HP OpenView Select Identity RACF LDAP Bridge

RACF LDAP Bridge Version: 3.3

Installation and Configuration Guide

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

Legal Notices

Warranty

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

The information contained herein is subject to change without notice.

Restricted Rights Legend

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

Copyright Notices

This product includes software developed by the Apache Software Foundation (http://www.apache.org/). Portions Copyright (c) 1999-2003 The Apache Software Foundation. All rights reserved.

Select Identity uses software from the Apache Jakarta Project including:

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

Additional third party software used by Select Identity includes:

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

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

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 Select Identity RACF LDAP Bridge is an LDAP gateway that provides access to the RACF database. By enabling you to access mainframe security-based data with LDAP, the RACF LDAP Bridge allows you to extend mainframe authentication and authorization to your environment.

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.
- RACF concepts such as password verification and resource authorization.
- Have RACF authority to create data sets and HFS files.

About the Select Identity RACF 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 RACF.

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 RACF LDAP Bridge.

RACF LDAP Bridge

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

- Mirror database
- racf2ldap
- ldap2racf
- Configuration database

Mirror database

The mirror database represents a real-time image of the entire RACF database as it resides on the host z/OS system. The RACF database and mirror database are automatically updated with the racf2ldap and ldap2racf synchronization processes.

racf2ldap

racf2ldap updates the mirror database to reflect the current status of the RACF database.

Whenever a change is made to the RACF database, racf2ldap intercepts the SMF record generated by the RACF command. The RACF command is then translated into an equivalent LDAP modify command that updates the mirror database accordingly.

ldap2racf

ldap2racf modifies the mirror database to reflect changes initiated within the RACF LDAP Bridge.

Whenever users make a change to the RACF database, ldap2racf translates the LDAP modify command into an equivalent RACF command to update the RACF database accordingly. Once the change has been made to the RACF database, racf2ldap 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 RACF LDAP Bridge.

2 Installing and Configuring the RACF LDAP Bridge

The RACF LDAP Bridge enables HP OpenView Select Identity to perform tasks on RACF databases.

The following tasks are supported by the RACF 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 RACF LDAP Bridge.

z/OS Requirements

The RACF LDAP Bridge requires the following elements:

- OS/390 V2R10M0, z/OS V1R1M0 or later
- RACF releases included with above versions of z/OS
- UNIX System Services
- TCP/IP for OS/390 or z/OS
- LE runtime libraries, with C-language support

TCP/IP Requirements

The RACF LDAP Bridge can communicate using any port. The default ports for LDAP communications are 2389 for unencrypted data, and 2390 for SSL-encrypted data.

Before You Begin

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

Configuring UNIX System Services

The RACF 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 RACF LDAP Bridge. The amount of disk space required for the directory can be determined using the following formula:

  Disk Space = 200MB + (size of RACF database x 3.2)
- For the RACF LDAP Bridge to function correctly, the parent directories of the RACF 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 RACF LDAP Bridge to function correctly, the directory for the RACF LDAP Bridge itself must have appropriate permissions:
  - OWNER: read/write/execute
  - GROUP: read/write/execute
  - OTHER: execute

  If, for example, you are installing HPV33R into the /usr/lpp/hpv33r directory, assign the appropriate permissions by issuing the following commands:

  chmod 0771 /usr/lpp/hpv33r
- The group owner of the HPV33R directory must be a RACF group of which the user-ID associated with the RACF LDAP Bridge started task is a member. If, for example, the HPV33R directory is /usr/lpp/hpv33r, and you plan to run the RACF LDAP Bridge under a user-ID that is a member of the ADMIN group, then the group owner of the /usr/lpp/hpv33r should be ADMIN. To see the group owner of /usr/lpp/hpv33r, issue the following command:

  ls -ld /usr/lpp/hpv33r

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

  chgrp ADMIN /usr/lpp/hpv33r

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

Configuring Your z/OS TCP/IP and Firewall

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

- Port 2389
- Port 2390 (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 RACF LDAP Bridge, you must modify your firewall to enable access ports 2389 or 2390.

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

Assuring Sufficient Region Size

RACF 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 RACF 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 may be specific limitations that apply regardless of the REGION=0M parameter. These limitations, usually coded in an IEFUSI user-exit, may 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 RACF LDAP Bridge can allocate a region size of 200 megabytes or more. If a process fails to allocate memory, it may exit with a return code 9. This indicates that the region size is too small and needs to be adjusted upwards.

Verify RACF Access to Control and Authorize FACILITY Class Resources

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

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

Setting the RACF System Options (SETROPTS)

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

SETROPTS AUDIT(*) SAUDIT OPERAUDIT

where:

- The AUDIT(*) parameter instructs RACF to create SMF records whenever any RACF profiles are added, modified, or deleted. Without these SMF records, racf2ldap cannot propagate RACF changes to the RACF LDAP Bridge.

- The SAUDIT parameter instructs RACF to create SMF records whenever RACF profiles are changed by administrators with the SPECIAL and GROUP-SPECIAL attributes. Without these SMF records, racf2ldap cannot propagate RACF changes made by these administrators to the RACF LDAP Bridge.
- The OPERAUDIT parameter instructs RACF to create SMF records whenever RACF profiles are changed by administrators with the OPERATION attribute. Without these SMF records, racf2ldap cannot propagate RACF changes made by these administrators to the RACF LDAP Bridge.

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

Verify RACF access to BPX.DAEMON

This section applies only if you have defined BPX.DAEMON to RACF.

The user ID under which the RACF LDAP Bridge LDAP executable is run must have RACF READ access to the BPX.DAEMON Facility 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*.

Assuring program control for the SCEERUN2 Library

If program control is active on your system, you may have to place the Language Environment library SCEERUN2 under program control.

The RACF LDAP Bridge requires BPX.DAEMON authority. With program control active, the RACF LDAP Bridge can run modules only from program controlled libraries. By default, the Language Environment library SCEERUN2 is not program controlled.

To place this library under program control, issue the following command from TSO:

RALTER PROGRAM * ADDMEM('*xxx*.SCEERUN2'//NOPADCHK)

Where *xxx* is the prefix for your language environment libraries (usually “CEE”). After performing this command, you will have to refresh the in storage program control tables by issuing the following command:

SETROPTS REFRESH WHEN(PROGRAM)

Alternatively, you may perform this refresh from option 5.6 of the RACF ISPF Administration Panels.

Installing from CD or using FTP

The CD or downloaded version of the RACF LDAP Bridge release media contains the compressed file hpv33r.pax.Z. hpv33r.pax.Z is used to install the RACF LDAP Bridge on an HFS file system.

Expanding the RACF LDAP Bridge PAX Archive

Insert the CD in a machine that has FTP transfer abilities to your z/OS system, then transfer the hpv33r.pax.Z file using FTP, to your HFS directory. During the transfer, be sure to specify binary mode. To expand the PAX file, type OMVS from a TSO command line, and issue the following commands:

```
cd sdir pax -rv -px -f hpv33r.pax.Z
```

Where *sdir* is the name of the HFS directory that you created for the RACF LDAP Bridge.

Running the Install Script

Execute the installation script to install the RACF 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 OS/390 or z/OS with the `doalloc` subscript.
- Moves the source, load, JCL, and attributes from UNIX System Services to OS/390 or z/OS with the `doget` subscript.
- Frees the file allocations for SRCLIB, LOADLIB, JCLLIB, and ATTR with the `dofree` subscript.
- Installs the RACF LDAP Bridge along with the racf2ldap and ldap2racf components.

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. Once 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 RACF LDAP Bridge. If you are using the default installation directory, the install script is located in */usr/lpp/hpv33r*. 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 must supply the following site-specific information:

SQUAL: Dataset Names

Default:

SQUAL='HPV33R'

The SQUAL variable represents the high-level qualifier for the RACF LDAP Bridge datasets. Enter a value that conforms to your site standards. It is recommended that you leave the high-level qualifier as HPV33R.

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

Default:

sdir='current directory'

The sdir variable represents the name of the product root directory. The default directory is directory from which the install script is being run.

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 RACF LDAP Bridge. Change the hostname operand to match the Domain Name or TCP/IP address of the host for the RACF LDAP Bridge. The default value that initially appears is the hostname defined for the z/OS system where the install script is running. For typical installations 127.0.0.1 will give the best performance.

Hostport DNS host port

Default:

hostport='2***389***'

The hostport variable represents the port number used for unencrypted communications by the host system. Change the 2389 operand to match the TCP/IP port used by the RACF LDAP Bridge for unencrypted communications. The default port is 2389. If you change this default port, you must use an unreserved port that is available on the host running the RACF 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 may 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 may 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 RACF 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 may 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 RACF LDAP Bridge.

Populating the LDAP Directory

The RACF LDAP Bridge uses a directory database that is populated with data from your RACF repositories. Once the RACF LDAP Bridge database is initially loaded, the racf2ldap and ldap2racf components of the RACF LDAP Bridge keep all databases synchronized.

Directory Load Conversion Jobs

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

The RACF user-ID under which you run the SLCONVR job must be a member of the RACF group that owns of the HFS directories for the RACF 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.

