITRAN	CICS start transaction.
tssSmsAppl		SMSAPPL	Default SMS application ID.
tssSmsData		SMSDATA	Default SMS data class.
tssSmsMgmt		SMSMGMT	SMS management class.
tssSmsStor		SMSSTOR	SMS storage class.
tssSource		SOURCE	Multi-valued: source reader or terminal for ACID entry.
tssSuspend		n/a	Boolen: indicates whether user is suspended.

LDAP Attribute Name	Read Only	TSS Field/ Value	Description
tssSuspendDate		SUSPEND FOR|UNTIL	Date on which a suspension ends.
tssTimeZone		TZONE	Time zone of ACID, range -12 to +12.
tssTsoCommand		TSOCOMMAND	Initial TSO Command.
tssTsoDefPrfG		TSODEFPRFG	TSO default performance group.
tssTsoDest		TSODEST	TSO default destination ID for JCL.
tssTsoHClass		TSOHCLASS	TSO default hold class.
tssTsoJClass		TSOJCLASS	TSO default job Class.
tssTsoLAcct		TSOACCT	TSO default account number.
tssTsoLProc		TSOLPROC	TSO default logon procedure.
tssTsoLSize		TSOLSIZE	TSO default region size, in kilobytes.
tssTsoMClass		TSOMCLASS	TSO default message class.
tssTsoMSize		TSOMSIZE	TSO maximum region size, in kilobytes.
tssTsoOption		TSOOPT	Multi-valued: TSO options, such as MAIL or NOMAIL.
tssTsoSClass		TSOSCLASS	TSO default SYSOUT class.
tssTsoUData		TSOUDATA	TSO user data (4 bytes, hexadecimal characters: 0 – F).
tssTsoUnit		TSOUNIT	TSO default unit.
tssUntil		UNTIL	Date on which this ACID expires.
tssUserResource		A site-defined resource class	User-defined resource owned by this ACID, format class(resource).

LDAP Attribute Name	Read Only	TSS Field/ Value	Description
tssXauth		A class from the RDT, or "ACID".	Permitted resource for this user, format class(resource) [ACCESS(level)] [keyword(value) …] • Class can be any class defined in the RDT, or ACID • Level can be any defined access level: READ, etc. • Keyword can be one of: ACTION, APPLDATA, FACILITY, FOR, LIBRARY, MAPREC, MASKREC, MODE, PRIVPGM, SELECT, SYSID, TIMEREC, TIMES, UNTIL, VMUSER (for CPCMD only)
tssXCommand		XCOMMAND	An excluded command for the Limited Command Facility (LCF).
tssZoneAcid		ZONE	Used only on LDAP add operations. The ZONE for this ACID. Do not specify for testing: all ACIDs are automatically defined in ROCZONE.
tssZoneName		n/a	The name of this zone.
uid		ACID	The ACID. Required on LDAP Add operations.
userPassword		PASSWORD	Password for this ACID. Required on LDAP add operations.

Index

P

R

S

T

U

Z