For the SLCONVR conversion job, change the input DSN to match the database or backup database that will be unloaded. The job creates temporary data sets (&&IRRDBU00) that may require a change to their SIZE parameters. The temporary data sets must be twice the size of the data set being unloaded.

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

To customize SLVCONVR, copy the information from the JOBCARD member into the SLVCONVR file and change the SYS1.RACF data set name in STEP 1 to match the RACF data set name at your site.

Once you have made the required 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 RACF LDAP Bridge requires approximately 200 MB of space in addition to the size of your RACF 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 RACF 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 SLCONVR 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 RACF LDAP Bridge. The RACF 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 RACF 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 SLCONVR.
- 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 RACF 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 SLCONVR 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.
- If you have to re-run this job to rebuild the database at a later date, you will need to perform one additional step to reload the RACF system options. From TSO or ISPF option 6, enter the following command:

  SETROPTS

After the racf2ldap component is installed, this command will reload the RACF system options.

Configure and Start the RACF LDAP Bridge

Configure and start the RACF LDAP Bridge by performing the tasks detailed in the sections below.

Note: You must ensure that permissions for all files and directories within your installation directory are set to 0770 by issuing the following command:

chmod -R 770 *sdir*

Verify Sufficient z/OS Resource Allocations

As the RACF LDAP Bridge is a high-volume online process, it 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 RACF LDAP Bridge environment.

Verify Module File Attributes

The RACF LDAP Bridge LDAP modules slapd, pwdbind, and ldap2racf must all be APF-Authorized and Program-Controlled in order to perform authentications against RACF.

Verify that these modules have the required authorizations by entering the following command for each file:

```
extattr sdir/sbin/module
```

where ***sdir*** is the name of the RACF LDAP Bridge HFS directory and ***module*** is one of the following LDAP module names:

- slapd
- ldap2racf.so
- pwdbind.so
- authtso
- libshr.so

The APF-Authorized and Program-Controlled parameters must both be set to “YES” for each of the modules. If any module does not have the required authorizations, enter OMVS from TSO and issue the following command:

```
extattr +ap sdir/sbin/module
```

Starting the RACF LDAP Bridge

You can run the RACF 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 RACF LDAP Bridge as a started task or a submitted job, the RACF user-ID of the RACF LDAP Bridge must be a member of the RACF group that owns the HFS directories for the RACF 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 RACF 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 RACF LDAP Bridge.

Started Tasks

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

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

Testing the RACF LDAP Bridge

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

Running the Dotestserver Script

To test the RACF 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 RACF user ID and password. This test should return information on your RACF user ID as stored in the LDAP repository.

Stopping the RACF LDAP Bridge

Successful completion of the tests described above indicates that the RACF LDAP Bridge is running properly on your system. To conclude testing, stop the RACF 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 RACF 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 RACF 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 RACF 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 may also have to adjust the REGION parameter in the START JCL, as described in Assuring Sufficient Region Size.

3 Installing and Configuring Racf2ldap

This chapter describes how to install and configure the racf2ldap portion of the RACF LDAP Bridge. Racf2ldap synchronizes RACF changes with the RACF LDAP Bridge.

Installing Racf2ldap

Racf2ldap runs as a stand-alone UNIX daemon in a separate address space from the RACF LDAP Bridge. It reads the SMF records generated whenever RACF changes are made, and propagates the changes to the RACF LDAP Bridge using LDAP. The SMF records are written to the ***sdir*/racf2ldap/new** directory by the SLAPU83 program that runs in the SMF user exit points **SYSSTC.IEFU83**, and either **SYSTSO.IEFU83**, **SYSJES2.IEFU83**, or **SYS.IEFU83**.

To use racf2ldap, you must activate the SMF user exits described below.

Enabling the IEFU83 User Exit Points

Before implementing the racf2ldap IEFU83 user-exit program, you must verify that user-exit points are enabled on your system for the following environments:

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

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

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

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

The procedure for enabling SYSSTC.IEFU8, SYSTSO.IEFU83, SYSJES2.IEFU83, and SYS.IEFU83 is described in the IEFU83 section of the *IBM z/OS MVS Installation Exits Manual*. This information is available from IBM online at the following location:

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

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

1. Edit the **SMFPRMnn** member of the SYS1.PARMLIB data set, where *nn* is the SMF parameter member currently active on your system.

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

   SUBSYS(STC,EXITS(IEFU83,**xxx**))

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

3 If TSO is defined as a separate SMF subsystem, then this member contains a statement starting with "SUBSYS(TSO)". In this case, verify that IEFU83 is specified in the EXITS clause parameters. For example:

   SUBSYS(TSO,EXITS(IEFU83,**xxx**))

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

4 If JES2 is defined as a separate SMF subsystem, then this member contains a statement starting with "SUBSYS(JES2)". In this case, verify that IEFU83 is specified in the EXITS clause parameters. For example:

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

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

5 If neither TSO nor JES2 are defined as separate SMF subsystems, verify that IEFU83 is specified in the EXITS clause parameters for the SYS statement. For example:

   SYS(**xxx**,EXITS(IEFU83,**xxx**)**xxx**)

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

Activating the IEFU83 Dynamic User-exit Program

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

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

Activating SLAPU83 Dynamically

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

SETPROG EXIT,ADD,EXITNAME=SYSSTC.IEFU83,MODNAME=SLAPU83,
DSNAME=***SQUAL***.LOADLIB

and either:

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

or:

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

or:

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

where ***SQUAL*** is the high-level qualifier you created for the RACF LDAP Bridge.

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

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

If JES2 is defined as a separate SMF subsystem, use the command that references SYSJES2.IEFU83.

If neither TSO nor JES2 are defined as separate SMF subsystems, use the command that references SYS.IEFU83.

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

Activating SLAPU83 Permanently

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

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

   EXIT ADD
   EXITNAME(SYSSTC.IEFU83)
   MODNAME(SLAPU83)
   STATE(ACTIVE)
   DSNAME(***SQUAL***.LOADLIB)

   and either:

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

   or:

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

   or:

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

   where ***SQUAL*** is the high-level qualifier you created for the RACF LDAP Bridge. Alternatively, you can move SLAPU83 from *SQUAL*.LOADLIB to the LPALIB, in which case you can omit the DSNAME statement in the above example.

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

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

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

COM='SET PROG=*nn*'

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

Starting Racf2ldap

Racf2ldap starts automatically using the same START JCL that is used to start the RACF LDAP Bridge. Whenever you start the RACF LDAP Bridge, racf2ldap is also active.

If racf2ldap ever needs to be started separately from the RACF LDAP Bridge, submit the STARTR2L member of the SQUAL.JCLLIB.

Reloading the System Options

After starting the RACF LDAP Bridge for the first time, you will need to issue the following command (from TSO or ISPF Option 6) to load the RACF system options:

SETROPTS

You should also issue this command after you rebuild the database, as described in "Populating the LDAP directory".

Testing Racf2ldap

To test the racf2ldap daemon by running the dotestr2l script, follow the series of steps below:

1. Verify that the SLAPU83 program is enabled and start the RACF LDAP Bridge if it is not already running.
2. From TSO, issue the following command:

   ALTUSER *testuserID* NAME('RACF2LDAP TEST')

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

   cd /sdir/samples

   dotestr2l
5. At the prompts, enter your RACF user ID and password along with *testuserID*. This test should return the distinguished name of the entry along with the following text:

   cn: RACF2LDAP TEST

   If you do not receive this result, consult sdir/racf2ldap.log to determine the cause of the error.

Stopping Racf2ldap

Racf2ldap stops automatically using the same STOP JCL that is used to stop the RACF LDAP Bridge. Whenever you stop the RACF LDAP Bridge, racf2ldap also stops.

If racf2ldap ever needs to be stopped separately from the RACF LDAP Bridge, submit the STOPR2L member of the SQUAL.JCLLIB.

Recovering Data After Restarting Racf2ldap

As long as SYS.IEFU83 and SYSSTC.IEFU83 are active, all RACF changes are captured to the ***sdir*/racf2ldap/new** directory. If racf2ldap is stopped, RACF changes accumulate in the directory so none are lost when it is restarted.

Similarly, if the RACF LDAP Bridge is stopped, racf2ldap accumulates any RACF changes so none are lost when the RACF LDAP Bridge is restarted.

If the SYSSTC.IEFU83, SYSTSOIEFU83, SYSJES2.IEFU83, or SYS.IEFU83 user exits are disabled, RACF changes cannot be captured or propagated, and are therefore lost. The RACF LDAP Bridge cache must be rebuilt using the SLCONVR job.

Archiving RACF Changes

Once a RACF change has been processed, racf2ldap moves the SMF record from the ***sdir*/racf2ldap/new** directory to the ***sdir*/racf2ldap/old** or ***sdir*/racf2ldap/error** directories, where:

- **/old** acts as an archive of RACF SMF records that can be used for debugging purposes, or to rebuild the RACF database.
- **/error** acts as an holding area for RACF SMF 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 RACF 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 R2LCLEAN job to run regularly.

Setting the RETAIN parameter

The racf2ldap.conf configuration file contains the parameters that control the operation of racf2ldap. Within racf2ldap.conf, the RETAIN parameter determines how SMF records are to be archived by racf2ldap.

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

1. Open the racf2ldap.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 racf2ldap/old. Note that if you choose this setting, you will not have to schedule the R2LCLEAN job as no records are written to the archive.
- 0 = SMF records are written to racf2ldap/old and are not deleted.
- *nn* = SMF records are written to racf2ldap/old and records older than *nn* (0-999) days are deleted once the R2LCLEAN job is run.

Running R2LCLEAN

The R2LCLEAN job is used to purge the archive of SMF records that exceed the age limit specified by the RETAIN parameter. The R2LCLEAN 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 R2LCLEAN job if RETAIN=-1, as no SMF records are written to the archive.

4 Installing and Configuring Ldap2racf

Ldap2racf propagates changes made using the RACF LDAP Bridge to the RACF database. The RACF LDAP Bridge also installed ldap2racf as part of its installation procedure. As ldap2racf runs in the same address space as the RACF LDAP Bridge, you do not need to run ldapracf separately from the RACF LDAP Bridge. Whenever you start the RACF LDAP Bridge, ldap2racf is also active.

Testing Ldap2racf

To test Ldap2racf by running the dotestl2r script, follow the series of steps below:

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

   cd /sdir/samples

   dotestl2r
4. At the prompts, enter your RACF user ID and password along with a *newuserID* that will be created on your RACF database using ldap2racf.

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

   LU *newuserID*

   You should see output similar to the following for this new RACF user:

   USER=*newuserID* NAME=TEST LDAP2RACF OWNER=*racfuser* CREATED=04.267
   DEFAULT-GROUP=*racfgrp* PASSDATE=00.000 PASS-INTERVAL=180
   ATTRIBUTES=NONE
   REVOKE DATE=NONE RESUME DATE=NONE
   LAST-ACCESS=UNKNOWN
   CLASS AUTHORIZATIONS=NONE
   INSTALLATION-DATA=
   NO-MODEL-NAME
   LOGON ALLOWED (DAYS) (TIME)

   ANYDAY ANYTIME
   GROUP=*racfgrp* AUTH=USE CONNECT-OWNER=*racfuser* CONNECT-DATE=04.267
   CONNECTS= 00 UACC=NONE LAST-CONNECT=UNKNOWN
   CONNECT ATTRIBUTES=NONE

REVOKE DATE=NONE RESUME DATE=NONE
SECURITY-LEVEL=NONE SPECIFIED
CATEGORY-AUTHORIZATION
NONE SPECIFIED
SECURITY-LABEL=NONE SPECIFIED
***

Where:

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

— ***racfgrp*** is the default RACF group assigned to this user.

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

5 Operating and Tuning the RACF LDAP Bridge

This chapter describes how to operate and tune the RACF LDAP Bridge within your environment.

Running the RACF LDAP Bridge

You can run the RACF 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 RACF LDAP Bridge from MVS, using SDSF and the operator console.

Submitted Jobs

To run the RACF 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 RACF LDAP Bridge, submit the STOP member of the *SQUAL*.JCLLIB dataset.

Started Tasks

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

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

Running in the Foreground (OMVS or Telnet)

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

cd */sdir*/samples

doslap *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 RACF LDAP Bridge, issue the following command:

cd /*sdir*/samples

dostop

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

Running in the Background (OMVS or Telnet)

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

cd /sdir/samples

nohup doslap port sslport debug &

To stop the RACF LDAP Bridge, issue the following command:

cd /*sdir*/samples

dostop

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

Setting the Debugging Level

The RACF 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 RACF LDAP Bridge is started. To change the debugging level, stop the RACF LDAP Bridge, make the required changes, then restart the RACF 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 RACF 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 RACF LDAP Bridge. Available ports are usually above 1023.

// SSLPORT='636',

Import the Test Digital Certificate

As delivered, the RACF LDAP Bridge has three certificate files that enable the RACF 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 RACF LDAP Bridge certificate from a recognized certificate authority. To test, however, you may 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 RACF LDAP Bridge presents the client with its RACF LDAP Bridge 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 RACF LDAP Bridge 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 may 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 RACF LDAP Bridge Certificate

To implement in production, your RACF LDAP Bridge should use its own site-specific certificate. To do this, you may order a certificate from a variety of certificate authorities, including www.thawte.com, www.verisign.com, and www.rsasecurity.com. You may also generate a RACF LDAP Bridge certificate yourself from RACF using the RACDCERT command. However you acquire your RACF LDAP Bridge certificate, you must store that 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 RACF LDAP Bridge certificate. You can usually acquire this file directly from the CA web site. You may also export a CA certificate from RACF using the RACDCERT EXPORT command.
- **server_cert.pem** - The RACF LDAP Bridge certificate presented to clients during the SSL/TLS handshake to verify RACF LDAP Bridge identity and establish trust. This certificate must be signed by the CA referred to by the CA certificate, above.
- **server_key.pem** - The RACF LDAP Bridge 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 RACF LDAP Bridge or other security breaches. Ideally, only the user-id of the RACF LDAP Bridge should have access to this file. This may be implemented by the following commands:

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

Where *userid* is the RACF userid for the RACF 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 to 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 to 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 may 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 to the RACF LDAP Bridge Certificate, in base64 format. The delivered value is *sdir*/samples/server_cert.pem. If you order your own RACF LDAP Bridge certificate, you may either replace server_cert.pem with the new RACF LDAP Bridge certificate (in base64 format), or place the new RACF LDAP Bridge certificate into a new file. If you do the latter, you should modify this parameter to point to this new file.

Parameter	Description
TLSCertificateKeyFile	The path to the RACF LDAP Bridge Certificate Private Key, in base64 format. The delivered value is *sdir*/samples/server_key.pem. If you order your own RACF LDAP Bridge certificate, the certificate request should generate a private key file. You may 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 RACF LDAP Bridge will accept. The delivered value allows the RACF LDAP Bridge to accept high and medium strength ciphers, which should be sufficient for most uses.
TLSVerifyClient	Determines whether the RACF LDAP Bridge will require client certificate authentication. As delivered, this is set to never.

Tuning the RACF LDAP Bridge

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

Slapd.conf Configuration File

In the ***sdir*/samples** directory, where ***sdir*** is the HFS directory you created for the RACF 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 • hpv33r.ldif contains RACF LDAP Bridge LDAP definitions • slapd.acl.conf contains the access control rules for your RACF 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 RACF LDAP Bridge returns for an individual search operation. This parameter must be set to a number larger than the total number of groups in your RACF database.
Timelimit	Controls the maximum number of seconds that the RACF LDAP Bridge spends attempting to service a search operation.
Idletimeout	The number of seconds the RACF LDAP Bridge will keep an inactive session alive. Decreasing this parameter may 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.racf.conf Backend Configuration File

The slapd.racf.conf file contains the following online configuration parameters specific to your RACF 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 may require adjusting the heap parameter in the *sdir*/samples/stdenv.slapd file.
Lastmod	Controls whether the RACF 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 RACF 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 SLCONVR 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 may create additional index files to improve online performance.

Creating Additional Index files

To create additional index files, edit the ***sdir*/samples/slapd.racf.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 may 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 RACFCONV

The DEBUGL parameter within the RACFCONV 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 “-1” 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 RACF LDAP Bridge processing (START).
- **stdenv.slapadd** - Affects database load processing (SLCONVR)
- **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 RACF LDAP Bridge executables (sdir/sbin), and the RACF 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.”

Parameter	Description
_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 RACF LDAP Bridge.
_CEE_RUNOPTS: H	Controls the size of the overall storage heap in UNIX. This parameter should be delivered optimized for the RACF LDAP Bridge.
_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 RACF 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 RACF 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 RACF LDAP Bridge processing (START).
- **DB_CONFIG.slapadd** - Affects database load processing (SLCONVR)

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, but should not be set for the online RACF LDAP Bridge. DB_TXN_NOT_DURABLE controls whether the database logs changes for recovery. Speeds up database loads, but should not be set for the online RACF LDAP Bridge.

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.racf.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/racf
   ```

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/racf/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 RACF LDAP Bridge can acquire as much storage as it needs. As delivered, the RACF 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 RACF LDAP Bridge in an initiator that permits sufficient storage. Similarly, the DOLDIF portion of the SLCONVR 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 RACF LDAP Bridge can run. Without this parameter, the RACF 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 RACF LDAP Bridge is run in a class that permits no time restrictions.

The ATTR file

The *SQUAL*.ATTR file determines which RACF fields and profile types are exposed in your RACF LDAP Bridge. You may 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.

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

Column	Name	Description
001	Used	The following settings are valid: Y - Directs the RACF LDAP Bridge to expose this field to the LDAP directory. N - Directs the RACF LDAP Bridge to not expose this field to the LDAP directory.
006 - 025	Field Name	RACF security field name. Do not change.
026 - 045	Attribute	LDAP attribute name. You may 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/racf.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	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	Modify flag	No longer used.
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.

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

Data Manipulation Rules

The following table describes the delivered data manipulation rules you may specify.

Rule	System	Description
USEREXIT	All	Call the appropriate user-exit to perform this manipulation: SLCONVRU. Described below.
LASTNAME	All	Extract the last string from the field value.
FIRSTNAME	All	Extract the first string from the field value.
EMAIL	All	Create an email address from the *first* and *last* strings in the value: *first*.*last*@company.com
PASSWORD	All	Create a SASL password value, based on the *uid* and the security system (secs): {secs}uid. The result is all upper-case. For example: {RACF}TESTUSER.
DNUSER	RACF	Use the field value to create an LDAP distinguished name for a user entry: uid=*value*,ou=people,o=*company*.
DNGROUP	RACF	Use the field value to create an LDAP distinguished name for a group entry: cn=*value*,ou=groups,o=*company*.
USERFLD	RACF	Use the default naming to create a RACF user-defined field.
SETTOP	RACF	Set the value to TOP. Used for objectclasses.
SEGOC	RACF	Set the value to racf*Profile_Type*, where *Profile_Type* is defined in columns 166 – 185 of this record.
SETVAL	RACF	Set the value to Field Name, as defined in columns 6 – 25 of this record.
BOOLEAN	RACF	Transform YES to TRUE, anything else to FALSE.
WHENDAYS	RACF	Create a space-delimited list of days the user may access the system based YES/NO flags in the value.

Rule	System	Description
WHENTIME	RACF	Create a RACF time period specification for the hours the user may access the system based on the field value.
BOOLVAL	RACF	If the value is YES, transform the value to the Field Name specified in columns 6 – 25 of this record.
AUDIT	RACF	Create a RACF audit specification based on the field value.
ZONE	RACF	Create a RACF time zone specification based on the field value.
TMESPEC	RACF	Create a Tivoli Management Environment access specification for RACF based on the field value.

JCLLIB members

The *SQUAL*.JCLLIB MVS file, where *SQUAL* represents your high-level qualifier, contains several members you may 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.

Members	Statements	Description
KEY	KEYVAL	Contains the product key. Do not modify.
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* OS/390 repository
RACFCONV	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 (SLCONVRF) 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
SLCONVRF	COBOL	Filter user-exit called by SLCONVR, the RACF conversion process. Filters the RACF profiles loaded into the LDAP directory. By default, SLCONVR 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(RACFCONV), which must be set to YES for it to be enabled.
SLCONVRU	COBOL	Rule user-exit called by SLCONVR, the RACF 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 SLCONVR jobs:

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

where ***SQUAL*** represents the high-level qualifier you used for your RACF 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 RACF 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 RACF. The group must also contain the user IDs under which the START, and SLCONVR 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 RACF user ID must be that of either the owner ID, or a member of the group ID.
User ID for the SLCONVR job	User ID in JOBCARD	This RACF 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 RACF LDAP Bridge are members of the specified group ID.

LDAP Security

The RACF 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 RACF 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 RACF 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 RACF LDAP Bridge processes ACLs by selecting the first ACL definition in slapd.acl.conf that applies to the specified <*db entries*>. Once found, the RACF 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.

RACF LDAP Bridge Default Settings

As delivered, the RACF LDAP Bridge is configured to permit read-only 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 RACF 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.

ACL Variable	Syntax	Meaning
<*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 RACF 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 RACF 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 RACF 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 RACF 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/racfGroup/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/racfGroup/member.exact=”cn=*GROUP1*,ou=groups,o=*company*"	group/racfGroup/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 RACF 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/racfGroup/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/racfGroup/member.regex="cn=*.*,ou=groups,o=*company*"	group/racfGroup/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 RACF 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 RACF 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 RACF 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 RACF 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.

6 Operating and Tuning Racf2ldap

This chapter describes how to operate and tune racf2ldap within your environment.

Customizing Racf2ldap

Almost all customization of racf2ldap occurs in the racf2ldap.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.

racf2ldap configuration settings are stored in ***sdir*/samples/racf2ldap.conf**, where *sdir* is the install directory of the RACF LDAP Bridge. As delivered, this file enables racf2ldap to synchronize RACF with the RACF LDAP Bridge, propagating certain limited profiles and RACF. To propagate additional profiles or fields, or to synchronize with another remote directory, you will have to modify this file, as described in the sections below.

Racf2ldap General Definitions

The following parameters control the global functioning of the racf2ldap daemon, including which RACF LDAP Bridges to synchronize, how to handle error conditions, etc.

Parameter	Default Value	Description
HOST	RACF LDAP Bridge host name	The target RACF LDAP Bridge.
PORT	RACF LDAP Bridge host port	The target port.
version	3	The supported LDAP version. Do not change.
DIR	RACF LDAP Bridge root directory	Base directory for LDAP processing. racf2ldap looks in *sdir*/racf2ldap/new for new SMF80 files to process.
ORGDN	o=*company*	The root dn in the RACF LDAP Bridge.
MANAGERDN	cn=racfManager,	The LDAP Distinguished Name used to perform LDAP updates.

Parameter	Default Value	Description
MANAGERPW	secret	Password for managerdn. May be changed by you to a more secure value. If you change this value, however, you should also change the corresponding value in *sdir*/slapd.conf.
POLL	2	Propagation speed, in seconds.
DEBUG	0	0=All messages 1=Information and error messages 2=Error messages only
RETRY	3	Number of retry attempts for a non-responsive RACF LDAP Bridge.
NOTIFY	Email addresses	Email addresses of personnel to notify in case of errors equal to or greater than the NOTIFYLEVEL, below.
NOTIFYLEVEL	SERIOUS	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
SSL	N	Specifies if SSL is to be used for communication to the RACF LDAP Bridge. This is usually not necessary for local communications with the RACF LDAP Bridge.
SSLKEYFILE	xyz.key	Specifies the path to the SSL keyfile.
SSLKEYPW	secret	Specifies the password for the SSL key.

Parameter	Default Value	Description
REPLOG	/<DIR>/ replog.ldif	Specifies the location of the replog.
LOGLEVEL	.<DIR>/ racf2ldap.log	Path to the logfile containing all messages.
RETAIN	0	Specifies how records are to be written to racf2ldap/old. Values are: -1 = SMF records are deleted once they are processed and are not written to racf2ldap/old. 0 = SMF records are written to racf2ldap/old and are not deleted. *nn* = SMF records are written to racf2ldap/old and records older than *nn* days are deleted once the R2LCLEAN job is run.

Racf2ldap.conf Error Definitions

This section of racf2ldap.conf describes how the racf2ldap daemon should handle various LDAP error conditions returned from the RACF LDAP Bridge. When an LDAP add, modify or delete request from racf2ldap fails on the target RACF LDAP Bridge, the RACF 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 RACF LDAP Bridge.
- **level** - The racf2ldap severity level for this error code: WARNING, SERIOUS, SEVERE or FATAL. See NOTIFYLEVEL, above.
- **action**- The action racf2ldap should take in the event of this error.
  - NONE - Take no action.
  - ABEND - Terminate the racf2ldap task.
  - SLEEP - Retry in 10 seconds.
  - SEND - Email those identified in the NOTIFY statement.
  - MOVE - Move the RACF change to the error directory.

Sample ERROR Definitions

ERROR LDAP_SUCCESS 0 WARNING NONE

This rule tells racf2ldap to take no action on successful LDAP requests.

ERROR LDAP_OPERATIONS_ERROR 1 FATAL ABEND

This rule tells racf2ldap terminate in the event of an LDAP operations error (error code 1).

ERROR LDAP_SERVER_DOWN 81 WARNING SLEEP

This rule tells racf2ldap to wait and then try again in the event that the RACF LDAP Bridge is down (error code 81).

Racf2ldap.conf Rule Definitions

Rules come in two types: DATA and UPDATE. DATA rules manipulate the value provided by RACF into a different format. UPDATE rules control how racf2ldap processes add, modify, or delete operations with the RACF LDAP Bridge.

You may code your own DATA and UPDATE rules to implement custom processing for any given LDAP attribute. If you create your own rule, you should define it with a RULE definition in this section of the configuration file.

RULE name type entry library

All parameters must be separated by one or more spaces.

- **RULE** - Static text identifying this as a RULE statement.
- **name** - The name of this rule, for use in subsequent KEYWORD statements.
- **type** - The type of rule:
  - DATA - For reformatting attribute values.
  - UPDATE - For updating the RACF LDAP Bridge.
- **entry** - The entry point for this rule in the shared library, below.
- **library** - The name of the shared library (DLL) file containing this rule. The product delivers its default rules in *sdir*/sbin/default.dll. Any new shared libraries should reside in *sdir*/sbin.

Sample RULE Definitions

RULE VAL DATA VAL default.dll

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

RULE SetValue UPDATE SetValue default.dll

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

Delivered Rules in Default.dll

In default.dll, racf2ldap delivers the following rules:

Rule	Type	Description
VAL	DATA	Use the value from RACF as-is.
VALS	DATA	Use multiple RACF values as-is.
NOVAL	DATA	Do not populate a value.
BOOLTRUE	DATA	Set the value to TRUE.
BOOLFALSE	DATA	Set the value to FALSE.
GROUPDN	DATA	Create a group DN from the value: cn=value,ou=groups
USERDN	DATA	Create a user DN from the value: uid=value,ou=people
BOOL	DATA	If value is YES or ONE or TRUE, set to TRUE. Otherwise, set to FALSE.
RANGE	DATA	Transform numeric ranges into discreet numbers. Ranges are determined by the – character. For example, change 1-3 into 1 2 3.
SetSuperGroup	UPDATE	Update the superior group of the target with the target dn in the racfSubGroup attribute.
SetValue	UPDATE	Replace the attribute value. If the attribute does not already exist for the entry, add it.
SetMultiValue	UPDATE	Replace the multivalued attribute values. If the attribute does not already exist for the entry, add it.
AddMultiValue	UPDATE	Add the values to those already existing for the multivalued attribute. If the attribute does not already exist for the entry, add it.
DelMultiValue	UPDATE	Delete the values from those already existing for the multivalued attribute.
RemoveAttr	UPDATE	Delete the attribute value.
RemoveAttrs	UPDATE	Delete all attributes matching the wildcard specification in attribute on the KEYWORD statement.
CreateEntry	UPDATE	Create a new entry.

Rule	Type	Description
SetBoolValue	UPDATE	If the value is TRUE, set the value to the last 4 characters of the attribute name.
Copy	UPDATE	Create a resource dn based on the attribute value, and copy the entry referenced by that dn to the target dn.
CopyPermit	UPDATE	Create a permit dn based on the attribute value, and copy the entry referenced by that dn to the target dn.
RemoveSubEntry	UPDATE	In addition to removing the attribute value, remove the dn referenced by the attribute value.
RemoveEntry	UPDATE	Remove the entry referenced by the target.
Reset	UPDATE	Remove a dataset or resource permission.

Racf2ldap.conf Target Definitions

Targets define how racf2ldap names the entries it adds, modifies, or deletes. If you are using racf2ldap 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 racf2ldap uses to construct specific dns. Substitution variables are prefixed by &, indicating a mandatory substitution, or !, indicating optional substitution. Racf2ldap 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 racf2ldap uses when constructing new entries for this target.

Sample TARGET Definitions

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

This target definition, named GROUP, defines the prototype dn for group entries. This prototype dn requires the GROUP keyword. It also specifies that these entries have a fixed parent not defined in this file. Finally, it directs racf2ldap to create new groups that use the racfGroup, top and groupOfNames objectclasses.

Racf2ldap.conf Keyword Definitions

Keywords define how racf2ldap propagates individual RACF fields to the RACF LDAP Bridge. Most KEYWORD statements are delivered disabled (commented-out). To expose other fields, simply uncomment the appropriate keywords in this file. If you are using racf2ldap to synchronize with a remote directory, you may have to add new KEYWORD statements corresponding to the LDAP attributes you wish to synchronize on that remote directory.

KEYWORD command segment keyword target attribute datarule updaterule

All parameters must be separated by one or more spaces:

- **KEYWORD** - Static text identifying this as a KEYWORD statement.
- **command** - The RACF command manipulating the LDAP attribute.
- **Segment** - The RACF segment manipulating the LDAP attribute.
- **keyword** - The RACF keyword manipulating the LDAP attribute.
- **target** - The target for this update operation. The target referenced here must correspond to a TARGET definition, as described in racf2ldap.conf TARGET Definitions.
- **datarule** - The data manipulation rule for this LDAP attribute. The rule referenced here must correspond to a RULE definition, as described in racf2ldap.conf RULE Definitions.
- **updaterule** - The update rule for this LDAP operation. The rule referenced here must correspond to a RULE definition as described in racf2ldap.conf RULE Definitions.

Sample KEYWORD Definitions

KEYWORD ADDUSER BASE NAME User cn VAL SetValue

This keyword definition controls how racf2ldap acts when a RACF administrator issues an ADDUSER command specifying the NAME keyword for the BASE segment. In this case, it will create a target dn based on the User target specified previously in the TARGET definitions. The attribute name updated for this target dn is cn. The data manipulation rule is VAL, as defined in the RULE definitions above. This rule simply moves the keyword value as-is. The update rule is SetValue. This sets the attribute value for cn to the keyword value specified in NAME, adding the cn attribute to the target entry if it does not already exist.

KEYWORD CONNECT BASE USERID Group member USERDN AddMultiValue

This keyword definition controls how racf2ldap acts when a RACF administrator issues a CONNECT command specifying the USERID keyword for the BASE segment. In this case, it will create a target dn based on the Group target specified previously in the TARGET definitions. The attribute name updated for this target dn is member. The data manipulation rule is GROUPDN, as defined in the RULE definitions above. This rule takes the keyword value and uses it to create a user dn. The update rule is AddMultiValue. This adds the attribute value to the existing values for the member attribute. If member does not exist for the target entry, then this rule adds it.

#RKEYWORD RDEFINE STDATA GROUP ResourceSegStdat racfStdatGroup VAL SetValue

This keyword is commented out. To activate it, simply remove the “#R” from the beginning of the line, so that it looks like this:

KEYWORD RDEFINE STDATA GROUP ResourceSegStdat racfStdatGroup VAL SetValue

Comments always start with the # character, and may be followed by an optional character before the “KEYWORD” text.

A Appendix: The LDAP Schema File

The RACF LDAP Bridge interacts with OVSI using a mapping file (RACF.xml) that is provided by OVSI and a schema provided by the RACF 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 RACF 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 'racfData'
DESC 'racfData'
EQUALITY caseIgnoreMatch
SUBSTR caseIgnoreSubstringsMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE)
```

Attribute definitions support the following statements:

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 may 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/racf.schema file, near the bottom. Here is a typical objectclass definition:

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

Objectclass definitions support the following statements:

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

RACF Mapping Information

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

LDAP object class	LDAP attribute name	RACF field name
racfClass	racfClass	Class
racfClass	racfRefreshGeneric	Refresh GENERIC
racfClass	racfRefreshGlobal	Refresh GLOBAL
racfClass	racfRefreshRaclist	Refresh RACLIST
racfDataset	racfAudit	Audit
racfDataset	racfCategory	Category
racfDataset	racfClass	Class
racfDataset	racfCreatedate	Create Date
racfDataset	racfCreateGroup	Create Group
racfDataset	racfData	Installation Data
racfDataset	racfDataset	Dataset Name
racfDataset	racfDatasetType	Dataset Type
racfDataset	racfErase	Erase
racfDataset	racfFileseq	File Sequence
racfDataset	racfGlobalaudit	Global Audit
racfDataset	racfLevel	Log Level
racfDataset	racfModifierGroup	Last Modifier Group
racfDataset	racfModifierUser	Last Modifier User
racfDataset	racfModifyDate	Last Modified Date
racfDataset	racfNotify	Notify
racfDataset	racfOwner	Owner
racfDataset	racfRetpd	Retention Period
racfDataset	racfSeclabel	Security Label
racfDataset	racfSeclevel	Security Level
racfDataset	racfUacc	Universal Access
racfDataset	racfUnit	Unit
racfDataset	racfVolser	Volume Serial
racfDataset	racfVolsers	Volume Serials
racfDataset	racfWarning	Warning
racfDatasetAppcport	racfDatasetAppcport	APPC Port
racfDatasetAppcport	racfDataset	Dataset Name
racfDatasetAppcport	racfVolser	Volume Serial
racfDatasetConsole	racfConsole	Console

LDAP object class	LDAP attribute name	RACF field name
racfDatasetConsole	racfDataset	Dataset Name
racfDatasetConsole	racfVolser	Volume Serial
racfDatasetJesinput	racfDataset	Dataset Name
racfDatasetJesinput	racfJesinput	JES Input Device
racfDatasetJesinput	racfVolser	Volume Serial
racfDatasetPermitAp	racfAccess	Access
racfDatasetPermitAp	racfAppcport	APPC Port
racfDatasetPermitAp	racfDataset	Dataset Name
racfDatasetPermitAp	racfPermitId	Permitted ID
racfDatasetPermitAp	racfVolser	Volume Serial
racfDatasetPermitCo	racfAccess	Access
racfDatasetPermitCo	racfConsole	Console
racfDatasetPermitCo	racfDataset	Dataset Name
racfDatasetPermitCo	racfPermitId	Permitted ID
racfDatasetPermitCo	racfVolser	Volume Serial
racfDatasetPermitId	racfAccess	racfAccess
racfDatasetPermitId	racfDataset	Dataset Name
racfDatasetPermitId	racfPermitId	Permitted ID
racfDatasetPermitId	racfVolser	Volume Serial
racfDatasetPermitJe	racfAccess	Access
racfDatasetPermitJe	racfDataset	Dataset Name
racfDatasetPermitJe	racfJesinput	JES Input Device
racfDatasetPermitJe	racfPermitId	Permitted ID
racfDatasetPermitJe	racfVolser	Volume Serial
racfDatasetPermitPr	racfAccess	Access
racfDatasetPermitPr	racfDataset	Dataset Name
racfDatasetPermitPr	racfPermitId	Permitted ID
racfDatasetPermitPr	racfProgram	Program
racfDatasetPermitPr	racfVolser	Volume Serial
racfDatasetPermitSy	racfAccess	Access
racfDatasetPermitSy	racfDataset	Dataset Name
racfDatasetPermitSy	racfPermitId	Permitted ID
racfDatasetPermitSy	racfSysid	System ID
racfDatasetPermitSy	racfVolser	Volume Serial
racfDatasetPermitTe	racfAccess	Access
racfDatasetPermitTe	racfDataset	Dataset Name
racfDatasetPermitTe	racfPermitId	Permitted ID

LDAP object class	LDAP attribute name	RACF field name
racfDatasetPermitTe	racfTerminal	Terminal
racfDatasetPermitTe	racfVolser	Volume Serial
racfDatasetProgram	racfDataset	Dataset Name
racfDatasetProgram	racfProgram	Program
racfDatasetProgram	racfVolser	Volume Serial
racfDatasetSegDfp	racfDataset	Dataset Name
racfDatasetSegDfp	racfDfpOwner	DFP Resource Owner
racfDatasetSegDfp	racfSegment	Segment Identifier
racfDatasetSegDfp	racfVolser	Volume Serial
racfDatasetSegTme	racfDataset	Dataset Name
racfDatasetSegTme	racfSegment	Segment Identifier
racfDatasetSegTme	racfTmeRoles	Role Access Spec
racfDatasetSegTme	racfVolser	Volume Serial
racfDatasetSysid	racfDataset	Dataset Name
racfDatasetSysid	racfSysid	System ID
racfDatasetSysid	racfVolser	Volume Serial
racfDatasetTerminal	racfDataset	Dataset Name
racfDatasetTerminal	racfTerminal	Terminal
racfDatasetTerminal	racfVolser	Volume Serial
racfGroup	cn	Group Name
racfGroup	racfCreatedate	Create Date
racfGroup	racfData	Installation Data
racfGroup	racfMember	Member
racfGroup	racfModel	Model Dataset Profile
racfGroup	racfModifierGroup	Last Modifier Group
racfGroup	racfModifierUser	Last Modifier User
racfGroup	racfModifyDate	Last Modified Date
racfGroup	racfOwner	Owner
racfGroup	racfSubgroup	Subordinate Group
racfGroup	racfSupgroup	Superior Group
racfGroup	racfTermuacc	Terminal Unv. Access
racfGroup	racfUacc	Universal Access
racfGroup	racfUniversal	Universal Group
racfGroupSegDfp	cn	Group Name
racfGroupSegDfp	racfDfpDataappl	DFP Data Appl-ID
racfGroupSegDfp	racfDfpDataclas	DFP Dflt. Data Class
racfGroupSegDfp	racfDfpMgmtclas	DFP Management Class

LDAP object class	LDAP attribute name	RACF field name
racfGroupSegDfp	racfDfpStorclas	DFP Storage Class
racfGroupSegDfp	racfSegment	Segment Identifier
racfGroupSegOmvs	cn	Group Name
racfGroupSegOmvs	racfOmvsGid	Unix Group-ID (GID)
racfGroupSegOmvs	racfSegment	Segment Identifier
racfGroupSegOvm	cn	Group Name
racfGroupSegOvm	racfOvmGid	OVM Group-ID (GID)
racfGroupSegOvm	racfSegment	Segment Identifier
racfGroupSegTme	cn	Group Name
racfGroupSegTme	racfSegment	Segment Identifier
racfGroupSegTme	racfTmeRoles	Roles
racfOption	racfAddCreator	ADDCREATOR
racfOption	racfAdsp	ADSP
racfOption	racfApplaudit	APPLAUDIT
racfOption	racfAuditSeclabel	SECLABELAUDIT
racfOption	racfAuditSeclevel	SECLEVELAUDIT
racfOption	racfAuditSpecial	SAUDIT
racfOption	racfCatDsns	CATDSNS
racfOption	racfClassActive	CLASSACT
racfOption	racfClassAudit	AUDIT
racfOption	racfClassGencmd	GENCMD
racfOption	racfClassGeneric	GENERIC
racfOption	racfClassGenlist	GENLIST
racfOption	racfClassGlobal	GLOBAL
racfOption	racfClassRaclist	RACLIST
racfOption	racfClassStatistics	STATISTICS
racfOption	racfCmdViol	CMDVIOL
racfOption	racfCompatMode	COMPATMODE
racfOption	racfControlSeclabel	SECLABELCONTROL
racfOption	racfDataset	RACF Dataset
racfOption	racfEgn	EGN
racfOption	racfErase	ERASE ALL
racfOption	racfEraseLevel	ERASE SECLEVEL
racfOption	racfGenericOwner	GENERICOWNER
racfOption	racfGrplist	GRPLIST
racfOption	racfInactiveInterval	INACTIVE
racfOption	racfInitStats	INITSTATS

LDAP object class	LDAP attribute name	RACF field name
racfOption	racfKerbLevel	KERBLVL
racfOption	racfMlactive	MLACTIVE
racfOption	racfMlquiet	MLQUIET
racfOption	racfMls	MLS
racfOption	racfMlstable	MLSTABLE
racfOption	racfModel	MODEL
racfOption	racfModifierGroup	Last Modifier Group
racfOption	racfModifierUser	Last Modifier User
racfOption	racfModifyDate	Last Modified Date
racfOption	racfOperAudit	OPERAUDIT
racfOption	racfOption	Option Type
racfOption	racfPrefix	PREFIX
racfOption	racfProtectAll	PROTECTALL
racfOption	racfRealDsn	REALDSN
racfOption	racfRefreshWhenProg	REFRESH WHEN PROGRAM
racfOption	racfRetpd	RETPD
racfOption	racfSecLabelControl	SECLABELCONTROL
racfOption	racfSessInterval	SESSIONINTERVAL
racfOption	racfSingleDsn	SINGLEDSN
racfOption	racfTapeDsn	TAPEDSN
racfOption	racfTapevolProt	Tape Volser Protection
racfOption	racfTerminalUacc	TERMINAL
racfOption	racfUnit	RACF Database Unit
racfOption	racfVolser	RACF Database Volume
racfOption	racfWhenProgram	WHEN PROGRAM
racfOption	racfXbmAllRacf	JES XMBALLRACF
racfOptionSegJes	racfJesBatchAllRacf	JES BATCHALLRACF
racfOptionSegJes	racfJesEarlyVerify	JES EARLYVERIFY
racfOptionSegJes	racfJesNjeUserid	JES NJEUSERID
racfOptionSegJes	racfJesUndefinedUser	JES UNDEFINED
racfOptionSegJes	racfJesXbmAllRacf	JES XBMALLRACF
racfOptionSegJes	racfOption	Option Type
racfOptionSegJes	racfSegment	Segment Identifier
racfOptionSegLang	racfLanguagePrimary	LANGUAGE PRIMARY
racfOptionSegLang	racfLanguageSecondar	LANGUAGE SECONDARY
racfOptionSegLang	racfOption	Option Type
racfOptionSegLang	racfSegment	Segment Identifier

LDAP object class	LDAP attribute name	RACF field name
racfOptionSegLog	racfClassLogAlways	LOGOPTIONS ALWAYS
racfOptionSegLog	racfClassLogDefault	LOGOPTIONS DEFAULT
racfOptionSegLog	racfClassLogFailure	LOGOPTIONS FAILURES
racfOptionSegLog	racfClassLogNever	LOGOPTIONS NEVER
racfOptionSegLog	racfClassLogSuccess	LOGOPTIONS SUCCESS
racfOptionSegLog	racfOption	Option Type
racfOptionSegLog	racfSegment	Segment Identifier
racfOptionSegModel	racfModelGdg	MODEL GDG
racfOptionSegModel	racfModelGroup	MODEL GROUP
racfOptionSegModel	racfModelUser	MODEL USER
racfOptionSegModel	racfOption	Option Type
racfOptionSegModel	racfSegment	Segment Identifier
racfOptionSegPass	racfOption	Option Type
racfOptionSegPass	racfPasswordHistory	PASSWORD HISTORY
racfOptionSegPass	racfPasswordInterval	PASSWORD INTERVAL
racfOptionSegPass	racfPasswordRevoke	PASSWORD REVOKE
racfOptionSegPass	racfPasswordRule1	PASSWORD RULE1
racfOptionSegPass	racfPasswordRule2	PASSWORD RULE2
racfOptionSegPass	racfPasswordRule3	PASSWORD RULE3
racfOptionSegPass	racfPasswordRule4	PASSWORD RULE4
racfOptionSegPass	racfPasswordRule5	PASSWORD RULE5
racfOptionSegPass	racfPasswordRule6	PASSWORD RULE6
racfOptionSegPass	racfPasswordRule7	PASSWORD RULE7
racfOptionSegPass	racfPasswordRule8	PASSWORD RULE8
racfOptionSegPass	racfPasswordWarning	PASSWORD WARNING
racfOptionSegPass	racfSegment	Segment Identifier
racfOptionSegRvary	racfOption	Option Type
racfOptionSegRvary	racfRvaryStatusPw	RVARY STATUS
racfOptionSegRvary	racfRvarySwitchPw	RVARY SWITCH
racfOptionSegRvary	racfSegment	Segment Identifier
racfResource	racfAppldata	Application Data
racfResource	racfAudit	Audit
racfResource	racfCategory	Category
racfResource	racfClass	Class
racfResource	racfClassnum	Class Number
racfResource	racfCreatedate	Create Date
racfResource	racfData	Installation Data

LDAP object class	LDAP attribute name	RACF field name
racfResource	racfGeneric	Generic
racfResource	racfGlobalaudit	Global Audit
racfResource	racfLevel	Log Level
racfResource	racfModifierGroup	Last Modifier Group
racfResource	racfModifierUser	Last Modifier User
racfResource	racfModifyDate	Last Modified Date
racfResource	racfNotify	Notify
racfResource	racfOwner	Owner
racfResource	racfResource	Resource Name
racfResource	racfSeclabel	Security Label
racfResource	racfSeclevel	Security Level
racfResource	racfTapeAuto	Auto Protect
racfResource	racfTapeSingledsn	Single DSN
racfResource	racfTapeVtoc	Tape VTOC
racfResource	racfTimezone	Time Zone
racfResource	racfTvolCreatedate	Tape Create Date
racfResource	racfTvolCreatename	Tape Create Name
racfResource	racfTvolDiscrete	Tape Discrete
racfResource	racfUacc	Universal Access
racfResource	racfVolser	Volume Serial
racfResource	racfVolsers	Volume Serials
racfResource	racfWarning	Warning
racfResourceAppcport	racfAppcport	APPC Port
racfResourceAppcport	racfClass	Class
racfResourceAppcport	racfResource	Resource Name
racfResourceConsole	racfClass	Class
racfResourceConsole	racfConsole	Console
racfResourceConsole	racfResource	Resource Name
racfResourceJesinput	racfClass	Class
racfResourceJesinput	racfJesinput	JES Input Device
racfResourceJesinput	racfResource	Resource Name
racfResourceMember	racfCategory	Category
racfResourceMember	racfClass	Member Class
racfResourceMember	racfPadsData	Program Access Data
racfResourceMember	racfResource	Resource Group Name
racfResourceMember	racfResourceMember	Resource Member
racfResourceMember	racfSeclevel	Security Level

LDAP object class	LDAP attribute name	RACF field name
racfResourceMember	racfUacc	Universal Access
racfResourceMember	racfVmexitdata	VMX Event Auditing
racfResourceMember	racfVolser	Volume Serial
racfResourcePermitAp	racfAccess	Access
racfResourcePermitAp	racfAppcport	APPC Port
racfResourcePermitAp	racfClass	Class
racfResourcePermitAp	racfPermitId	Permitted ID
racfResourcePermitAp	racfResource	Resource Name
racfResourcePermitCo	racfAccess	Access
racfResourcePermitCo	racfClass	Class
racfResourcePermitCo	racfConsole	Console
racfResourcePermitCo	racfPermitId	Permitted ID
racfResourcePermitCo	racfResource	Resource Name
racfResourcePermitId	racfAccess	Access
racfResourcePermitId	racfClass	Class
racfResourcePermitId	racfPermitId	Permitted ID
racfResourcePermitId	racfResource	Resource Name
racfResourcePermitJe	racfAccess	Access
racfResourcePermitJe	racfClass	Class
racfResourcePermitJe	racfJesinput	JES Input Device
racfResourcePermitJe	racfPermitId	Permitted ID
racfResourcePermitJe	racfResource	Resource Name
racfResourcePermitPr	racfAccess	Access
racfResourcePermitPr	racfClass	Class
racfResourcePermitPr	racfPermitId	Permitted ID
racfResourcePermitPr	racfResource	Resource Name
racfResourcePermitSy	racfAccess	Access
racfResourcePermitSy	racfClass	Class
racfResourcePermitSy	racfPermitId	Permitted ID
racfResourcePermitSy	racfResource	Resource Name
racfResourcePermitSy	racfSysid	System ID
racfResourcePermitTe	racfAccess	Access
racfResourcePermitTe	racfClass	Class
racfResourcePermitTe	racfPermitId	Permitted ID
racfResourcePermitTe	racfResource	Resource Name
racfResourcePermitTe	racfTerminal	Terminal
racfResourceProgram	racfClass	Class

LDAP object class	LDAP attribute name	RACF field name
racfResourceProgram	racfResource	Resource Name
racfResourceSegDlf	racfClass	Class
racfResourceSegDlf	racfDlfJobs	DLF Job Names
racfResourceSegDlf	racfDlfRetain	DLF Retain
racfResourceSegDlf	racfResource	Resource Name
racfResourceSegDlf	racfSegment	Segment Identifier
racfResourceSegKerb	racfClass	Class
racfResourceSegKerb	racfKerbDeftktlfe	Default Ticket Life
racfResourceSegKerb	racfKerbEncryptdes	Encrypt DES
racfResourceSegKerb	racfKerbEncryptdes3	Encrypt DES3
racfResourceSegKerb	racfKerbEncryptdesd	Encrypt DESD
racfResourceSegKerb	racfKerbKerbname	Local Realm Name
racfResourceSegKerb	racfKerbKeyver	Key Version
racfResourceSegKerb	racfKerbMaxtktlfe	Maximum Ticket Life
racfResourceSegKerb	racfKerbMintktlfe	Minimum Ticket Life
racfResourceSegKerb	racfKerbPassword	Kerberos Password
racfResourceSegKerb	racfResource	Resource Name
racfResourceSegKerb	racfSegment	Segment Identifier
racfResourceSegSess	racfClass	Class
racfResourceSegSess	racfResource	Resource Name
racfResourceSegSess	racfSegment	Segment Identifier
racfResourceSegSess	racfSessFailMax	Max Session Failures
racfResourceSegSess	racfSessInterval	Session Interval
racfResourceSegSess	racfSessKey	Session Key
racfResourceSegSess	racfSessKeyDate	Session Key Date
racfResourceSegSess	racfSessLocked	Session Locked
racfResourceSegSess	racfSessSecurity	Session Security
racfResourceSegSsign	racfClass	Class
racfResourceSegSsign	racfResource	Resource Name
racfResourceSegSsign	racfSegment	Segment Identifier
racfResourceSegSsign	racfSsignKeyencrypt	Encrypted Key
racfResourceSegSsign	racfSsignKeymasked	Masked Key
racfResourceSegStdat	racfClass	Class
racfResourceSegStdat	racfResource	Resource Name
racfResourceSegStdat	racfSegment	Segment Identifier
racfResourceSegStdat	racfStdatGroup	Group-ID
racfResourceSegStdat	racfStdatPrivileged	Privileged Task

LDAP object class	LDAP attribute name	RACF field name
racfResourceSegStdat	racfStdatTrace	Trace Task
racfResourceSegStdat	racfStdatTrusted	Trusted Task
racfResourceSegStdat	racfStdatUser	User-ID
racfResourceSegSvfmr	racfClass	Class
racfResourceSegSvfmr	racfResource	Resource Name
racfResourceSegSvfmr	racfSegment	Segment Identifier
racfResourceSegSvfmr	racfSvfmrParm	Parameter File
racfResourceSegSvfmr	racfSvfmrScript	Script File
racfResourceSegTme	racfClass	Class
racfResourceSegTme	racfResource	Resource Name
racfResourceSegTme	racfSegment	Segment Identifier
racfResourceSegTme	racfTmeChild	Child Role
racfResourceSegTme	racfTmeGroup	Group-ID
racfResourceSegTme	racfTmeParent	Parent Role
racfResourceSegTme	racfTmeResource	Resource Access Spec
racfResourceSegTme	racfTmeRoles	Role Access Spec
racfResourceSegWhen	racfClass	Class
racfResourceSegWhen	racfResource	Resource Name
racfResourceSegWhen	racfSegment	Segment Identifier
racfResourceSegWhen	racfWhenDays	When Days
racfResourceSegWhen	racfWhenTime	When Time
racfResourceSysid	racfClass	Class
racfResourceSysid	racfResource	Resource Name
racfResourceSysid	racfSysid	System ID
racfResourceTerminal	racfClass	Class
racfResourceTerminal	racfResource	Resource Name
racfResourceTerminal	racfTerminal	Terminal
racfResourceTvtoc	racfDataset	Dataset Name
racfUser	cn	User Name
racfUser	racfAdsp	ADSP
racfUser	racfAuditor	Auditor
racfUser	racfAuthority	Group Authority
racfUser	racfCategory	Category
racfUser	racfClauth	Class Authorizations
racfUser	racfCreatedate	Create Date
racfUser	racfData	Installation Data
racfUser	racfDfltgrp	Default Group

LDAP object class	LDAP attribute name	RACF field name
racfUser	racfExpired	Expired
racfUser	racfGrpacc	Group Access
racfUser	racfLastjobdate	Last Job Date
racfUser	racfLastjobtime	Last Job Time
racfUser	racfModel	Model Dataset Profile
racfUser	racfModifierGroup	Last Modifier Group
racfUser	racfModifierUser	Last Modifier User
racfUser	racfModifyDate	Last Modified Date
racfUser	racfNopassword	No Password
racfUser	racfOidcard	Operator-ID Card
racfUser	racfOperations	Operations
racfUser	racfOwner	Owner
racfUser	racfPassword	Password
racfUser	racfPassworddate	Password Date
racfUser	racfPasswordgen	Password Generation
racfUser	racfPasswordInterval	Password Interval
racfUser	racfRestricted	Restricted
racfUser	racfResumeDate	Resume Date
racfUser	racfRevoke	Revoke
racfUser	racfRevokecount	Revoke Count
racfUser	racfRevokeDate	Revoke Count
racfUser	racfSeclabel	Security Label
racfUser	racfSeclevel	Security Level
racfUser	racfSpecial	Special
racfUser	racfUacc	Universal Access
racfUser	racfUaudit	User Audit
racfUser	uid	User-ID
racfUserConnect	racfAdsp	ADSP
racfUserConnect	racfAuditor	Auditor
racfUserConnect	racfAuthority	Authority
racfUserConnect	racfCreatedate	Create Date
racfUserConnect	racfGroup	Group
racfUserConnect	racfGrpacc	Group Access
racfUserConnect	racfModifierGroup	Last Modifier Group
racfUserConnect	racfModifierUser	Last Modifier User
racfUserConnect	racfModifyDate	Last Modified Date
racfUserConnect	racfOperations	Operations

LDAP object class	LDAP attribute name	RACF field name
racfUserConnect	racfOwner	Owner
racfUserConnect	racfResumeDate	Resume
racfUserConnect	racfRevoke	Revoke
racfUserConnect	racfRevokeDate	Revoke Date
racfUserConnect	racfSpecial	Special
racfUserConnect	racfTermuacc	Terminal Unv. Access
racfUserConnect	racfUacc	Universal Access
racfUserConnect	uid	User-ID
racfUserSegCics	racfCicsOpclass	CICS Operator Class
racfUserSegCics	racfCicsOpident	Cics Operator-ID
racfUserSegCics	racfCicsOpprty	CICS Oper. Priority
racfUserSegCics	racfCicsTimeout	CICS Timeout
racfUserSegCics	racfCicsXrfsoff	Force XRF Logoff
racfUserSegCics	racfSegment	Segment Identifier
racfUserSegCics	uid	User-ID
racfUserSegDce	racfDceAutologon	DCE Autologon
racfUserSegDce	racfDceHomecell	DCE Home Cell Name
racfUserSegDce	racfDceHomeuuid	DCE Home UUID
racfUserSegDce	racfDceName	DCE Name
racfUserSegDce	racfDceUuid	DCE UUID
racfUserSegDce	racfSegment	Segment Identifier
racfUserSegDce	uid	User-ID
racfUserSegDfp	racfDfpDataappl	DFP Data Appl-ID
racfUserSegDfp	racfDfpDataclas	DFP Dflt. Data Class
racfUserSegDfp	racfDfpMgmtclas	DFP Management Class
racfUserSegDfp	racfDfpStorclas	DFP Storage Class
racfUserSegDfp	racfSegment	Segment Identifier
racfUserSegDfp	uid	User-ID
racfUserSegKerb	racfKerbEncryptdes	Encrypt DES
racfUserSegKerb	racfKerbEncryptdes3	Encrypt DES3
racfUserSegKerb	racfKerbEncryptdesd	Encrypt DESD
racfUserSegKerb	racfKerbKerbname	Principal Name
racfUserSegKerb	racfKerbKeyver	Key Version
racfUserSegKerb	racfKerbMaxtktlfe	Maximum Ticket Life
racfUserSegKerb	racfSegment	Segment Identifier
racfUserSegKerb	uid	User-ID
racfUserSegLanguage	racfLanguagePrimary	Language Primary

LDAP object class	LDAP attribute name	RACF field name
racfUserSegLanguage	racfLanguageSecondar	Language Secondary
racfUserSegLanguage	racfSegment	Segment Identifier
racfUserSegLanguage	uid	User-ID
racfUserSegLnotes	racfLnotesSname	Short Name
racfUserSegLnotes	racfSegment	Segment Identifier
racfUserSegLnotes	uid	User-ID
racfUserSegNds	racfNdsUname	NDS User Name
racfUserSegNds	racfSegment	Segment Identifier
racfUserSegNds	uid	User-ID
racfUserSegNetview	racfNetviewConsname	Console Name
racfUserSegNetview	racfNetviewCtl	Control
racfUserSegNetview	racfNetviewDomains	Domains
racfUserSegNetview	racfNetviewIc	Initial Commands
racfUserSegNetview	racfNetviewMsgrecvr	Receive Messages
racfUserSegNetview	racfNetviewNgmfadmn	Graphic Monitor
racfUserSegNetview	racfNetviewNgmfvspn	View Span
racfUserSegNetview	racfNetviewOpclass	Operator Scope Class
racfUserSegNetview	racfSegment	Segment Identifier
racfUserSegNetview	uid	User-ID
racfUserSegOmvs	racfOmvsAssizemax	Max Address Space
racfUserSegOmvs	racfOmvsCputimemax	Max CPU Time
racfUserSegOmvs	racfOmvsFileprocmax	Max Open Files
racfUserSegOmvs	racfOmvsHome	Home Directory Path
racfUserSegOmvs	racfOmvsMmapareamax	Max MMAP Area
racfUserSegOmvs	racfOmvsProcusermax	Max Active Processes
racfUserSegOmvs	racfOmvsProgram	Shell Program Path
racfUserSegOmvs	racfOmvsThreadsmax	Max Threads
racfUserSegOmvs	racfOmvsUid	Unix User-ID (UID)
racfUserSegOmvs	racfSegment	Segment Identifier
racfUserSegOmvs	uid	User-ID
racfUserSegOper	racfOperAltgrp	Alt console group
racfUserSegOper	racfOperAuth	Console Authorities
racfUserSegOper	racfOperAuto	Automated Messages
racfUserSegOper	racfOperCmdsys	Command System
racfUserSegOper	racfOperDom	Delete Operator Msgs
racfUserSegOper	racfOperKey	Console Key
racfUserSegOper	racfOperLevel	Message Level

LDAP object class	LDAP attribute name	RACF field name
racfUserSegOper	racfOperLogcmdresp	Log Command Response
racfUserSegOper	racfOperMform	Message Format
racfUserSegOper	racfOperMigid	Migration ID
racfUserSegOper	racfOperMonitor	Monitor Events
racfUserSegOper	racfOperMscope	Message Scope
racfUserSegOper	racfOperRoutcode	Message Routing Code
racfUserSegOper	racfOperStorage	Storage
racfUserSegOper	racfOperSystem	System
racfUserSegOper	racfOperUd	Undelivered Messages
racfUserSegOper	racfSegment	Segment Identifier
racfUserSegOper	uid	User-ID
racfUserSegOvm	racfOvmFsroot	File System Root
racfUserSegOvm	racfOvmGid	OVM Group-ID (GID)
racfUserSegOvm	racfOvmHome	Home
racfUserSegOvm	racfOvmProgram	Initial Program
racfUserSegOvm	racfOvmUid	OVM User-ID (UID)
racfUserSegOvm	racfSegment	Segment Identifier
racfUserSegOvm	uid	User-ID
racfUserSegTso	racfSegment	Segment Identifier
racfUserSegTso	racfTsoAcctnum	Account Number
racfUserSegTso	racfTsoCommand	Command
racfUserSegTso	racfTsoDest	Destination
racfUserSegTso	racfTsoHoldclass	Hold Class
racfUserSegTso	racfTsoJobclass	Job Class
racfUserSegTso	racfTsoMaxsize	Region Size Max
racfUserSegTso	racfTsoMsgclass	Message Class
racfUserSegTso	racfTsoPerfgroup	Performance Group
racfUserSegTso	racfTsoProc	Logon Procedure
racfUserSegTso	racfTsoSeclabel	Security Label
racfUserSegTso	racfTsoSize	Region Size Default
racfUserSegTso	racfTsoSysoutclass	Sysout Class
racfUserSegTso	racfTsoUnit	Unit for Procedures
racfUserSegTso	racfTsoUserdata	User Data
racfUserSegTso	uid	User-ID
racfUserSegWhen	racfSegment	Segment Identifier
racfUserSegWhen	racfWhenDays	When Days
racfUserSegWhen	racfWhenTime	When Time

LDAP object class	LDAP attribute name	RACF field name
racfUserSegWhen	uid	User-ID
racfUserSegWorkattr	racfSegment	Segment Identifier
racfUserSegWorkattr	racfWorkattrAccnt	Account
racfUserSegWorkattr	racfWorkattrAddr1	Address Line 1
racfUserSegWorkattr	racfWorkattrAddr2	Address Line 2
racfUserSegWorkattr	racfWorkattrAddr3	Address Line 3
racfUserSegWorkattr	racfWorkattrAddr4	Address Line 4
racfUserSegWorkattr	racfWorkattrBldg	Building
racfUserSegWorkattr	racfWorkattrDept	Department
racfUserSegWorkattr	racfWorkattrName	Name
racfUserSegWorkattr	racfWorkattrRoom	Room
racfUserSegWorkattr	uid	User-ID
racfVolser	racfVolser	Volume Serial

Index

R

S

T

U

Z